

# MSc Thesis Report

PERFORMANCE MODELLING OF A HYBRID ELECTRIC  
PROPULSION SYSTEM FOR EVTOL AIRCRAFT

ALFAROUK ADEL KHALIL



AYED-Engineering GmbH | Delft University of Technology



# MSC THESIS REPORT

## PERFORMANCE MODELLING OF A HYBRID ELECTRIC PROPULSION SYSTEM FOR eVTOL AIRCRAFT

by

**Alfarouk Adel Khalil**

in partial fulfillment of the requirements for the degree of

**Master of Science**  
in Aerospace Engineering

at the Delft University of Technology,  
to be defended publicly on Tuesday August 27<sup>th</sup>, 2024 at 10:00 AM.

Supervisors: Dr. ir. W.PJ (Wilfried) Visser  
Dr. ir. C. (Chiara) Falsetti  
Ir. Alexander Lautenschläger

Thesis committee: Prof. dr. ir. P. (Piero)  
Colonna di Paliano TU Delft  
Dr. ir. W.PJ (Wilfried) Visser, TU Delft  
Dr. ir. C. (Chiara) Falsetti TU Delft  
Ir. P. C. (Paul) Roling, TU Delft  
Ir. Alexander Lautenschläger AYED-ENGINEERING GmbH

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Cover Image: eVTOL concept by Vertical Aerospace [1]





# PREFACE

This thesis project represents the culmination of my journey toward earning a Master of Science degree in Aerospace Engineering from TU Delft. It does sound amazing I have to admit when reading that again, but my god that was a difficult journey. The report before you provides a comprehensive overview of the work I have undertaken over the past year. The work begins with an extensive research on eVTOLs, fuel cells and their balance of plant component. It will take you through the methodologies I used to develop the tool that I used later to estimate the propulsion system weight. Readers with a background in propulsion, or aerospace engineering should find the content accessible. While prior knowledge of fuel cells is not required, I have included a thorough introduction to help guide the reader.

That thesis might sound simple, but as A. Einstein said: *"The definition of genius is taking the complex and making it simple"*. I am not saying I am a genius god forbid, but it was not an easy job working out this thesis topic. What made my job even more difficult is that, I had to work as a part-time teaching assistant to pay for the 2284 euros tuition fees and the expensive living costs of the Netherlands. A financial stress monster, but thanks to God and my kind mother, god bless her soul, I was able to face this monster. Thank you mama...te amo mucho!

Obtaining a masters degree from this university is a very challenging and demanding task. If you are reading this, that means I have made it somehow and if I can do it anyone can. I have learned a lot from this experience and I would like to thank my supervisors: Alexander Lautenschläger, Chiara Falsetti and Wilfried Visser for their feedback and guidance throughout my thesis. I also want to thank them for their patience, because I am not the best student to supervise.

I have included an acknowledgements chapter before the references at the end of the report, because I am trying to be concise here in the preface. Please go and read it!

*Alfarouk Adel Khalil  
Delft, August 2024*



# EXECUTIVE SUMMARY

This report presents an in-depth study titled "Performance Modeling of a Hybrid Electric Propulsion System for eVTOL Aircraft," which aims to develop a preliminary performance model for hybrid propulsion systems tailored for electric vertical take-off and landing (eVTOL) aircraft. The primary goal is to achieve a balance between efficiency, and optimal component dimensions for different flight missions. The hybrid system integrates a fuel cell and a turbo-generator to address the limitations of current propulsion technologies and meet the diverse power requirements of eVTOL aircraft.

The research begins with an exploration of the eVTOL market, identifying the current technological trends and performance capabilities of existing electric vertical and short take-off and landing aircraft. This section underscores the need for hybrid propulsion systems to overcome the constraints of pure electric propulsion, such as limited range and payload capacity. The study then delves into the architecture of hybrid propulsion systems, examining various configurations including series, parallel, and series-parallel systems. The findings highlight the advantages of integrating fuel cells with turbo-generators to enhance power density and efficiency, essential for the operational demands of eVTOL aircraft.

A performance model is developed, focusing on the critical components of the hybrid system, including fuel cells and turbo-generators. The model provides mass estimations and simulates system performance across various flight scenarios such as take-off and cruise. With a fuel cell voltage efficiency of 43%, an eVTOL of MTOW of 3175 kg and a structural weight of 1905 kg can carry a maximum of 174.03 kg as a payload. The propulsion system weights approximately 1041 kg, in which the fuel cell and its balance of plant components contributes with 819 kg and the turbogenerator contributes with 222 kg. Moreover, the model predicts that the use of a compressor efficiency of 85% results in a higher payload mass of approximately 200 kg. The analysis reveals that the optimal fuel cell efficiency for maximizing payload capacity is approximately 43%. At this efficiency, the fuel cell achieves a balance between weight and performance, significantly impacting the overall system mass and aircraft payload capabilities. Furthermore, the results indicate that higher voltage efficiencies, although beneficial for reducing fuel consumption, result in heavier fuel cells and associated components, thus affecting the overall system weight and aircraft performance.

This research provides insights into the complexities of designing hybrid electric propulsion systems for eVTOL aircraft. The developed model serves as a preliminary tool for future advancements in sustainable aviation technology, supporting the creation of more efficient and capable eVTOL systems. These findings contribute to the ongoing efforts to develop greener urban air mobility solutions, addressing the growing demand for environmentally sustainable transport options.

In conclusion, the performance model developed in this study offers a robust framework for evaluating and optimizing hybrid electric propulsion systems. The insights gained from this research are crucial for advancing eVTOL technology, paving the way to do initial trade off studies. The model developed can be further refined in the future to study the new generation of eVTOLs.

# Contents

<b>Executive Summary</b>	<b>v</b>
<b>Nomenclature</b>	<b>ix</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Research goal &amp; Questions</b>	<b>2</b>
2.1 Research Goal	2
2.2 Research Question(s)	2
<b>3 eVTOL &amp; eSTOL Applications</b>	<b>3</b>
3.1 Basic concepts & current market	3
3.2 Vehicle performance	4
<b>4 Hybrid Propulsion System</b>	<b>6</b>
4.1 Hybrid propulsion architectures	6
4.1.1 Series power-train architecture	6
4.1.2 Parallel power-train architecture	7
4.1.3 Series-Parallel power-train architecture	7
4.2 Fuel cells	8
4.2.1 Fuel cell stack performance	10
4.2.2 PEMFC modelling in literature	10
4.3 The turbo-electric generator	12
4.4 Balance of plant components	13
4.4.1 Compressor	13
4.4.2 Heat exchanger and thermal management techniques	14
4.4.3 Humidifier and water management	16
<b>5 Propulsion System Modelling</b>	<b>19</b>
5.1 Purpose of the model	19
5.2 System boundaries & variables	19
5.3 Relevant processes	22
5.4 Assumptions	23
5.5 Conservation laws	24
5.5.1 Conservation of mass	24
5.5.2 Conservation of energy	25
<b>6 Component Modelling</b>	<b>26</b>
6.1 The Fuel Cell	26
6.1.1 Fuel cell analytical model	26
6.1.2 Fuel cell semi-empirical model	28
6.1.3 Fuel cell Model Verification	30
6.2 Balance of Plant components	31
6.2.1 Compressor	31



---

6.2.2	Humidifier	32
6.2.3	Thermal management system	33
6.3	The Turbo-generator	34
6.3.1	Gas turbine simulation program (GSP)	34
6.3.2	GSP output data processing	35
6.4	Preliminary weight estimation	36
<b>7</b>	<b>Case study implementation &amp; Results</b>	<b>38</b>
7.1	Mission analysis with initial design assumption	38
7.1.1	Fuel cell on design model	38
7.1.2	Initial point Off-design mission simulation	41
7.2	Case study and design optimisation	45
<b>8</b>	<b>Conclusions</b>	<b>49</b>
	<b>Acknowledgments</b>	<b>52</b>
<b>A</b>	<b>Saturation vapour pressure VS Temperature table</b>	<b>58</b>
<b>B</b>	<b>eVTOL Data</b>	<b>59</b>
<b>C</b>	<b>Code Files Project</b>	<b>60</b>
C.1	Flight data reader	60
C.2	Model functions	62
C.3	GSP data processing	67
C.4	Propulsion system integration	70
C.5	Main file	77
C.6	Processing data and plotting	88



# NOMENCLATURE

## ABBREVIATIONS

Abbreviation	Definition
UAM	Urban Air Mobility
HPS	Hybrid Propulsion System
ICE	Internal Combustion Engine
FC	Fuel Cell
BED	Battery Energy Density
HF	Hybridisation Factor
EM	Electrical Motor
HHV	High Heating Value
PEM(FC)	Proton Exchange Membrane(Fuel Cell)
HEX	Heat Exchanger
CS	Cooling System
SOFC	Solid Oxide Fuel Cell
DT	Delaunay triangulation
BoP	Balance of Plant
MEA	More Electric Aircraft
rxn	Reaction
abs	absorbed
del	delivered
DEP	Distriputed Electric Propulsion
OPEM	Open Source [PEM] Cell Simulation Tool
MEA	Membrane electrolyte assembly

## SYMBOLS

Symbol	Definition	Unit
$G, g$	Reaction Gibbs free energy	[J, J/mol]
$S, s$	Entropy	[J/K, J/K · mol]
$F$	Faraday's constant	[96,485 C/mol]
$N_A$	Avogadro's constant	[6.02E23 /mol]
$q$	Electron charge	[C / electron]
$j_0$	Exchange current density	[A/cm <sup>2</sup> ]
$j$	Current density	[A/cm <sup>2</sup> ]
$i$	Current	[A]
$n$	Reaction number of electron	[-]
$ASR$	Area Specific Resistance	[Ω · cm <sup>2</sup> ]
$j_L$	Limiting current density	[A/cm <sup>2</sup> ]
$E$	Voltage	[V]
$E_{conc}$	Concentration losses	[V]
$E_{Ohmic}$	Ohmic losses	[V]
$E_{act}$	Activation losses	[V]
$M$	Molar mass	[g/mol]
$v$	Velocity	[m/s]
$T$	Temperature	[K]

Symbol	Definition	Unit
$P$	Pressure	[Pa]
$M$	Mach number	[-]
$c_p$	Specific heat at constant pressure	[kJ/kg · K]
$\dot{m}$	Mass flow rate	[kg/s]
$P_{sat}$	Saturated water pressure	[Pa]
$A$	Area	[m <sup>2</sup> ]
$R_M$	Equivalent Membrane Resistance due to proton conduction	[Ω]
$S$	State variable	[-]
$J$	Jacobian matrix	[-]
$R$	Gas Constant	[J/K · mol]
$V$	Volume	[m <sup>3</sup> ]
$f$	Thrust Force	[N]
$h$	Specific Enthalpy	[J/kg]
$s$	Specific Entropy	[J/kgK]
$U$	Incoming wind velocity	[m/s]
$U_r$	Wind velocity at the rotor	[m/s]
$U_e$	Wake/exit wind velocity	[m/s]
$N1$	Low speed spool number of rotations	[rpm]
$N2$	High speed spool number of rotations	[rpm]
$\alpha$	Charge transfer coefficient	[-]
$\lambda$	Stoichiometric ratio	[-]
$\lambda_{Water}$	Ratio of water moles for each sulfonic group in the MEA	[-]
$\phi$	Relative humidity	[-]
$\rho$	Density	[kg/m <sup>3</sup> ]
$\eta$	Efficiency	[-]
$\rho_M$	Membrane specific resistivity	[Ωcm]
$\kappa$	Specific heat ratio	[-]
$\Pi$	Pressure ratio	[-]
$\Delta$	Change	[-]

## List of Figures

3.1	Electric UAM aircraft categorisation [9]. . . . .	3
3.2	The German eVTOL: lilium. [13]. . . . .	4
3.3	The British eVTOL: Vertical Aerospace VA-X4 [14]. . . . .	4
3.4	A multi rotor eVTOL aircraft concept designed by VOLOCOPTER flying over Tokyo [15]. . . . .	4
3.5	Hybrid versus all electric propulsion system for eVTOL [20]. . . . .	5
3.6	Rolls Royce hybrid eVTOL powered by its M250 turbo-generator [22]. . . . .	5
4.1	An example of an electric series hybrid propulsion architecture [30]. . . . .	7
4.2	An example of an electric parallel hybrid propulsion architecture [5]. . . . .	7
4.3	An example of a series-parallel hybrid propulsion system [5]. . . . .	8
4.4	Fuel cell architecture [37]. . . . .	8
4.5	Fuel cell Turbogenerator hybrid architecture example [37]. . . . .	8
4.6	PEM Fuel cell schematic and operation principle [42]. . . . .	9

4.7	A fuel cell model example showing different levels of fuel cell sophistication levels. Each element of this model either use fundamental or semi-empirical physical equations to explain the fuel cell phenomena [48]. . . . .	11
4.8	The PEM fuel cell spatial modelling domains [49]. . . . .	11
4.9	A conventional turbofan engine [57]. . . . .	12
4.10	Turboelectric propulsion system [59]. . . . .	13
4.11	A centrifugal compressor air supply feeding scheme [60] . . . . .	13
4.12	An example of a typical radial compressor performance map [4]. . . . .	14
4.13	Different cooling strategies for the PEMFC [67] . . . . .	15
4.14	Evaporatively cooled PEMFC [69] . . . . .	15
4.15	Liquid cooled PEMFC [69] . . . . .	16
4.16	Gas bubbling humidifier schematic diagram [73] . . . . .	17
4.17	Liquid water injection humidification method [73] . . . . .	17
4.18	Enthalpy wheel humidifier Schematic diagram [73] . . . . .	17
4.19	Gas to gas membrane humidifier schematic diagram [73] . . . . .	17
4.20	Liquid to gas membrane humidifier schematic diagram [73] . . . . .	17
5.1	eVTOL flight mission and power requirement versus time. . . . .	19
5.2	Simplified flow diagram example of the hybrid propulsion system. . . . .	20
5.3	System components and associated variables . . . . .	21
6.1	An example of the PEM fuel cell polarization and power density curves. The initial part of the j-V curve is dominated by the activation losses. The middle linear like shape is dominated by the Ohmic losses. Concentration losses shapes the end of the curve since it dominates at high currents. . . . .	27
6.2	The fuel cell polarization curve with varying operating temperature. . . . .	30
6.3	The fuel cell polarization curve with varying hydrogen partial operating pressure. . . . .	30
6.4	The fuel cell polarization curve with varying oxygen partial operating pressure. . . . .	31
6.5	The fuel cell polarization curve with varying water content parameter. . . . .	31
6.6	GSP model of a turbo-shaft [85]. . . . .	34
6.7	Flight envelope including the series data points used as an input for GSP . . . . .	34
6.8	The Flight/Ambient condition tab sheet with the altitude, Mach number and limits specified. . . . .	34
6.9	The Power/Control settings tab sheet with the turbine inlet temperature values presented. . . . .	34
6.10	Shaft power output vs fuel mass flow rate for altitudes 0 to 3000 m. The mach number is kept constant to show the effect of increasing the altitude. . . . .	35
6.11	Shaft power output vs fuel mass flow rate for 0 m altitude. The mach number is varied to show the effect it has on the line trend. . . . .	35
6.12	Left: barycentric coordinates in 2D and 3D planes. Right: Delaunay triangulation in 2D [86]. . . . .	36
6.13	The PEM fuel cell in an exploded view [87] . . . . .	36
7.1	The fuel cell operating point during cruise at 53% voltage efficiency . . . . .	39
7.2	Fuel cell total power versus the fuel cell voltage efficiency. . . . .	39
7.3	The loop used to calculate the required fuel cell area for the on design model . . . . .	40
7.4	Detailed power division of the hybrid propulsion system during take off and descend, including the balance of plant components power demands. . . . .	42
7.5	Detailed power division of the hybrid propulsion system during approach and emergency mission, including the balance of plant components power demands. . . . .	42
7.6	Species mass flow rates during the eVTOL take-off, descent and the corresponding current variation. . . . .	43
7.7	Species mass flow rates during the eVTOL approach, emergency mission and the corresponding current variation. . . . .	43
7.8	eVTOL flight mission power requirement HPS fuel cell & gas turbine efficiencies. It should be noted that efficiencies are zero when the corresponding power requirement is zero. . . . .	44
7.9	Fuel cell voltage variation with respect to the flight mission. Voltage is taken as zero when the power requirement is zero. . . . .	44
7.10	The resultant fuel cell gross power when designing the fuel cell at voltage efficiencies of 55% and 64%. . . . .	45

7.11	Turbogenerator power when operating with fuel cell at voltage efficiencies of 55% and 64%. . . .	45
7.12	Number of cells at different design points. . . . .	46
7.13	Fuel cell at different design points. . . . .	46
7.14	Fuel cell, turbogenerator mass and the resultant free payload mass. . . . .	47
7.15	Power division of the fuel cell, compressor, cooling system and turbogenerator at different design points. . . . .	47
7.16	The compressor and cooling system weights at different design points. . . . .	48
7.17	Hydrogen consumed and the corresponding tank weight. . . . .	48
8.1	Simplified flow diagram example of the hybrid propulsion system. . . . .	50

## List of Tables

3.1	Battery powered aircraft vs hybrid aircraft . . . . .	5
5.1	Hybrid propulsion system key variables . . . . .	20
6.1	Summary of variables and materials used to estimate the fuel cell stack mass . . . . .	36
6.2	Relevant densities used in preliminary estimate the HPS weight . . . . .	37
7.1	The fuel cell design point requirements used to estimate the needed number of cells and fuel cell area. . . . .	38
7.2	On design characteristics at 53 % voltage efficiency . . . . .	41
7.3	Component weight variation at compressor efficiencies of 70% and 85% . . . . .	48
8.1	The hybrid propulsion system weights and the resultant maximum payload weight . . . . .	51
A.1	Saturated vapour pressure of water at selected temperatures [96] . . . . .	58
B.1	eVTOL aircraft mission data used for the case study . . . . .	59

# 1

## INTRODUCTION

Year 2023 has witnessed the end of global warming and the start of the **global boiling** era, says the UN secretary general, António Guterres<sup>1</sup>. According to the 2022 EASA report [2], aviation emissions contributed to 3.7 % of the overall global warming. This is enough to point out that taking an action and developing sustainable technologies is a must for the aviation industry.

Electrical propulsion systems can be the answer to the aviation industry sustainability problem and a fuel cell turbogenerator hybrid propulsion system studied in this thesis could be a choice of a potential eVTOL customer. At AYED-ENGINEERING, it is planned to provide a turbogenerator, capable of providing electric power to an eVTOL or a small aircraft in general. Using turbogenerators enables the use of multirotor configuration and also enables vertical take-off and landing for eVTOLs, while maintaining a long cruise flight capability. However, this can be done with a kerosene turbogenerator which is not sustainable. Hydrogen can be the solution for this issue, but it will introduce another problem. Turbogenerators are very low in efficiency, which means it consumes a lot of fuel which is manageable if kerosene was the fuel. However, if hydrogen was used, more fuel means heavier tanks due to hydrogen low energy density. An answer to this problem is coupling the turbogenerator with a fuel cell, which is more efficient if used for long cruise flights. Consequently, the aim of this thesis.

The primary goal of this research is to develop a tool for a hybrid propulsion system, specifically designed for eVTOL aircraft. This model is intended to preliminary predict the mass and performance of the propulsion system components, which include a turbo-generator and a fuel cell. By simulating different flight scenarios and power demands, the model provides essential insights into how these components can be preliminary sized.

This study is motivated by the pressing need for sustainable aviation solutions as the aviation sector faces increasing pressure to reduce its environmental footprint. As climate change intensifies, with significant contributions from aviation emissions, the development of greener propulsion technologies becomes critical. The hybrid electric propulsion system explored in this thesis represents a step towards achieving significant reductions in emissions and energy consumption for UAM applications.

The structure of this thesis is designed to guide the reader through a detailed exploration of hybrid propulsion system modeling and its implications for eVTOL aircraft. In the beginning, [chapter 2](#) outlines the research goals and questions, focusing on the creation of a performance model capable of estimating the mass of the propulsion system components. Afterwards, [chapter 3](#) offers a comprehensive review of eVTOL technology, including basic concepts and current market trends, and discusses the advantages and limitations of existing propulsion systems.

Subsequent chapters delve into the specifics of hybrid propulsion system components and architectures. Detailed examination of various hybrid propulsion architectures, highlighting their respective benefits and potential applications in eVTOL aircraft is presented in [chapter 4](#). Moreover, [chapter 5](#) details the development of the performance model, including a discussion of the relevant processes, system boundaries, and key variables essential for accurate mass estimation. An in depth analysis is then provided in [chapter 6](#) where the propulsion system components are modelled, focusing on fuel cells and turbo-generators, and outlining the methodologies used for mass estimation. The model is then used on a case study in [chapter 7](#), demonstrating its utility in estimating the mass and performance of a hybrid propulsion system for an eVTOL aircraft. Finally, [chapter 8](#) synthesizes the findings and offers conclusions and recommendations for future research.

By meticulously modeling and evaluating the mass and performance of hybrid electric propulsion systems, this thesis aims to contribute to the development of sustainable aviation technologies and support the next generation of eVTOL aircraft.

---

<sup>1</sup><https://www.theguardian.com/science/2023/jul/27/scientists-july-world-hottest-month-record-climate-temperatures>

## RESEARCH GOAL & QUESTIONS

### 2.1. RESEARCH GOAL

The main focus of this thesis is to **develop a performance calculation model of a hybrid propulsion system consisting of a turbo-generator and fuel cell (TG-FC)**. The developed model should provide preliminary component weights for a given flight mission. A fuel cell is more efficient, greener and produce less noise when compared to combustion engines, but it has a lower power density than a gas turbine[3, 4]. This introduces the challenge of using fuel cells solely in powering manned aircraft. Based on this, *hybrid propulsion is introduced as a promising alternative to fill the deficiency of the fuel cell power density.*

### 2.2. RESEARCH QUESTION(S)

Based on the research goal stated in the previous section, the following main research question is proposed :

***How can a hybrid propulsion system consisting of a fuel cell and a turbogenerator be modelled and used to assess the performance of an eVTOL aircraft during its different flight phases ?***

The following structure will help in formulating the sub questions. Fuel cells such as PEMFCs are efficient when compared to turbo-generators, but are relatively very low in their power densities, where a 1 kg fuel cell generates a maximum of 1.1 kW [3].

A fuel cell system requires other components in order to operate efficiently, such as the compressor and cooling system. This adds more weight to an already heavy fuel cell. In aerospace applications, weight is a top priority and any decrease in the empty weight adds to the efficiency of the vehicle. So, the following sub-question can be formulated:

How can the developed tool be used to make a preliminarily estimate of the weight of the hybrid propulsion system components?



## EVTOL & eSTOL APPLICATIONS

This chapter presents a general overview of the current urban air mobility (UAM) vehicle developments in the market and presents the current performance of flight missions with different propulsion systems.

### 3.1. BASIC CONCEPTS & CURRENT MARKET

In literature electric Urban Air Mobility (UAM) vehicles are divided into two types, electric vertical take off and landing aircraft (eVTOL) or electric short take off and landing aircraft (eSTOL) [5, 6]. The main difference between the two concepts is that an STOL aircraft requires a runway of some length for the take off and landing phases. M. Burton et al. in their STOL aircraft feasibility study [6] argues that STOL aircraft has some advantages over VTOL. In their arguments, they stated that STOL aircraft due to its fixed wing configuration requires less power to take off, resulting in a lighter structure that is able to deliver higher payloads, longer ranges or results in a smaller aircraft [6]. However, it is important to note that it is expected from these aircraft to operate in a congested, densely populated metropolitan areas where availability of infrastructure is severely limited. So, despite the advantages of the STOLs compared to the VTOLs, this is considered STOL aircraft Achilles heel, since the main point of these aircraft existence is its low amount of infrastructure required for its operation [7, 8].









	Rotor Lift	Actuating Combined Lift	Static Combined Lift	Wing Lift
<b>Few Propulsors (1-3)</b>	 Helicopter	 Tilt-lift	 Static Hybrid	 Fixed Wing
	Powered Lift			
<b>Many Propulsors (4+)</b>	 Multirotor	 DEP Tilt-lift	 Stopped Rotor	 DEP Fixed Wing
	Multirotor	DEP Powered Lift		Fixed Wing

Figure 3.1: Electric UAM aircraft categorisation [9].

The UAM market is full of various players such as Airbus, Boeng Aurora flight sciences, NASA, Joby, Lilium, Vertical Aerospace, Hyundai Supernal, Volocopter, EHang, Kitty Hawk, and Archer aviation Midnight [10, 11]. Their different propulsion architectures can be categorized into four groups as shown in Figure 3.1 [9, 8, 12]: The fixed wing design, a distributed electric propulsion (DEP) powered lift, multi rotor and rotorcraft designs. The configurations shown in Figure 3.1 are based on the how the vehicle generates lift which can be by the wings, rotors or a combination of both [9]. Rolls Royce ACCEL and NASA-X57 are an example of fixed wing electric aircraft [7]. An example of an electric rotorcraft is Aquinea Volta light helicopter [7]. Two promising DEP tilt-lift eVTOL example designs are the 5/7 seater Lilium jet and the 5 seater VA-X4 as respectively shown in Figure 3.2 and Figure 3.3. A multi rotor eVTOL example design is the Volocopter shown in Figure 3.4.

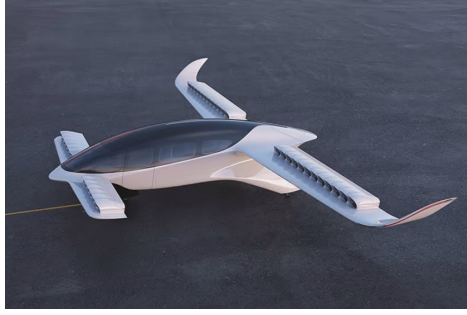


Figure 3.2: The German eVTOL: lilium. [13].



Figure 3.3: The British eVTOL: Vertical Aerospace VA-X4 [14].



Figure 3.4: A multi rotor eVTOL aircraft concept designed by VOLOCOPTER flying over Tokyo [15].

### 3.2. VEHICLE PERFORMANCE

Most of the eVTOLs use batteries as their main source of energy [16]. However, battery energy density with comparison to jet-A fuel is very low. Based on a study made by Berger [12], battery technology used in the automotive industry has a specific energy density between 250 and 300  $Wh/kg$ , while jet-A fuel has an energy density of approximately 11900  $Wh/kg$  (43  $MJ/kg$ ) [10]. PNathan et al. [17] in their lilium performance assessment report used a slightly higher battery energy density of 320  $Wh/kg$  which yielded a maximum range of 261  $km$ . This was increased to 330  $km$  if a battery specific energy density of 400  $Wh/kg$  was used [17].

It is important to note that the specific energy density of the battery has a big impact on the performance of the aircraft. By using the modified Breguet equation for electric aircraft shown in Equation 3.1 [18], one can conclude that the battery energy density  $e_{battery}$ , battery mass  $m_{battery}$ , total propulsive efficiency  $\eta_{total}$  and the lift to drag ratio has a direct impact on the range and the endurance in result.

$$R = \frac{m_{battery}}{m_{total}} \cdot \frac{L}{D} \cdot \frac{1}{g} \cdot e_{battery} \cdot \eta_{total} \quad (3.1)$$

It should be pointed out that the battery density affects the maximum take off weight and the payload. The lower the specific density the lower the battery mass and the more payload one can take for a constant maximum take-off weight. To take an example of how the payload is affected by the battery energy density (BED) from the market, the Bartini Flying car has a 400  $kg$  payload capacity and a BED of 200  $Wh/kg$  whereas the eHang 184 has 100  $kg$  payload capacity and a BED of 140  $Wh/kg$  [16, 19]. Fuel cells can be used instead of batteries. It is important to note that hydrogen has a low gravimetric and volumetric energy density ranging between 2.5 - 10  $MJ/kg$  and 2 - 20  $MJ/L$  respectively [3].

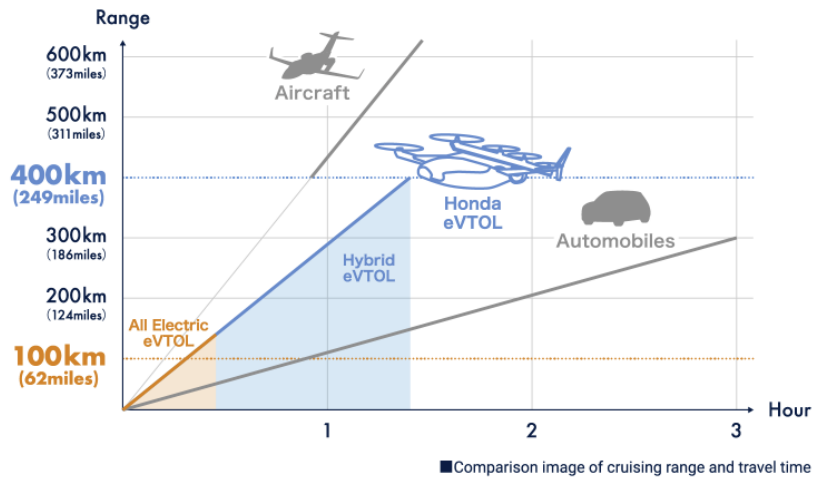


Figure 3.5: Hybrid versus all electric propulsion system for eVTOL [20].

Using a gas powered generator as an assistance in a hybrid system improves the overall performance of the whole aircraft, since it increases the overall specific energy [21]. As a result of such increase, the design boundaries widen allowing for heavier payloads and longer ranges as demonstrated by Honda in Figure 3.5 [20]. To demonstrate with an example from the market, the Rolls Royce hybrid eVTOL shown in Figure 3.6 uses its M250 gas turbine along with its batteries onboard to achieve a cruise speed of 402.3 km/hr and a range of 804.7 km [16, 21]. It is able to perform this performance with 5 passengers on board [16]. The M250 C20 has a specific power of 4.35 kW/kg and it can use liquid fuels with higher specific energies when compared to batteries [21].



Figure 3.6: Rolls Royce hybrid eVTOL powered by its M250 turbo-generator [22].

It can be concluded from this chapter that the UAM market is full of companies such as Lilium and Vertical Aerospace. Most of these aircraft are battery powered and they can not perform ranges longer than 300 km. Hybrid vehicles such as the Rolls Royce use a turbo generator beside the battery allowing it to reach 804.2 km. An example between battery powered aircraft Lilium and Rolls Royce eVTOL in terms of ranges and cruise speeds is summarised in Table 3.1. In the next chapter, the focus will be given to a hybrid propulsion system that consists of a fuel cell and a turbo generator such that the aircraft can cover longer range or endurance.

Performance review			
Aircraft type	Power source(s) & (energy density) [Wh/kg]	Range [km]	Cruise Speed [km/hr]
Lilium	Battery (400)	330	300
Rolls Royce eVTOL	Hybrid turbogenerator (11900) & battery (320)	804.7	402.3

Table 3.1: Battery powered aircraft vs hybrid aircraft

# 4

## HYBRID PROPULSION SYSTEM

A full electric propulsion system has the potential to mitigate the direct  $CO_2$  pollution and other air pollutants such as nitrogen oxide compounds  $NO_x$ . However, looking at the vehicle power source, Lithium  $Li$  ion batteries, one can see that the energy density is too low to deliver high power for longer duration. To have a feeling of the numbers, the best available lithium ion battery today has a specific energy of around 250 and 300  $Wh/kg$  with an 80% packing efficiency, while a short-range electric aircraft requires a minimum of 750  $Wh/kg$  as a battery pack specific energy [23, 10]. Hybrid propulsion is considered a green transition method to cover the power demand and improve the performance of the aircraft.

This chapter will address an overview of the most common hybrid engine architectures in [section 4.1](#). In [section 4.2](#), a more in details explanation on the fuel cell turbo generator hybrid propulsion system will be given since it is the focus of this thesis

### 4.1. HYBRID PROPULSION ARCHITECTURES

There are many types of hybrid propulsion systems that are currently in use or under development. The term hybrid in propulsion is generally used when two or more power sources are used on board of the vehicle. From literature, aircraft hybrid propulsion systems can be categorized into three main architectures: series, parallel or a complex mix of both serie and parallel architectures [24, 25, 26].

Before diving into the different hybrid configurations, it is useful to introduce the degree of hybridization (DoH), or the hybridization factor (HF), to determine how much power is provided by each component [27]. Hybridisation factor is defined as the ratio between the power delivered by the electric motor to the maximum total power of the combined system [28]. Isikveren et al. and Lorenz et al. in [29, 27] had the definition more detailed with respect to power  $HF_P$  and energy  $HF_E$  as shown in [Equation 4.1](#) and [Equation 4.2](#) respectively. To elucidate the use of the two definitions, a conventional kerosene based aero-engine has both terms  $HF_P$  and  $HF_E$  equal to 0. However,  $HF_P$  and  $HF_E$  are equal to 1 in the case of a fully electric vehicle with an electric energy storage such as batteries. A hybrid electric propulsion with only electric power, but a kerosene based energy storage has  $HF_P$  and  $HF_E$  of 1 and 0 respectively.

$$HF_P = \frac{P_{EM}}{P_{tot}} \quad (4.1)$$

$$HF_E = \frac{E_{EM}}{E_{tot}} \quad (4.2)$$

#### 4.1.1. SERIES POWER-TRAIN ARCHITECTURE

As shown in [Figure 4.1](#), a reference series configuration is given, where the propeller or the thrust generating unit is driven by an electrical motor, which takes its power from two different energy sources. The engine in this configuration is mechanically decoupled from thrust generation, which has a benefit in terms of engine-generator location flexibility. Also, the engine in this configuration will have the advantage in operation at its ideal operating power and speed in different operating phases [30, 5]. On the other hand, this might introduce a heavy and thus an expensive propulsion devices such as the electric motor which drives the propulsion device on its own [31, 32]. Moreover, despite the simplicity of the system, it is not very efficient, because of the power losses in combustion and electrical energy conversion [30]. The system can be operated in different modes [33]:

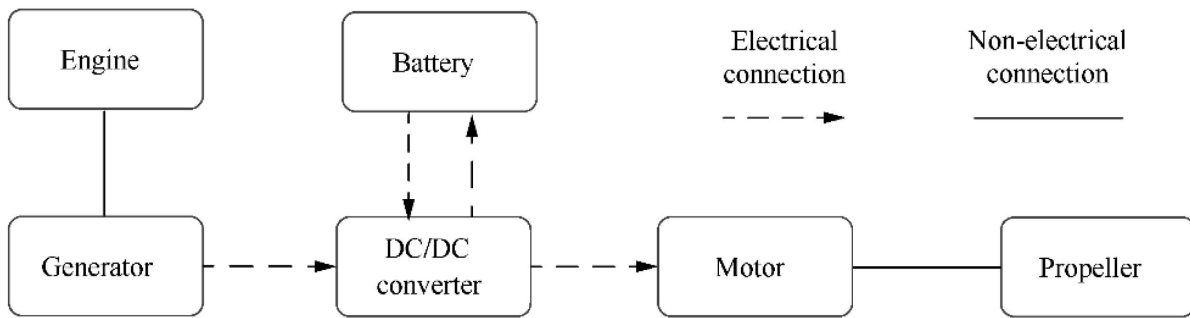


Figure 4.1: An example of an electric series hybrid propulsion architecture [30].

1. Only the battery powers the propulsion device.
2. The engine and the generator power the electric motor.
3. Both the engine and generator provide power to the electrical motor.
4. power split mode, in which the power from the engine and generator is divided between charging the battery and powering the electrical motor.

#### 4.1.2. PARALLEL POWER-TRAIN ARCHITECTURE

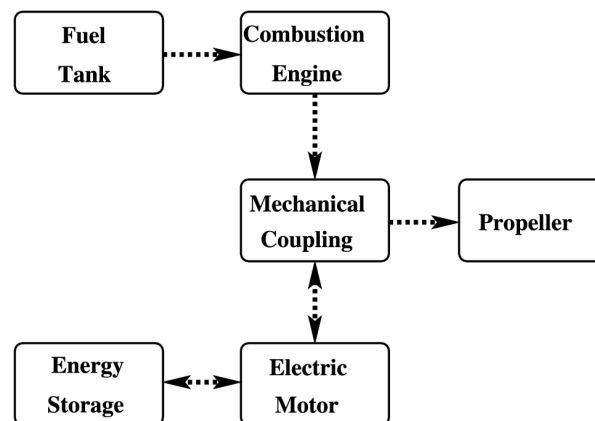


Figure 4.2: An example of an electric parallel hybrid propulsion architecture [5].

Parallel propulsion systems has the thrust unit driven either individually or simultaneously by the engine and the electrical motor as shown in Figure 4.2 [5]. With comparison to the series system, the engine and the electrical source in this configuration is mechanically coupled to a shaft which drives the propeller [25]. This configuration, apart of the added mechanical coupling, might introduces a lighter and compact propulsion system when compared to other configurations, since the propelling power is provided directly by both units [33, 34]. Moreover, in case the electric energy source was a battery, the engine can simultaneously charge the battery and drives the propeller through the mechanical coupling. the electrical motor in this case works as an electrical generator.

The engine in this configuration however can not operate optimally in different flight phases because of the limited degree of hybridisation [35]. Additionally, such system requires a sophisticated propulsion control system [5]. The system can be operated in the same manner as the series configuration except the difference here is that the engine and the EM directly drives the propeller and both power units are coupled [33].

#### 4.1.3. SERIES-PARALLEL POWER-TRAIN ARCHITECTURE

The series parallel configuration is a combination of both series and parallel and it is used in some hybrid vehicles, because of the easy switch between both configurations during operation. As shown in Figure 4.3, the system has an extra generator compared to the parallel configuration and an extra mechanical link when



compared to the series configuration [5, 33]. The configuration in general combines the advantageous features of both systems, but at the expense of increasing the system complexity and cost, making it the least popular configuration for aircraft application [30].

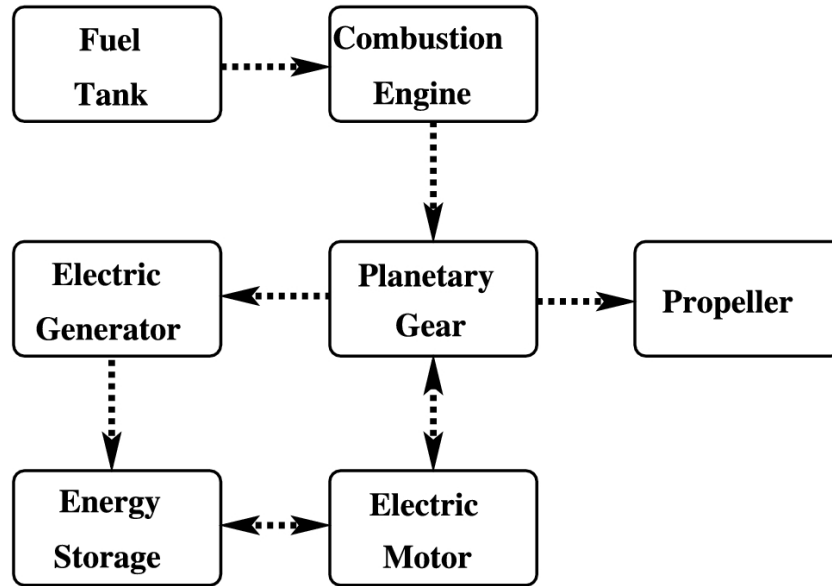


Figure 4.3: An example of a series-parallel hybrid propulsion system [5].

It can be concluded that the series configuration for aircraft application is considered the least complex, the easiest to design, maintain and control when compared to other configurations. Parallel systems on the other hand are more compact and more energy efficient. Moreover, T. S. Dean et al. [36] found out that the parallel hybrid configuration can provide greater range performance than the series hybrid. The series-parallel system combines the benefits of both but due to complexity its the least popular configuration for aircraft application.

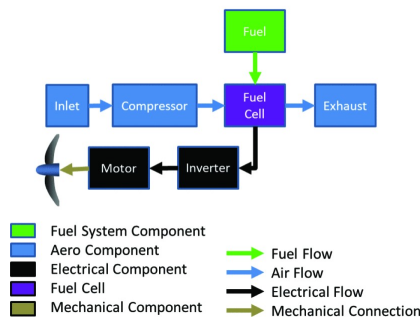


Figure 4.4: Fuel cell architecture [37].

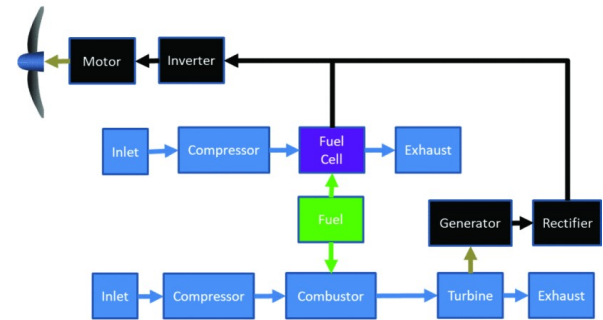


Figure 4.5: Fuel cell Turbogenerator hybrid architecture example [37].

Figure 4.4 shows the propulsion system architecture in series if fuel cells are solely producing power. However, this is not optimal because of their low power density, so a turbo-electric system is added as shown in Figure 4.5.

The architecture illustrated in Figure 4.5 is a series architecture and it will be used to base the model in the following chapters. In the following sections the fuel cell, turbo-electric generator and BoP used in these architectures will be discussed in detail.

## 4.2. FUEL CELLS

Fuel cells are an electro-chemical energy conversion devices, which share some characteristics in common with combustion engines [38]. In combustion, chemical energy is converted to heat which can be converted

to mechanical energy and then electricity. Going through more than one conversion process leads to loss of energy and thus lower efficiency which is why fuel cells are considered more efficient than combustion powered engines. That said, it is worth noting that fuel cells do not have any moving parts, resulting in a reliable, sustainable and silent propulsion unit [3]. Harmful emissions such as  $NO_x$  and  $SO_x$  are also omitted compared with other devices [38, 3].

On the other hand, PEM fuel cell has a lower volumetric and gravimetric power density when compared with IC engines, which means it produces less power per unit volume or mass, respectively [3]. Using hydrogen as fuel, which is preferred to fuel cells like the PEMFCs, adds to the fuel cell limitations, since hydrogen has a very low volumetric energy density compared to other fuels (e.g. gasoline) and it is difficult to store [38, 3].

Fuel cells are characterised by their electrolyte, and there are five major types: **Phosphoric acid (PA), polymer electrolyte membrane (PEM), alkaline, molten carbonate (MC) and solid oxide (SO)** [3]. For aerospace applications, SOFCs and PEMFCs are considered promising due their high efficiency and power density [39, 40]. However, it is worth noting that depending on the application, one fuel cell can have an edge on the other. SOFCs can operate at high temperatures up to  $1000^\circ C$ , which might be an advantage or a disadvantage depending on the thermal energy utilisation method [40]. They have higher power density and versatile as well when it comes in using different fuels other than hydrogen, such as hydrocarbon based fuels [40]. On the other hand, PEMFCs operate at lower temperatures up to  $100^\circ C$ , which is a plus, since it would require less maintenance and introduce wider range of operations [40]. In addition, PEMFCs have a faster start up, and better compactness which gives it an edge on SOFCs when it comes to aircraft applications [40, 41]. That said, it is decided to continue the thesis research using PEMFCs.

The reaction taking place in a fuel cell is divided into two electro-chemical half reactions as shown in Equation 4.3 and Equation 4.4. In Equation 4.3, hydrogen is oxidised by losing its electrons, while oxygen is reduced by gaining two electrons as shown in Equation 4.4. Both reactions are combined into Equation 4.5 which is the overall reaction. It is worth noting that the overall reaction is exothermic, because of the creation of new bonds in water which releases more heat than the required heat to break the hydrogen bonds.

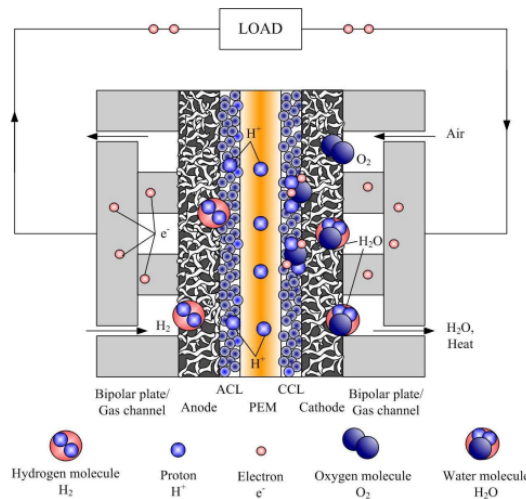


Figure 4.6: PEM Fuel cell schematic and operation principle [42].

Figure 4.6 gives an overview of the operational scheme of a PEM fuel cell. The figure also shows the fuel cell different components: the polymer electrolyte membrane (PEM), Bipolar plates, diffusion and catalyst membrane layers at both the anode and cathode sides. The PEM component main function is to attract

hydrogen protons and repel electrons so they are forced to go around the outer circuit and create an electrical load. Usually, the material used for the PEM is Nafion, which requires sufficient humidity for an efficient proton conductivity [38, 42, 4]. As will be discussed later in subsection 4.2.1, excess water can flood the cathode catalyst layer and stop hydrogen and oxygen from getting across, consequently affecting the fuel cell performance. Therefore, it is important to have the fuel cell hydration levels managed during the fuel cell operation.

As mentioned earlier, PEM fuel cells have a low operation temperature with a maximum of 100°C, which might cause a slow reaction especially at the cathode side [42, 4]. Platinum catalyst layers come here into play to speed up both reactions at both ends anode and cathode. It is worth mentioning that these layers are prone to carbon oxide poisoning if alcohol or hydrocarbons were used for hydrogen production [42]. Moreover, the gas diffusion layer (GDL) is a porous material that supports the catalyst layer and helps with the heat and water management within the cell [42, 4].

In the next two sub-sections, the performance of the fuel cell under different conditions and how it is modelled in literature are discussed in detail.

#### 4.2.1. FUEL CELL STACK PERFORMANCE

The fuel cell is expected to operate at high altitudes and to be the main power source of the aircraft during cruise. At high altitudes, operating parameters such as the pressure, the fuel cell temperature, reactant gases relative humidity and air stoichiometric ratio will be affected. Thus it is important to understand the effect of these operating parameters on the fuel cell behaviour and performance.

The theoretical maximum fuel cell voltage is 1.23 V. The maximum reversible voltage is subjected to irreversible losses arising from three main sources: Activation losses, Ohmic losses, and concentration losses. Operating the fuel cell at higher pressure and temperature, improves the overall performance of the fuel cell. Activation losses decrease at higher temperatures because of the improved kinetics at both electrodes especially the cathode since oxygen kinetics are sluggish at low temperatures [3]. Moreover, increasing the pressure increases the reactant gas partial pressures and the concentration of the reactant gasses at both electrodes. This improves the gas kinetics and thus the exchange current density and minimises activation and concentration losses. It is important to note that using air instead of oxygen introduces kinetic penalties because of the reduction of oxygen kinetics at the cathode. The ideal air flow stoichiometric rate  $\lambda$  found in literature is 2 [4], which insures that the fuel cell is supplied with enough oxygen at high current densities.

A well hydrated electrolyte membrane, allows a facile movement of hydrogen ions resulting in a high performance fuel cell. This means that the conductivity of the membrane governs the PEMFC performance [43]. The conductivity of the membrane is a linear function of its hydration state [43]. There are several complications such as electro-osmotic drag and operation at high temperature [4]. Electro-osmotic drag is a process where hydrogen ions drag water molecules with them during moving from the anode to the cathode, resulting in drying out of the anode side [4]. Moreover, when operating at temperatures over 60 °C, the air dries the electrodes at a faster rate than water production at the fuel cell which declines the fuel cell output performance [4, 44]. On the other extreme, the air might get very water saturated, and not able to dry the fuel cell causing flooding which is also not optimal. Therefore, humidity should be kept above about 80% to prevent excess drying and below 100% or otherwise the electrodes will flood blocking the gas reactants [4].

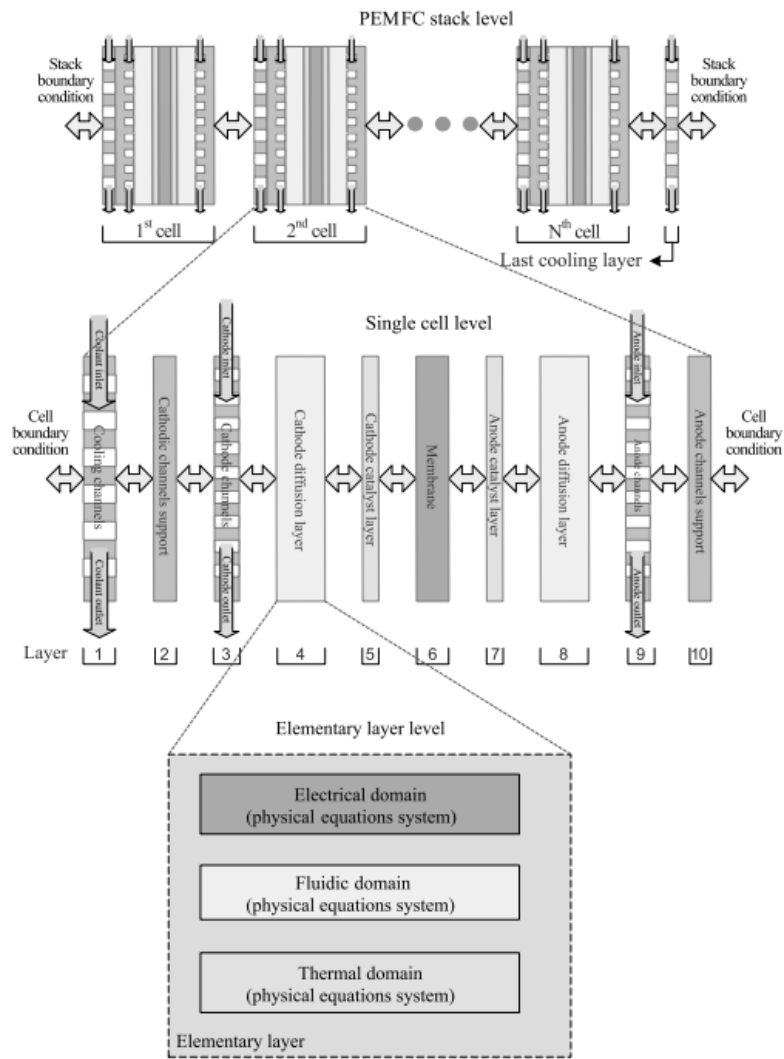
An experimental study made by Wang et. al [45] concurred with all the aforementioned effects. It shows that the PEM fuel cell performance increases by increasing the operating pressure and temperature [45]. Increasing the temperature increases the performance as long as the fuel cell is well humidified and the gas stream humidification temperature is higher than the operating temperature [45].

#### 4.2.2. PEMFC MODELLING IN LITERATURE

The state-of-the-art PEMFC modelling is extensively covered in the literature. The models in general differs in their method, dimension and modelling aim [46, 47]. This section presents how PEMFC are modelled in literature. .

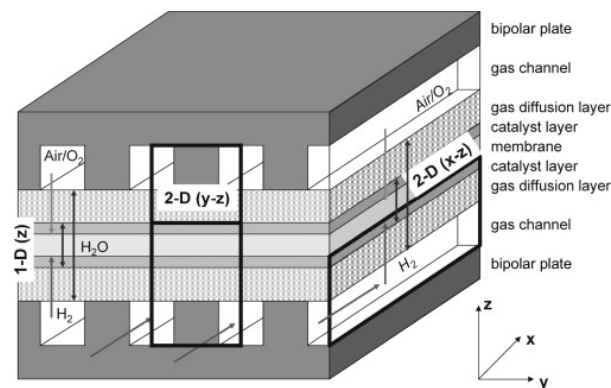
The main goal of modelling the fuel cell is to have an adequate understanding of the fuel cell behaviour under different operation schemes. The model can be as simple as a single cell model that simulates the influence of current load on the voltage drop and in result the effect on the fuel cell power output. However, this might not be enough to evaluate certain phenomena such as fuel cell humidity effect on the cathode or anode catalyst layers. The model can get very sophisticated to try and explain the behaviour of the fuel cell system as shown in Figure 4.7. Most of the models used in literature are of the grey box type or the semi-empirical type, since they do not model every aspect and phenomena occurring within the fuel cell and





**Figure 4.7:** A fuel cell model example showing different levels of fuel cell sophistication levels. Each element of this model either use fundamental or semi-empirical physical equations to explain the fuel cell phenomena [48].

instead they use empirical parameters based on experiments to calibrate their model [48].



**Figure 4.8:** The PEM fuel cell spatial modelling domains [49].

The fuel cell model can be classified based on its space dimension as shown in Figure 4.8, temporal behaviour such as static or dynamic, type such as analytical, semi-empirical or empirical, modelled areas and

modelled phenomena [48]. The space dimension depends on the modelling axis used in the model and it can be 0D, 1D, pseudo 2D, 2D or 3D. The most used ones in literature are the 1D and pseudo 2D models [48]. The 0D models are the ones that predict the polarization curve without the need of any spatial information, which will be further discussed later. The 1D and the pseudo 2D give a more detailed description on a single spatial axis aligned with the direction of the gas diffusion axis. The pseudo 2D is the same as the 1D but it also describes fluid phenomenal aspects on the channel direction axis [48]. The 2D and 3D are more sophisticated and time consuming and they are commonly used for scientific modelling purposes [48, 3].

The physical phenomena of the fuel cell can be represented by solving a set of governing equations, which are the conservation of mass, momentum, energy, species and current transport equations [48, 49]. Springer et al. [50, 51], Bernardi et al. [52, 53] and Amphlett et al. [54] have used these set of equations in their work to create a static 1D PEM model. Amphlett et al. have also modelled the PEM as a 0D model in both static [55] and dynamic states [56]. Bernardi et al. [52, 53] equations did not depend on empirical relations and they are considered analytical. The other models were using semi-empirical equations in their modelling approach.

### 4.3. THE TURBO-ELECTRIC GENERATOR

The fuel cell system is combined with a turboelectric generator to assist with the power demand at different phases of flight such as take off. Nowadays, the majority of civil transport aircraft are powered by a high bypass ratio turbo fan engine, as illustrated in Figure 4.9. The fan of that engine is mechanically driven by the core engine turbine which utilizes the fuel stored chemical energy. This configuration has a specific power range of 3-8  $kW/Kg$ , which is much larger compared to fuel cell specific power range of 0.8 - 1.1  $kW/Kg$  [3, 57].

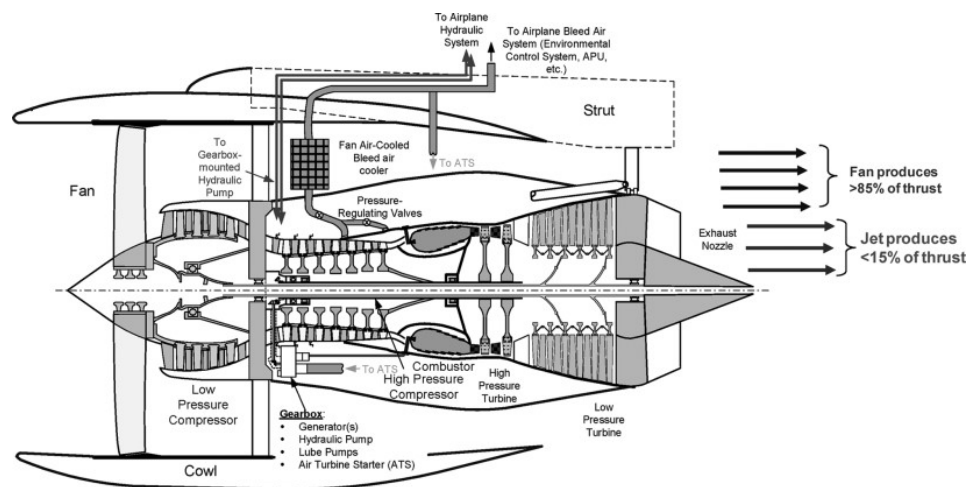


Figure 4.9: A conventional turbofan engine [57].

However, to achieve a quieter, more efficient and lower emissions aircraft, this propulsion system should be electrified. One way of doing this is illustrated in Figure 4.10, where the gas turbine core is mechanically coupled to a generator whereas the propeller device is electrically connected to the generator through an electrical gearbox [57]. This configuration does introduce a lighter and efficient system, since the generator can handle higher speeds than the fan which results in a much faster LP (Low pressure) shaft, and consequently fewer LP stages [57]. It is also worth noting that the speed and torque of that system are decoupled, offering choice flexibility between on and off design performances [57]. This configuration is also well suited for the current trend of transitioning to MEA (More Electric Aircraft) [58].

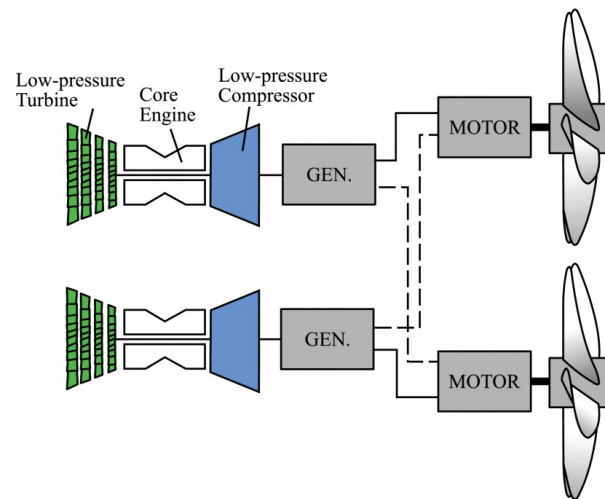


Figure 4.10: Turboelectric propulsion system [59].

In summary, a turboelectric generator is simply a gas turbine in which its shaft is connected to a generator to produce electricity. It converts mechanical energy into electrical energy.

#### 4.4. BALANCE OF PLANT COMPONENTS

In this section, the balance of plant components (BoP) used to assist the fuel cell in its operation are introduced and covered in detail. The balance of plant components considered are the compressor, the cooling system and the humidifier.

##### 4.4.1. COMPRESSOR

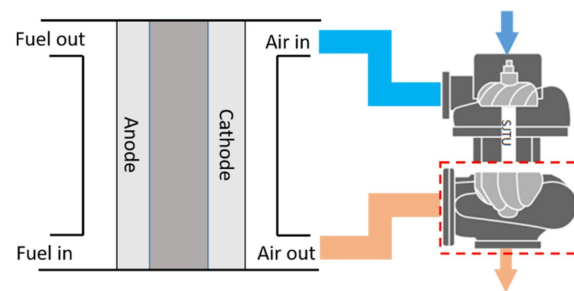


Figure 4.11: A centrifugal compressor air supply feeding scheme [60]

The compressor used by the fuel cell system is used to pressurise the air at high operational altitudes, as shown in Figure 4.11. It will be discussed later in chapter 6, that fuel cell efficiency does improve with increasing the operating pressure and vice versa. Hydrogen will be pressurized because of the high pressurized tanks whereas the ambient air at high altitudes will not, resulting in a decrease in the fuel cell performance from the cathode side. The compressor should be able to provide the fuel cell with at least  $0.25 \text{ MPa}$ , which is the preferred fuel cell operating pressure [60]. The high operating pressure improves the reaction kinetics within the fuel cell and enhance the water management within the fuel cell enabling the use of self-humidification [61] or a less complicated humidification system [60]. J. Hou, et al. [60] have made an extensive research on fuel cell compressors design and control strategies and they found out that centrifugal compressors are considered a good option in terms of the compression ratio, performance map, weight and cost. They does however have limitations such as surge and choke lines, which can be regulated by modifying an adequate control strategy.

The performance of the compressor can be assessed using a compressor map as shown in Figure 4.12. The pressure ratio on the y axis is driven by the fuel cell air inlet pressure requirements. The x axis has mass flow rate corrected to to the altitude of operation. For simplicity reasons, it will be assumed that the pressure ratio

will be constant throughout the mass flow rate and altitude spectrum.

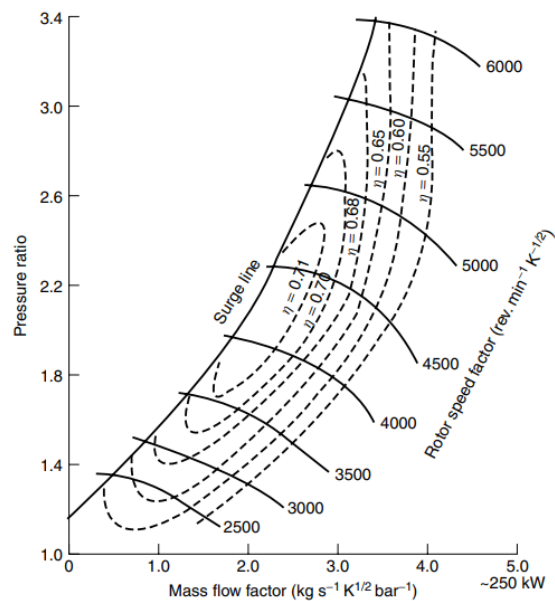


Figure 4.12: An example of a typical radial compressor performance map [4].

#### 4.4.2. HEAT EXCHANGER AND THERMAL MANAGEMENT TECHNIQUES

A heat exchanger working principle is thermal energy (enthalpy) transfer between two or more mediums – can be a fluid, a solid surface or a solid particulates – at different temperatures and in contact [62].

Around 50% of the chemical energy stored in hydrogen is converted into heat during the PEM fuel cell operation [4, 42]. Therefore, dissipation of heat is of great importance to ensure the best fuel cell performance. It is worth noting that the favorable operating temperature for the PEMFC is in the range of 60 - 80 °C [63]. From literature, heat generation within the fuel cell can be traced back to four main sources [64, 65, 66] <sup>1</sup>:

1. Entropic heat of reactions, accounting for approximately 30 % of the total heat [63].
2. The irreversible heat of electrochemical reactions, accounting for approximately 60 % of the total heat [63].
3. Ohmic resistance losses accounting for approximately 10 % [63].

<sup>1</sup>It should be noted that the following contribution percentage are not exact and they can slightly differ in literature, but they can be taken as a rough estimation.

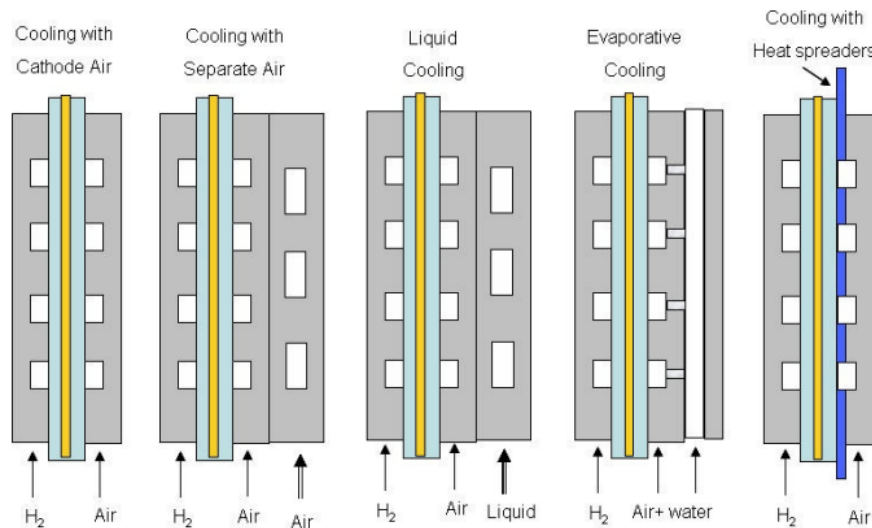


Figure 4.13: Different cooling strategies for the PEMFC [67]

Cooling the fuel cell can be categorized into three techniques depending on the medium used to remove the heat. These techniques are: air cooling, liquid cooling and two-phase cooling [4, 68]. Air cooling techniques, use air to remove the heat from the stack. Air flow at the cathode can be used directly to cool down the stack, or a separate air flow channel is used instead, so reactions at the cathode are not altered by the excess air flow [64]. When compared with other cooling techniques, it is the least complicated method of cooling and it is sometimes preferred in portable applications [68]. However, they are only applicable for PEMFC with low power outputs within the range of 100 and 2000  $kW$  [68]. This is due to the lower heat transfer capacity than liquid and two-phase flow cooling solutions.

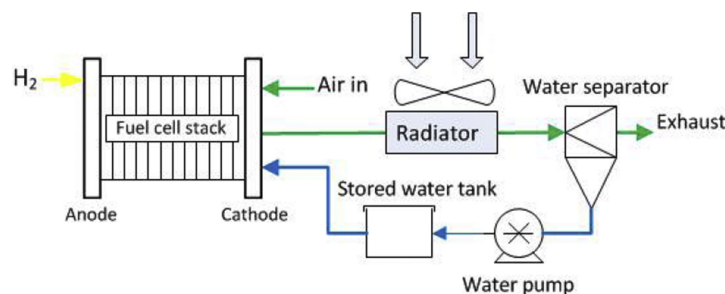


Figure 4.14: Evaporatively cooled PEMFC [69]

Phase change cooling is another active cooling technique, which is used by heat pipes in the edge cooling method and it uses the coolant latent heat to absorb and transfer the heat generated by the fuel cell stack [70]. This is one of the main advantages of this technique since the heat absorbed during the coolant phase change is much greater than if there is no phase change as in the sensible heat mode [70]. Edge cooling is categorized under as the phase change cooling method. It is a passive cooling technique since it does not require any sort of power input for heat transfer [68]. The two main mechanisms for this technique is the use of heat spreaders and the use of heat pipes [63, 68]. The former method uses highly thermally conductive materials to transfer the heat to the PEMFC stack edges as shown on the left of Figure 4.13 [68, 67]. Heat pipes on the other hand use two phase cooling to transfer the heat to the ambient air directly [63]. In its closed tubes, it has a fluid that evaporates at the pipe hot side (the stack side), cooled back again by convection at the cold side (The ambient side), and passively transported back again to the hot side by an action known as wicking through a special coating around the pipe [71]. It is worth mentioning that these techniques are only best for PEMFC of low output power which has a maximum of 1  $kW$  [63, 68].

Two additional approaches are used for two phase cooling: cooling through water and coolants with lower boiling point [64, 68]. Cooling through water, introduces the advantage of keeping the membrane humidified while removing the excess heat out of the stack [70]. The evaporated water is then condensed, using a heat ex-

changer, and re used again to cool the stack as shown in Figure 4.14 [69]. This method uses three approaches [64, 70]:

1. Direct liquid water introduction into the reactant gas channels.
2. bipolar plates with in-plane wicking material.
3. porous water transport bipolar plates.

Other coolants which have lower boiling points than water can be used. The boiling point of the coolant should be below the fuel cell operating temperature which ranges between 60 to 80 degrees [70]<sup>2</sup>. Two phase cooling can be used in high power output PEMFC of greater than 10 kW [70].

The last thermal management technique and the most common one in PEMFC applications is liquid cooling. It is so since heat capacity and heat transfer coefficients with liquid flow are much higher than that with air flow [64, 63, 70]. In the same manner as in evaporative cooling, the liquid coolant in this system passes through the fuel cell stack, removing its waste heat, and then get cooled back again by passing through a radiator as shown in Figure 4.15. Liquid cooling, similar to two phase cooling, is capable of functioning in high power output fuel cell stacks where powers can reach 80 kW or higher [64]. Two-phase cooling and liquid cooling are best suited to cool down the PEMFC of the HPS. Liquid cooling *now* has an edge over the two-phase cooling because of its higher technology readiness level [72]. However, two phase cooling, especially evaporative cooling has a higher heat transfer capability that allows reduced frontal area and thus more compact systems [69].

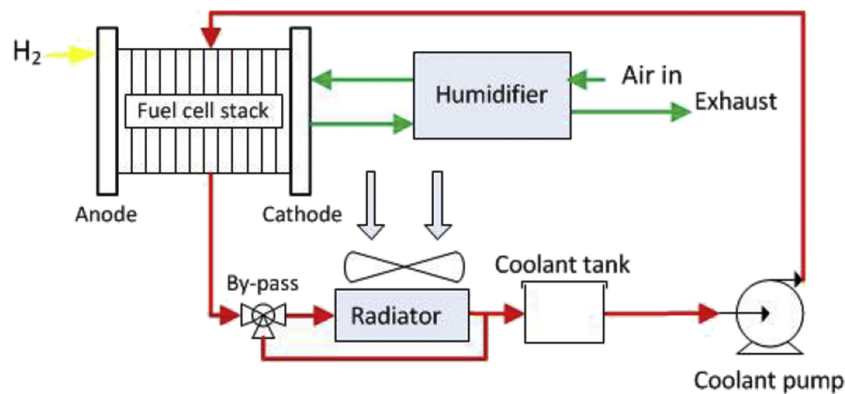


Figure 4.15: Liquid cooled PEMFC [69]

#### 4.4.3. HUMIDIFIER AND WATER MANAGEMENT

Keeping the PEM fuel cell membrane sufficiently hydrated is critical to maintain an optimum performance. The goal of humidification is to maintain a sufficient level of hydration to the fuel cell membrane, such that it is able to conduct protons efficiently and obtain the highest durability [4, 73]. Inadequate humidification may result in membrane dehydration and radical formation accelerating the membrane degradation process in consequence [73, 74, 75]. On the other hand, excess water supply may cause water flooding within the fuel cell porous electrode structures, impeding the reactants transport in result [4, 73].

Yafei et. al [73] in their review have categorised hydration techniques into two methods: internal and external humidification. Mengbo Ji et al. and M Fowler et al. [75, 76] concur with Yafei et. al [73] on these methods. Internal humidification method or the self-humidifying method, depends on hydrating the membrane, by changing the PEMFC internal structure or composition without the need of external devices [73], saving weight and volume, but on the expense of complicating the FC design. Moreover, this method is only applicable for low power applications, and its water management techniques have a limited operating condition especially if the hydration method is physical and not chemical [73].

External humidification, on the other hand humidifies the reactants in a separate humidifier before entering the fuel cell. Typical external humidification methods includes [73]:

<sup>2</sup>It can actually use water if the pressure within the fuel cell system is decreased [70]

1. Gas bubbling humidifier method
2. Direct water injection method
3. Enthalpy wheel humidifier method
4. Membrane humidifier method
5. Exhaust re-circulation method

Gas bubbling humidifier as shown in Figure 4.16, humidify the dry air by passing it through a tube to the bottom of a container of heated water. It then flows out after getting spread as bubbles in the water. Direct water injection mechanism shown in Figure 4.17, simply depends on directly supplying the dry air with water in either water or vapour forms. Both methods are better for stationary applications because of their large size [73]. Moreover they usually are subjected to significant parasitic losses, especially the gas bubbling humidification technique [76].

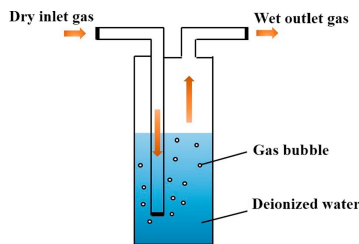


Figure 4.16: Gas bubbling humidifier schematic diagram [73]

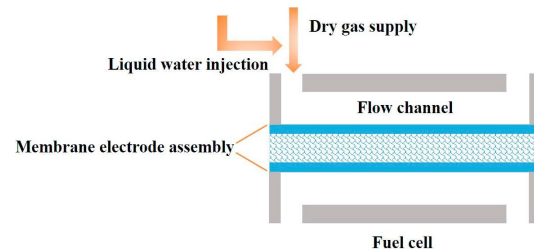


Figure 4.17: Liquid water injection humidification method [73]

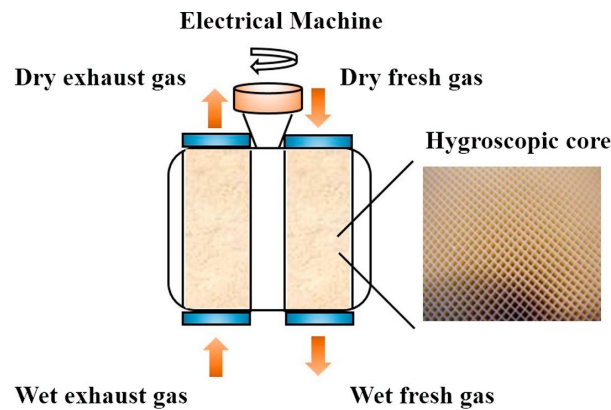


Figure 4.18: Enthalpy wheel humidifier Schematic diagram [73]

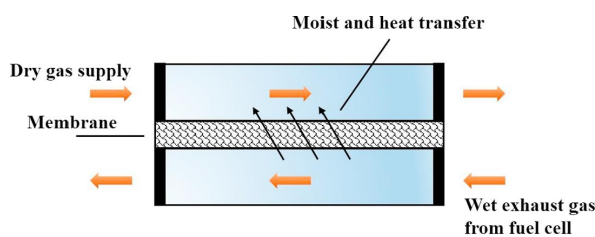


Figure 4.19: Gas to gas membrane humidifier schematic diagram [73]

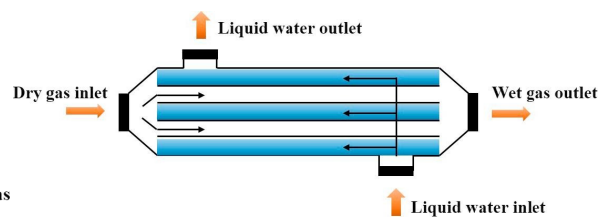


Figure 4.20: Liquid to gas membrane humidifier schematic diagram [73]

Enthalpy wheel humidifier as shown in Figure 4.18, uses a hygroscopic core with porous honeycomb shaped columns in which by slow rotation it transports heat and moisture from the hot moisture fuel cell exhaust gas to the dry inlet fresh gas [73, 76]. This method improves the overall efficiency, but gas leakage challenge and parasitic loss induced by the rotation of the enthalpy wheel are a draw back [73]. Membrane humidifiers can be classified as a gas to gas humidifier or a liquid to gas humidifier as shown in Figure 4.19 and Figure 4.20 respectively. Gas to gas membrane humidifier has two flow fields, one coming from the exhaust and the other going to the fuel cell inlet. In this case, the dry air get humidified by the wet exhaust gas. Liquid to gas membrane generally uses a tubular configuration, where it separates the liquid water and the dry reactant gas [73]. The liquid water is heated to vapour, transported across the membrane and get ab-



sorbed by the dry reactants. It is worth noting that the membrane based humidifiers are an attractive choice for mobile applications, due to their simplicity, light weight and good performance [73]. The final method, exhaust re-circulation uses the exhaust gas to humidify the inlet gases. However, the exhaust gases are mixed directly with the inlet gas reducing the complexity of the system. This method is simple, but usually it suffers from larger pumping losses [73].



## PROPULSION SYSTEM MODELLING

### 5.1. PURPOSE OF THE MODEL

The primary goal of the model is to predict the performance of a hybrid propulsion system that integrates a fuel cell system and a turbo generator. Therefore, performance parameters such as the fuel mass flow rate and efficiency are evaluated with respect to the different operating variables of the propulsion system such as the fuel cell operating temperature and reactant partial pressures.

The second goal of the model is to investigate the propulsion system performance beyond its design operation point. Furthermore, based on the analysed performance the weight is preliminary estimated based using power densities of the propulsion components, aiding in the design and optimization process. In this work, the developed model will be used to simulate the operation of the propulsion system throughout an eVTOL flight mission as shown in Figure 5.1. The mission in Figure 5.1, is an example mission for an eVTOL where it includes the major flight operations such as take-off, cruise, and landing.

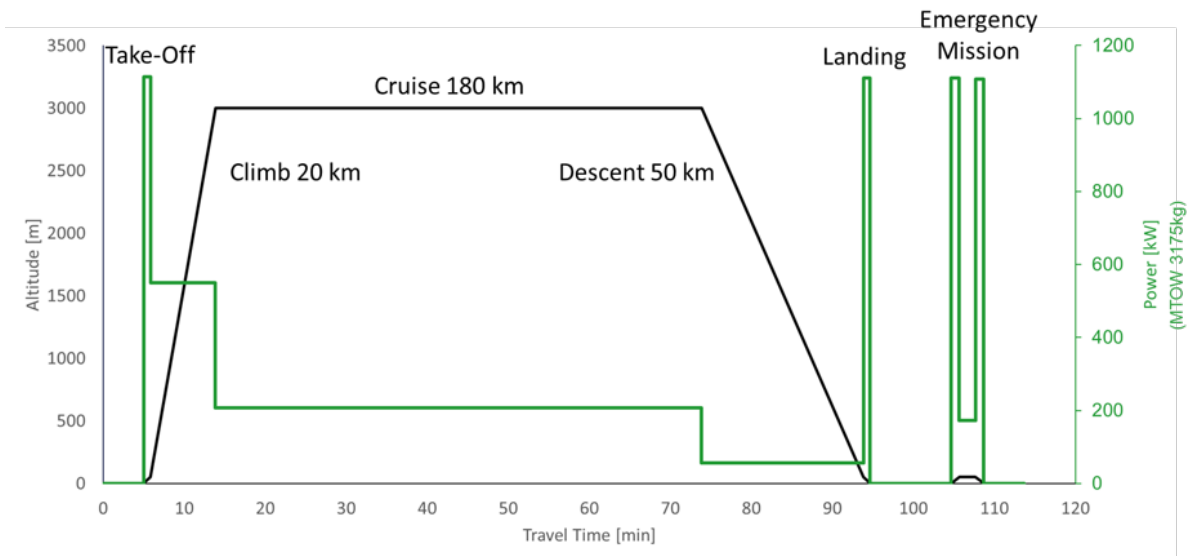


Figure 5.1: eVTOL flight mission and power requirement versus time.

For the scope of this work, 0D steady state models will be used. The model can be extended to 1D models if specific phenomena such as gas diffusion in the fuel cell diffusion layers require extra attention.

### 5.2. SYSTEM BOUNDARIES & VARIABLES

The hybrid propulsion system *HPS* include the following components within its boundaries:

- The fuel cell
- electric motor
- Intercooler
- Gearbox
- Balance of plant components:
- Heat exchanger / Cooling system
- AC/DC Converter
- Compressor
- Humidifier

- The turbo-generator:
- Gas turbine
- Generator

These components are integrated together as depicted in Figure 5.2.

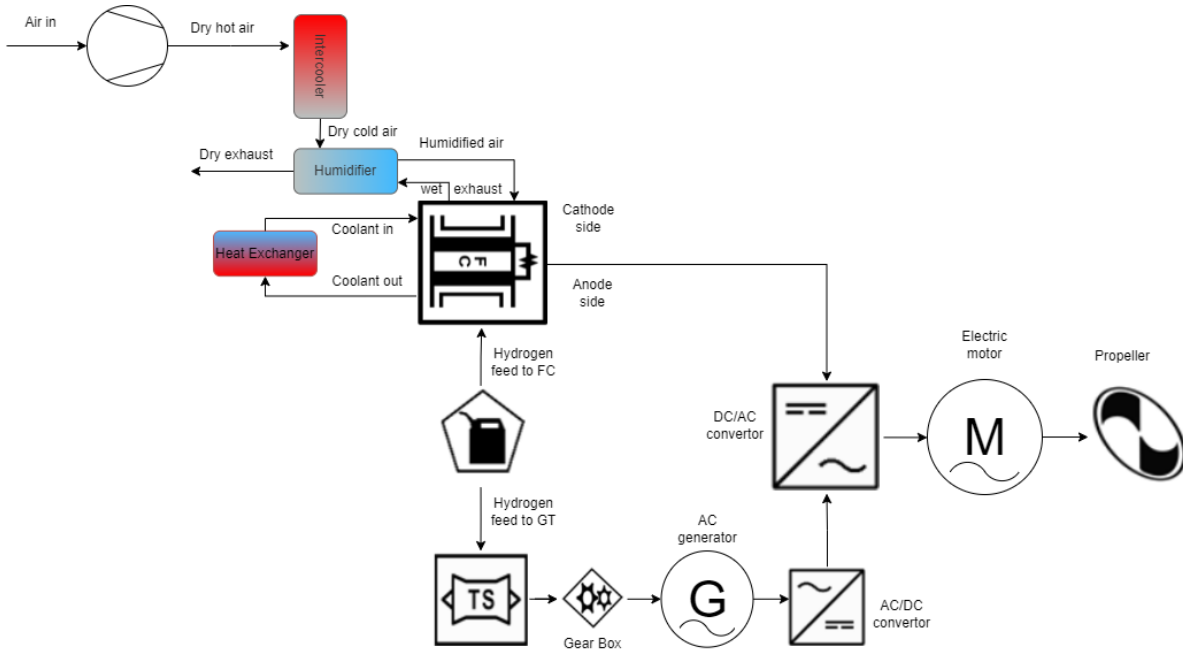


Figure 5.2: Simplified flow diagram example of the hybrid propulsion system.

It is crucial to define the system variables and any external influences. The key variables are summarized in Table 5.1 and the interdependencies of the fuel cell system key variables are sketched in detail in Figure 5.3. It should be noted that the external key variables in Table 5.1 represents also the model border variables together with the components power output.

Component	Variables
Atmosphere	Aircraft altitude $h$ , power and voltage requirement $P_{req}, V_{req}$ , Mach number $M$ , ambient temperature and pressure $p_{amb}, T_{amb}$ .
Fuel cell	Hydrogen mass flow rate $\dot{m}_{H_2}$ , Oxygen mass flow rate $\dot{m}_{O_2}$ , water production mass flow rate $\dot{m}_{H_2O}$ , fuel cell operating temperature $T_{op}$ , hydrogen and oxygen partial pressures $p_{H_2}, p_{O_2}$ , electric output voltage $V_{FC}$ , membrane water content ratio $\lambda_{water}$ , relative humidity $\phi$
Humidifier	inlet and outlet temperature and pressure $T_{humid}, p_{humid}$ , inlet and outlet relative humidity $\phi$
Compressor	Air mass flow rate $\dot{m}_{air}$ , inlet and outlet pressure $p_{t1}, p_{t2}$ , temperature $T_{t1}, T_{t2}$ and power consumption $P_{comp}$
Heat exchanger	Cooling system power consumption $P_{CS}$ and Fuel cell wasted heat $\dot{Q}_{wasted}$
Gas turbine	inlet and outlet temperatures and pressures and power output $P_{GT}$
Generator	Efficiency $\eta_{Gen}$ and electrical output
Gear box	Efficiency $\eta_{GB}$

Table 5.1: Hybrid propulsion system key variables

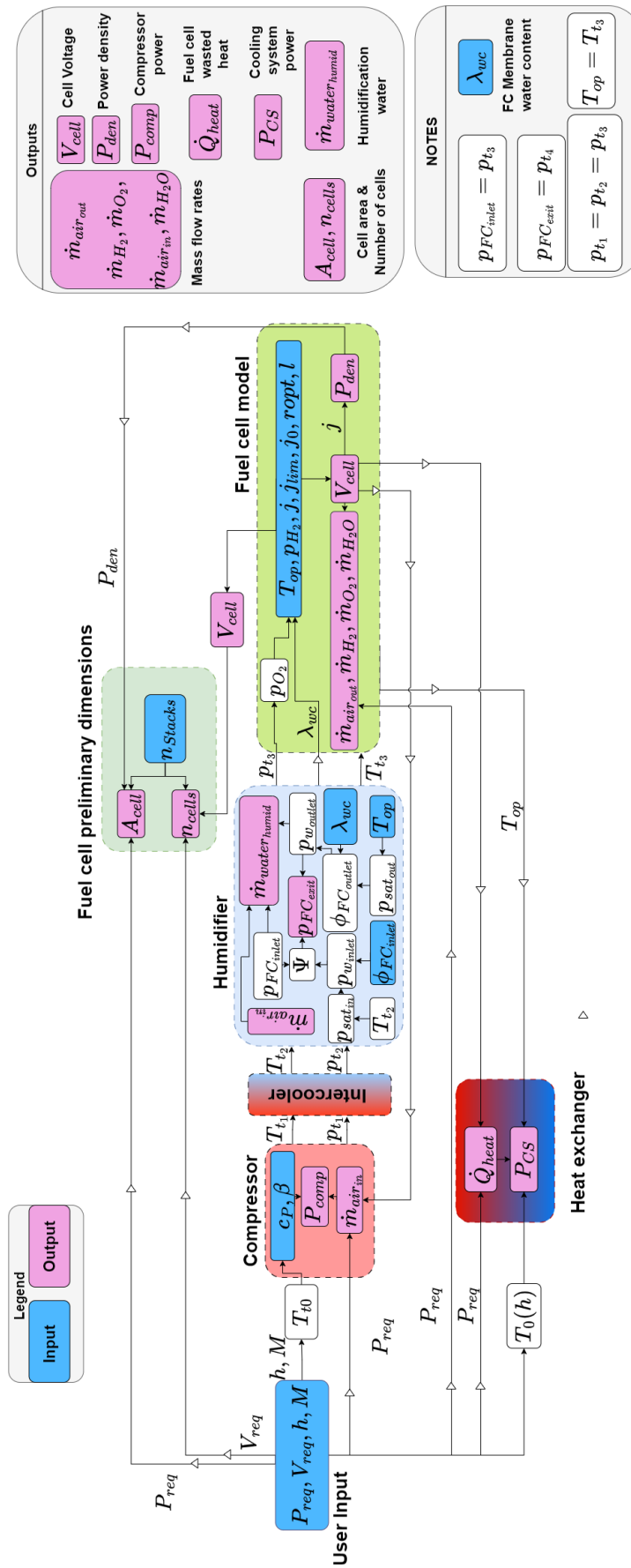


Figure 5.3: System components and associated variables

### 5.3. RELEVANT PROCESSES

In this section, the relevant processes associated with the HPS are outlined:

- The fuel cell is the main power source of the eVTOL unless the power requirement exceeds the maximum possible fuel cell power output.
- In case the eVTOL requested power exceeds the fuel cell provided output, the turbogenerator assists by delivering the power deficiency.
- The eVTOL altitude will vary the ambient air temperature and pressure which in result will affect the fuel cell cathode air partial pressure, compressor and cooling system power.
- The compressor of the fuel cell system run on an electric motor powered by the fuel cell and its main role is to pressurizes the ambient air to increase the performance of the fuel cell.
- The humidifier acts as a moisture exchanger where it recycles the water produced by the fuel cell to maintain the fuel cell membrane relative humidity between 80 % and 100 % [4].
- In the fuel cell, electrochemical reaction takes place between hydrogen and oxygen producing electricity, heat and water.
- The fuel cell main losses are activation, Ohmic and concentration losses.
- The fuel cell anode side is fed with pure gaseous hydrogen.
- The heat exchanger maintains the fuel cell operating temperature at 80 ° [4].
- Air is supplied with a stoichiometric ratio of 2.
- The compressor and cooling system are powered by the fuel cell.
- Thermodynamic processes like compression, expansion and heat transfer takes place within the compressor, gas turbine and heat exchanger.
- The architecture of the hybrid propulsion system is in series as shown in [Figure 4.5](#), and it can operate in three modes:
  - The fuel cell is the main power source and it derives the eVTOL propulsion device.
  - The turbo-generator solely powers the eVTOL propulsion device, in case of the fuel cell system malfunction.
  - A combined mode where both the fuel cell and the turbo-generator provides power to the eVTOL electric motors.

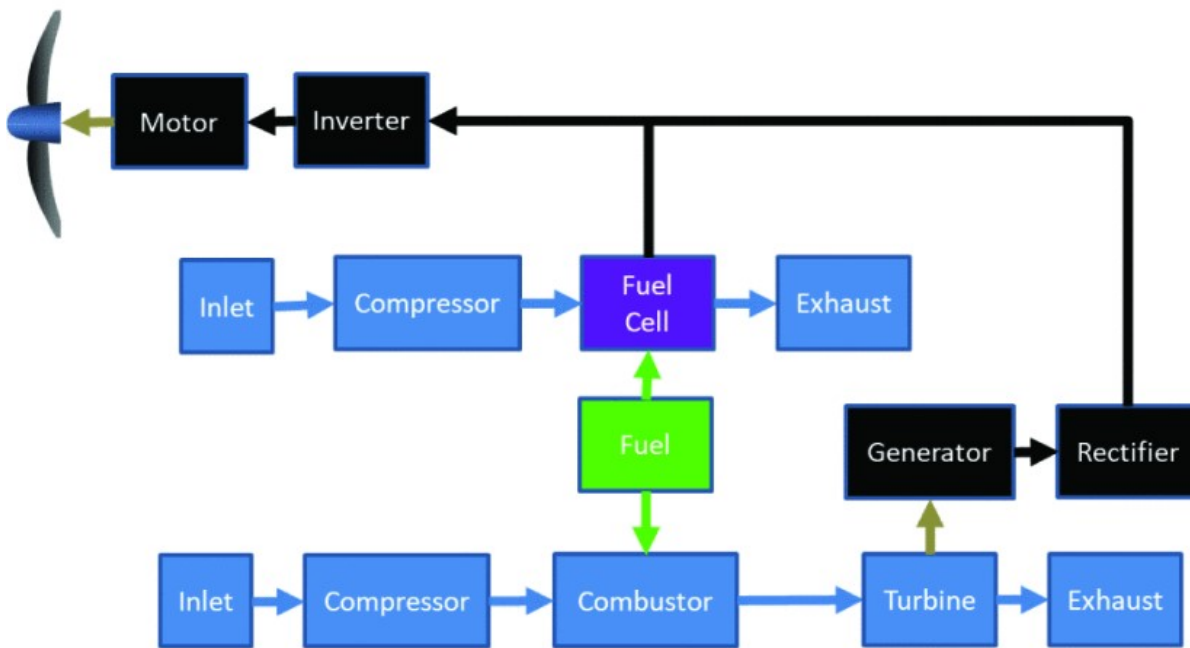


Figure 4.5 revisited: Fuel cell Turbogenerator hybrid architecture example [37].

## 5.4. ASSUMPTIONS

A model is only as accurate as its assumptions allow it to be [77]. Hence, the adopted assumptions to streamline the modelling process are addressed here:

- Overall system:
  1. The gas mixtures are ideal.
  2. The flow is laminar and adiabatic.
  3. Gravitational effects are neglected.
  4. The system adiabatic which means no heat is transferred to the system.
  5. The model assumes a steady state operation of the propulsion system components which means that any transient operation will be ignored.
  6. 0D models are used to have a simpler system of equations to be solved.
  7. There is no accumulation of mass or momentum in the HPS components.
- Fuel cell:
  1. Ohmic potential drop in solid components is negligible [49].
  2. The membrane and electrode structures are isotropic and homogeneous [49].
  3. Due to the constant water production at the fuel cell cathode side, the water content is assumed constant at the fuel cell electrolyte membrane and the cathode interfaces.
  4. Due to the constant water concentration assumption, the proton concentration will be constant. This assumption will help in deriving the semi-empirical activation losses later in [subsection 6.1.2](#).
  5. Electron resistivity  $R_{electron}$  is relatively small with comparison to the proton resistivity  $R_{proton}$ , because it is easier to transport [3]. Therefore,  $R_{electron}$  is neglected.
  6. Inlet and outlet air pressure of the humidifier stays constant. In other words, pressure drops across the humidifier are considered negligible.
  7. Hydrogen utilization efficiency is 100% .
  8. The produced water is gaseous and thus the low heating value is used.

9. It is expected that the humidification process will cool down the flow coming from the compressor because it involves water evaporation in which the energy required for evaporation comes from the inlet air. However, because of the increased complexity of such phenomenon, it will be assumed that the flow temperature stays the same after the compressor.
  10. Nafion is assumed to be the material used for the fuel cell membrane.
- Gas turbine:
    1. Two specific heat constants ( $c_{p_{air}}$ ,  $c_{p_{gas}}$ ) and two specific heat ratios ( $k_{air}$ ,  $k_{gas}$ ) are used according to the engine station number. It is assumed that the temperature effect on the fluid properties is zero.
    2. Total temperatures and pressures have been used to simplify the calculations. This will allow the change in kinetic energy between inlets and outlets of the engine components to be ignored, as they are taken care of by the total properties e.g. total fuel cell compressor inlet temperature  $Tt_1$ .
    3. Mechanical shaft losses are accounted for in the mechanical efficiency.
    4. The compression and expansion processes are considered adiabatic but not reversible, so isentropic efficiencies of the engine components are included in the simulation.
    5. Constant mass flow rate is taken throughout the core and the bypass sections of the engine. Fuel mass flow will be added in the core but will remain constant after that.
    6. No turbine cooling or bleed air is considered.
  - Balance of plant components:
    1. The compressor isentropic efficiency, electric motor efficiency and generator efficiencies are assumed constant and equal to 80 % , 90 % [4] and 90%, respectively.
    2. Compression ratio  $\beta$  is constant throughout the flight envelope and it does not change with the air mass flow rate.
    3. Frictional effects at the compressor inlet are ignored and it is assumed that the ambient total pressure is equal to the compressor inlet pressure.
    4. The cooling fan in the HEX system pull the air through the heat exchanger resulting in a net zero thrust/drag and therefore the parasitic drag is neglected [78].
    5. The hybrid propulsion system only powers the eVTOL propulsion device and the corresponding balance of plant components.

## 5.5. CONSERVATION LAWS

To ensure the accuracy and reliability of the model, fundamental conservation laws are used. This includes the mass, energy and momentum laws of conservation laws.

### 5.5.1. CONSERVATION OF MASS

The general conservation of mass equation implies that the total mass entering the system must equal the total mass leaving the system making sure no mass is lost or created. This is shown in Equation 5.1, where  $\dot{m}_V$  is the mass flow rate going through a control volume  $V$ . The model is in steady state and the equation is re-written to Equation 5.2.

$$\frac{dm_v}{dt} = \dot{m}_{in} - \dot{m}_{out} \quad (5.1)$$

$$\dot{m}_{in} = \dot{m}_{out} \quad (5.2)$$

For the fuel cell system, Equation 5.3 is formulated.

$$\dot{m}_{H_2} + \dot{m}_{air_{in}} = \dot{m}_{H_2O} + \dot{m}_{air_{out}} \quad (5.3)$$

### 5.5.2. CONSERVATION OF ENERGY

The conservation of energy ensures that the sum of the energy entering and leaving components are equal with taking into consideration the generated electrical energy. The conservation of energy equation for a gas turbine component is shown in Equation 5.4 [79], where  $Q$  and  $P_{\text{power}_{\text{abs}}}$  are the heat and power absorbed by the components.  $u$  and  $h$  are the internal energy and the enthalpy, respectively. The model is steady state and thus the time differentials are taken as zero. Furthermore, due to the assumption that no heat is added to the system from outside  $Q$  is taken as zero. For the drive shaft, Equation 5.5 is formulated [79].

$$\frac{dM_v}{dt} \cdot u + M_v \cdot \frac{dU}{dt} - \overset{0}{Q} = \dot{m}_{\text{in}} \cdot h_{\text{in}} - \dot{m}_{\text{out}} \cdot h_{\text{out}} + P_{\text{power}_{\text{abs}}} \quad (5.4)$$

$$I \cdot \frac{d\omega}{dt} = P_{\text{power}_{\text{abs}}} + P_{\text{power}_{\text{del}}} \quad (5.5)$$

For the fuel cell, the energy conservation equation can be summarized in Equation 5.6 where  $H_i$  is the species enthalpy in  $\text{kJ/mol}$ .  $-241.83 \text{ kJ/mol}$  is the enthalpy of formation of water in the gaseous form.  $\dot{W}_{\text{elec}}$  and  $\dot{Q}_{\text{wastedheat}}$  are the electric energy output and the wasted heat generated by the fuel cell:

$$\overbrace{\Sigma H_{i_{\text{in}}} - \Sigma H_{i_{\text{out}}}}^{\Delta H_{rxn}} = \dot{W}_{\text{elec}} + \dot{Q}_{\text{wastedheat}} \quad (5.6)$$

$$0 - (-241.83) \times \frac{\dot{m}_{\text{H}_2}}{M_{\text{H}_2}} = V_{\text{cell}} \times I + (E_0 - V_{\text{cell}}) I$$

# 6

## COMPONENT MODELLING

This chapter will focus on the methodology used to model the fuel cell and the balance of plant components. It will also include the approach used to model the turbo-generator using GSP.

### 6.1. THE FUEL CELL

#### 6.1.1. FUEL CELL ANALYTICAL MODEL

It is important to remember that the goal of the fuel cell is to extract the internal energy from the fuel and convert it into useful forms of energy such as heat or work [3]. The fuel cell voltage  $E_{FC}$  is subjected to multiple of losses as mentioned earlier in [subsection 4.2.1](#) and this can be formulated by [Equation 6.1](#) [3, 56].

$$E_{FC} = E_{thermo} - E_{act} - E_{ohmic} - E_{conc} \quad (6.1)$$

Each term in [Equation 6.1](#) will be further elaborated in mathematical terms. [Equation 6.2](#) calculates the thermodynamic voltage  $E_{thermo}$  which is the PEM reversible cell voltage at a given temperature and pressure. [Equation 6.2](#) is the Nernst equation but modified to account for temperature variations. The Nernst equation only accounts for pressure and concentration effects, and hence an additional term was added to account for temperature [3]:

$$E_{thermo} = E^0 + \underbrace{\frac{RT}{nF} \ln(p_{H_2} \sqrt{p_{O_2}})}_{\text{Nernst Equation}} + \frac{\Delta s}{nF} (T - T_0) \quad (6.2)$$

$E^0$  is the standard state reversible voltage and it can be computed by simply dividing the Gibbs free energy<sup>1</sup> difference for the chemical process  $\Delta g_{reaction}$  and the electrons charge  $Q$  as shown in [Equation 6.3](#) [54, 3].

$$E_0 = -\frac{\Delta g_{reaction}}{Q} = -\frac{\Delta g_{reaction}}{nF} \quad (6.3)$$

It is important to note that  $E^0$  is the maximum available potential the fuel cell chemical reaction can provide under a constant pressure and temperature. A reversible voltage of 1.229 V is calculated for a hydrogen oxygen reversible reaction under standard conditions of T equal to 298.15 K and P equal to 1 atm [3].

The activation loss  $E_{act}$  can be derived from the Butler-Volmer equation which relates the fuel cell activation over-potential and the electrochemical reaction rate  $j_0$  [3, 4]. It should be noted the usually the fuel cell operates at high current densities, so the Butler-Volmer equation can be simplified to the Tafel equation as shown in [Equation 6.4](#).  $j_{leak}$  is added to the current density term  $j$  to account for the parasitic current losses that occur due to current leakage, gas crossover and unwanted side reactions [3].

$$E_{act} = \frac{RT}{n\alpha F} \ln\left(\frac{j + j_{leak}}{j_0}\right) \quad (6.4)$$

Ohmic losses arise because of the resistance of the fuel cell against charge transport. Using Ohm's law, one can derive the amount of potential required to transport the charge. This potential is the ohmic loss and it is derived in [Equation 6.5](#). The current density  $j$  is used to normalise the current, and for that the area specific resistance (ASR) is used instead of  $R$ .

<sup>1</sup>Gibbs free energy in simple terms represents the exploitable energy potential of a system [3]. It can be expressed as  $\Delta G = H - TS = U + PV - TS$  where U is the internal energy, P is the pressure, V is the volume, T is the temperature and S is the entropy



$$E_{ohmic} = iR \quad (6.5)$$

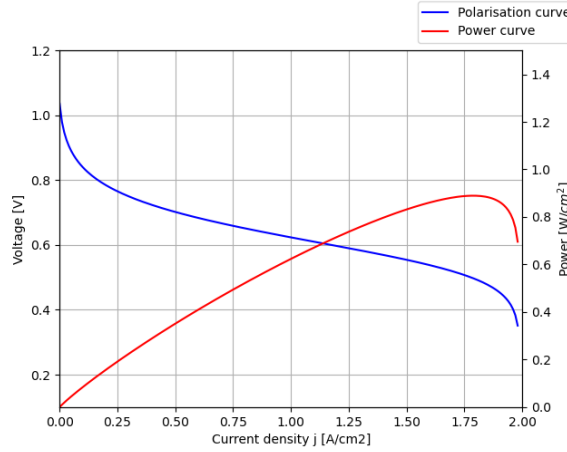
$$E_{ohmic} = ASR_{ohmic} \cdot j \quad (6.6)$$

Finally, Equation 6.7 calculates concentration losses which are derived from the Nernst reversible voltage and Butler-Volmer equations, since both are affected when reactant and product concentrations are affected. A new term had to be defined while derivation, the limiting current density  $j_L$ , which describes the maximum current density possible at the highest depletion (or production) rate possible<sup>2</sup>. Same as before  $j_{leak}$  is added to the current density term to account for parasitic current losses.

$$E_{conc} = E_{Nernst} + E_{BV} \quad (6.7)$$

$$E_{conc} = \left(\frac{RT}{nF}\right) \left(1 + \frac{1}{\alpha}\right) \ln\left(\frac{j_L}{j_L - (j + j_{leak})}\right) = c \ln\left(\frac{j_L}{j_L - (j + j_{leak})}\right) \quad (6.8)$$

Using the aforementioned method, the polarization curve can be plotted as shown in Figure 6.1. The figure also includes the power curve which is the product of the current density and the voltage. It is worth noting that it is better not to aim for the maximum power, since it occurs at higher current densities which results in lower efficiencies.



**Figure 6.1:** An example of the PEM fuel cell polarization and power density curves. The initial part of the j-V curve is dominated by the activation losses. The middle linear like shape is dominated by the Ohmic losses. Concentration losses shapes the end of the curve since it dominates at high currents.

The production and consumption rate of the reactant gases can be calculated using Equation 6.9, with  $M$ ,  $P$  and  $V$  are the reactants molar mass, power and voltage respectively. This equation can be further tailored for hydrogen, air and water as respectively shown in Equation 6.10, Equation 6.11, Equation 6.12 and Equation 6.13.

$$\dot{m} = \frac{M \cdot i}{2F} \quad [\text{kg/s}] \quad (6.9)$$

$$\dot{m} = \frac{M \cdot j}{2F} \quad [\text{kg}/(\text{s} \cdot \text{cm}^2)] \quad (6.9a)$$

$$\dot{m} = \frac{M}{2F} \cdot \frac{P_{total}}{V_{cell}} \quad [\text{kg/s}] \quad (6.9b)$$

$$\dot{m}_{H_2} = \frac{\lambda_{H_2} I M_{H_2}}{2F} = 1.05E-8 \frac{P_{total}}{V_{cell}} \quad [\text{kg/s}] \quad (6.10)$$

<sup>2</sup>The rate that can bring the concentration to zero!

$$\dot{m}_{air_{in}} = \frac{\lambda_{air} I M_{O_2}}{4F} = 3.57E-7 \cdot \lambda_{O_2} \frac{P_{total}}{V_{cell}} \quad [kg/s] \quad (6.11)$$

$$\dot{m}_{air_{out}} = (3.57E-7 \lambda_{O_2} - 8.29E-8) \frac{P_{total}}{V_{cell}} \quad [kg/s] \quad (6.12)$$

$$\dot{m}_{water} = (9.34E-8) \frac{P_{total}}{V_{cell}} \quad [kg/s] \quad (6.13)$$

$$\eta_{real} = \eta_{thermo} \cdot \eta_{voltage} \cdot \eta_{fuel} \quad (6.14)$$

$$\eta_{thermo} = \frac{\Delta g_{reac}}{\Delta h_{HHV}} \quad (6.15)$$

$$\eta_{voltage} = \frac{V}{E} \quad (6.16)$$

$$\eta_{fuel} = \frac{i}{v_{fuel} nF} = \frac{1}{\lambda} \quad (6.17)$$

It is important to mention the fuel cell efficiency to have an understanding of its performance. Ideally, the efficiency of the fuel cell is represented by the thermodynamic efficiency which is 83% [3]. Most reversible (ideal) fuel cell voltages decrease with increasing temperature, however that does not mean that the lower the temperature the better the performance. This is due to the fact that kinetic losses tend to decrease with increasing the temperature [3]. The real efficiency will be lower than the thermodynamic efficiency and it can be calculated using Equation 6.14 [3] where  $\eta_{thermo}$  is the thermodynamic efficiency and is calculated using Equation 6.15. The two added efficiencies are the voltage  $\eta_{voltage}$  and fuel utilisation  $\eta_{fuel}$  efficiencies and they can be calculated using Equation 6.16 and Equation 6.17 respectively. The voltage efficiency  $\eta_{voltage}$  is the ratio between the real output of the fuel cell  $V$  and the reversible voltage  $E$ . The higher the load the lower the output because of the increased losses and thus the lower the voltage efficiency. The fuel utilisation efficiency  $\eta_{fuel}$  accounts for how much fuel does the fuel cell actually uses. It is preferred that fuel cells operate at a constant stoichiometric condition (constant  $\lambda$ ), where the fuel flow rate is adjusted according to the fuel cell load requirement.

### 6.1.2. FUEL CELL SEMI-EMPIRICAL MODEL

#### SEMI-EMPIRICAL MODELS IN LITERATURE

The study from A.Saadi et al. paper [80] made a comparison between three PEMFC static models: Amphlett, Larminie-Dicks, and Chamberline-Kim models. The Amphlett model with comparison to the two other models is precise and detailed in handling the fuel cell physical parameters such as the fuel cell membrane thickness and ionic conductivity [54]. It uses Nernst and Tafel equations to model activation and ohmic losses which will be further discussed later in this section. On the other hand, the Larminie-Dicks and Chamberline-Kim models are simpler and may not capture complex PEMFC behaviours as the Amphlett model does [80]. It is noted that the Larminie-Dicks model has a simpler modelling approach with good balance between simplicity and accuracy but may not capture all the nuances of the fuel cell behaviour as precisely as the Amphlett model. The Chamberline-Kim model, while the simplest, tends to have higher error margins, particularly in activation loss predictions [80]. The Open Source [PEM] Cell Simulation Tool (OPEM) [81] is used to carry out the fuel cell model simulations and in the following section, the steady state model used by the tool will be explained.

#### AMPHLETT SEMI-EMPIRICAL MODEL

The propose of this work is to predict the performance and have an accurate understanding of the exact impact of the varying each parameter on the fuel cell performance. Therefore, it is decided to use the Amphlett model as the base of the model. The Amphlett model is a semi-empirical model which uses both analytical and empirical components. It uses Equation 6.1 and Equation 6.18, to calculate the PEMFC open circuit voltage and its thermodynamic voltage.

$$E_{FC} = E_{thermo} - E_{act} - E_{ohmic} - E_{conc} \quad (6.1 \text{ revisited})$$

$$E_{thermo} = E^0 + \underbrace{\frac{RT}{nF} \ln(p_{H_2} \sqrt{p_{O_2}})}_{\text{Nernst Equation}} + \frac{\Delta s}{nF} (T - T_0) \quad (6.18 \text{ revised})$$

The model uses empirical relations based on data gathered from the single fuel cell of the MkIV produced by Ballard Power System Inc. [54]. They are used to calculate the activation losses, utilizing empirically determined Tafel equation coefficients  $\xi_1, \xi_2, \xi_3, \xi_4$  as shown in Equation 6.18. These coefficients are calculated using Equation 6.19 and they adjust the analytical calculations to reflect real-world performance, accounting for factors like electrode material properties and cell construction. It also includes empirical corrections that includes the real gas conditions, membrane humidity and gas pressure impacts, which explains the inclusion of species concentration terms  $C_{O_2}$  and  $C_{H_2}$ . These terms can be calculated using Equation 6.20 and Equation 6.21.

$$E_{act} = \xi_1 + \xi_2 T_{FC} \ln C_{O_2} + \xi_4 T_{FC} \ln i \quad (6.18)$$

$$\begin{aligned} \xi_1 &= -0.948 \\ \xi_2 &= 0.00286 + 0.197 \times 10^{-3} \ln A_{rea} + 4.3 \times 10^{-5} \ln C_{H_2} \\ \xi_3 &= 6.3 \times 10^{-5} \\ \xi_4 &= 0.72 \times 10^{-4} \end{aligned} \quad (6.19)$$

$$C_{O_2} = \frac{P_{O_2}}{5.08 \times 10^6 \exp\left(\frac{-498}{T_{FC}}\right)} \quad (6.20)$$

$$C_{H_2} = \frac{P_{H_2}}{1.09 \times 10^6 \exp\left(\frac{77}{T_{FC}}\right)} \quad (6.21)$$

Ohmic losses  $E_{Ohmic}$  can be calculated using Equation 6.22, where  $R_{proton}$  and  $R_{electron}$  are the resistances caused by the proton transfer at the solid FC membrane and electron transfer at the graphite electrodes respectively [54, 56].  $R_{electron}$  is assumed to be zero because of its relatively small value when compared to  $R_{proton}$ .  $R_{proton}$  is expressed by Equation 6.23 where  $l$  and  $A_{rea}$  are the fuel cell membrane thickness and area respectively.  $\rho_M$  is the membrane specific resistivity and it is a function of the membrane type, characteristics, Fuel cell temperature  $T_{FC}$ , water moles to each sulfonic group  $\lambda_{water}$  and current density as shown in Equation 6.24, which is an empirical expression for Nafion membrane resistivity [56].

$$E_{Ohmic} = i \times (R_{proton} + R_{electron}) \quad (6.22)$$

$$R_{proton} = \frac{\rho_M l}{A_{rea}} \quad (6.23)$$

$$\rho_M = \frac{181.6 \left[ 1 + 0.03 \left( \frac{i}{A_{rea}} \right) + 0.062 \left( \frac{T_{FC}}{303} \right)^2 \left( \frac{i}{A_{rea}} \right)^{2.5} \right]}{\left[ \lambda - 0.634 - 3 \cdot \frac{i}{A_{rea}} \right] \cdot \exp \left[ 4.18 \left( \frac{T_{FC} - 303}{T_{FC}} \right) \right]} \quad (6.24)$$

Finally, the concentration losses  $E_{con}$  are calculated using Equation 6.25 where  $B$  can be calculated using Equation 6.26 which is a function of the FC temperature [4].

$$E_{conc} = -B \times \ln \left( 1 - \frac{j}{j_{max}} \right) \quad (6.25)$$

$$B = \frac{RT_{FC}}{nF} \quad (6.26)$$

### 6.1.3. FUEL CELL MODEL VERIFICATION

The code used is verified and tested upon different variables: fuel cell operating temperature, reactants operational partial pressure and water content. The fuel cell voltage and power is expected to increase when increasing the aforementioned variables. Indeed the model behaved in an expected manner. Figure 6.2, Figure 6.3, Figure 6.4 and Figure 6.5 shows the effect of increasing the temperature, hydrogen, oxygen partial pressures and membrane water content respectively on the fuel cell voltage and power density output.

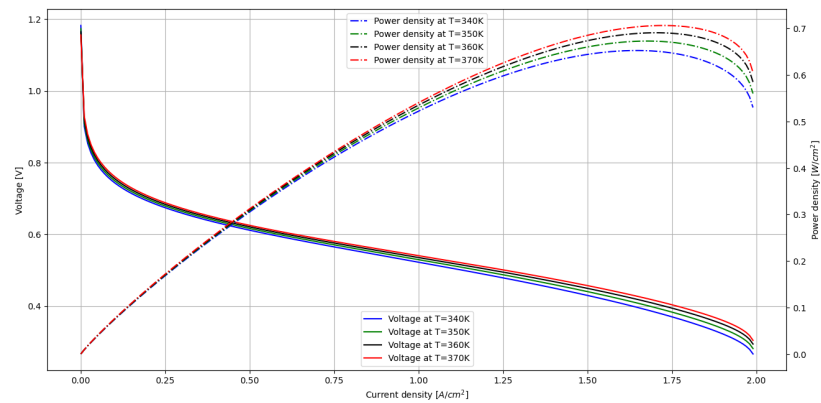


Figure 6.2: The fuel cell polarization curve with varying operating temperature.

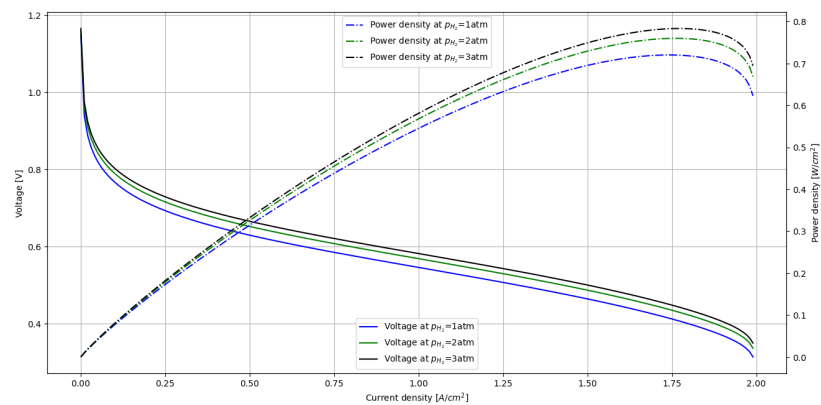
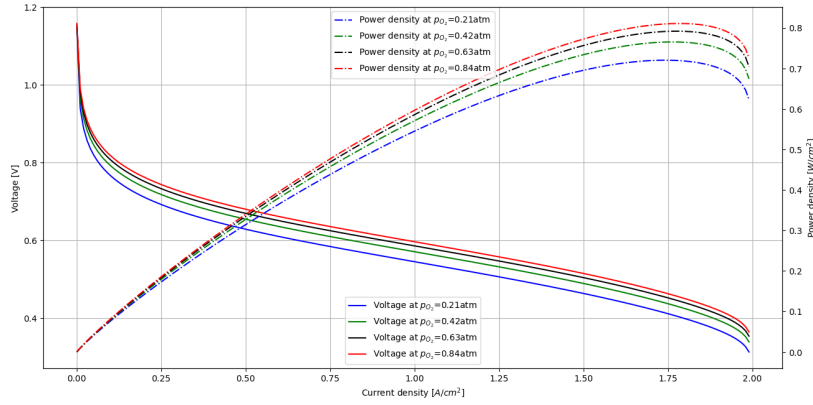
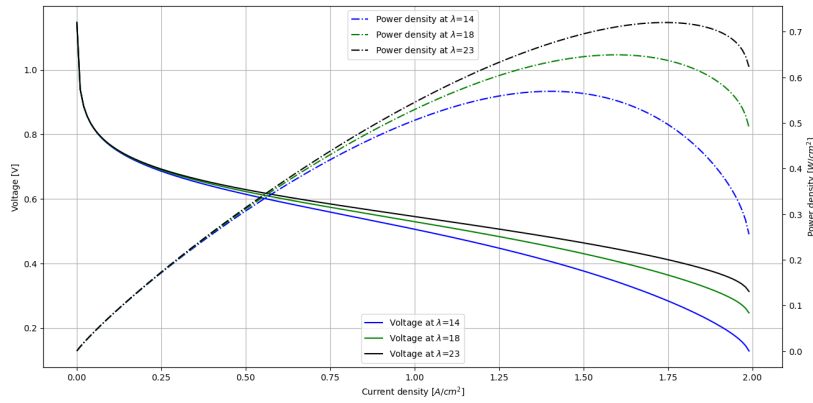


Figure 6.3: The fuel cell polarization curve with varying hydrogen partial operating pressure.



**Figure 6.4:** The fuel cell polarization curve with varying oxygen partial operating pressure.



**Figure 6.5:** The fuel cell polarization curve with varying water content parameter.

## 6.2. BALANCE OF PLANT COMPONENTS

Balance of plant components (BoP) calculations require air properties at the different altitudes of the eVTOL flight envelope. The Atmosphere python library is used to calculate the air temperature, pressure, speed of sound and density. The later two properties are used with the eVTOL flight speed to calculate the Mach number. The temperature and pressures obtained from the library are based on the altitude and they are static values. It is better to use total properties to simplify the calculations, since this will allow the change in kinetic energy between inlets and outlets of the compressor for instance to be ignored. This is due to the fact that the total properties takes care of both static and dynamic properties. Total ambient temperature and pressure are calculated using Equation 6.27 and Equation 6.28 respectively. The equations are written in terms of Mach number instead of the velocity.

$$T_{0,0} = T_0 + \frac{v_0^2}{2c_{p,a}} = T_0 \left( 1 + \frac{\kappa_a - 1}{2} M_0^2 \right) \quad (6.27)$$

$$p_{0,0} = p_0 \left[ 1 + \frac{v_0^2}{2c_{p,a} T_0} \right]^{\gamma_a / (\gamma_a - 1)} = p_0 \left[ 1 + \frac{\gamma_a - 1}{2} M_0^2 \right]^{\gamma_a / (\gamma_a - 1)} \quad (6.28)$$

### 6.2.1. COMPRESSOR

The compressor model is composed only with its power equation. At this stage of the project, it is only required to get an estimate of the power used by the compressor.

The compressor power is calculated using Equation 6.29, where  $\eta_m$  and  $\eta_{comp}$  are the motor and compressor efficiencies respectively [4]. The values taken for the motor and compressor efficiencies are 0.9 and 0.8 respectively.  $\beta$  is the pressure ratio between the compressor exit pressure and the ambient pressure. It is assumed to be a constant value of 3.5, derived from the fuel cell on design requirements. Furthermore, the air mass flow rate  $\dot{m}_{air_{in}}$  is calculated using Equation 6.11 which is a function of the fuel cell gross power and voltage. It should be noted that the gross power is the total power of all the cells in the fuel cell combined whereas the voltage used in the equation is the voltage of a single cell. Also, the outlet compressor temperature is calculated using Equation 6.31 which will be used later for humidity calculations.

$$\dot{m}_{air_{in}} = \frac{\lambda_{air} I M_{O_2}}{4F} = 3.57E-7 \cdot \lambda_{O_2} \frac{P_{total}}{V_{cell}} \quad (6.11 \text{ revisited})$$

$$P_{Comp} = \dot{m}_{air_{in}} C_P \frac{T_{t_0}}{\eta_m \eta_{comp}} \left( \beta^{\frac{\gamma-1}{\gamma}} - 1 \right) \quad (6.29)$$

$$T_{t_0} = T_0 \left( 1 + \frac{\gamma-1}{2} M_0^2 \right) \quad (6.30)$$

$$T_{t_1} = T_{t_0} \left( 1 + \frac{1}{\eta_{comp}} \left( \beta^{\frac{\gamma-1}{\gamma}} - 1 \right) \right) \quad (6.31)$$

### 6.2.2. HUMIDIFIER

The humidifier main goal is to add water to the fuel cell gaseous reactants: Air and hydrogen. It will be assumed in this work that the water vapour is added only through the cathode – the air side – for simplicity. In order to relate to the humidity level of the fuel cell membrane, ratio  $\lambda_{water}$  is used.  $\lambda_{water}$  is a ratio between the number of water moles and the sulfonic acid  $SO_3^-$  group in the nafion memberane which directly indicates the membrane's relative humidity [3, 82]. In Equation 6.24,  $\lambda_{water}$  is required to determine the membrane specific resistivity and thus the Ohmic losses  $E_{Ohmic}$ . As discussed earlier in subsection 4.2.1, fuel cell relative humidity shall be kept between 80% and 100% for a well hydrated electrolyte membrane [4] and therefore the outlet relative humidity  $\phi_{exit}$  of the fuel cell will be kept constant at 100 %. Using the correlation given by Equation 6.32, a 100 % outlet relative humidity corresponds to a ratio  $\lambda_{water}$  of approximately 14 [82].

$$\rho_M = \frac{181.6 \left[ 1 + 0.03 \left( \frac{i}{A_{rea}} \right) + 0.062 \left( \frac{T_{FC}}{303} \right)^2 \left( \frac{i}{A_{rea}} \right)^{2.5} \right]}{\left[ \lambda_{water} - 0.634 - 3 \cdot \frac{i}{A_{rea}} \right] \cdot \exp \left[ 4.18 \left( \frac{T_{FC} - 303}{T_{FC}} \right) \right]} \quad (6.24 \text{ revisited})$$

$$\lambda_{water} = 0.0043 + 17.81 \times \phi_{outlet} - 39.85 \times \phi_{outlet}^2 + 36 \times \phi_{outlet}^3 \text{ for } 0 < \phi_{outlet} \leq 1 \quad (6.32)$$

Equation 6.33 is used to calculate the water partial pressure  $p_{w_{exit}}$  at the fuel cell exit where  $p_{sat}$  is the saturated pressure. It is worth noting that the saturated vapour pressure  $p_{sat}$  is the partial water pressure when the air and water mixture are in equilibrium [4]<sup>3</sup>. The saturated vapour pressure varies with temperature non-linearly in which it increases rapidly with increasing temperature. Table A.1 contains the saturated vapour pressures for a range of temperatures, and linear regression is used to estimate the saturation vapour pressure at a given temperature. The saturation pressure  $p_{sat}$  at a temperature of 353 K is equal to 47.41 kPa.

$$\phi_{inlet/outlet} = \frac{p_{w_{inlet/outlet}}}{p_{sat}} \quad (6.33)$$

The model assumes that the inter-cooler present between the compressor and the humidifier will cool down the high temperature compressor exhaust  $T_{t_2}$  to the operating temperature of the fuel cell  $T_{FCOC}$ . This means that the inlet and the outlet temperatures of the flow at the fuel cell cathode side will be equal to the fuel cell operating temperature. Moreover, the inlet relative humidity is kept constant at 40 % and by using the saturation pressure at the fuel cell operating temperature, one can calculate the water partial pressure  $p_{w_{inlet}}$  at the cathode inlet using Equation 6.33. The water from the exit gas is reused and the fuel cell exit pressure is calculated using Equation 6.34 where  $\Psi$  is a coefficient that is determined using Equation 6.35. Furthermore,  $\lambda_{air}$  is the air stoichiometric ratio and it is kept constant at 2 [4].

<sup>3</sup>It is basically the pressure of the air when it can not hold any more water vapour.

$$p_{exit} = \frac{p_{W_{outlet}}((1 + \Psi)\lambda_{air} + 0.210)}{0.420 + \Psi\lambda_{air}} \quad (6.34)$$

$$\Psi = \frac{p_{W_{in}}}{p_{in} - p_{W_{in}}} \quad (6.35)$$

If the water is not recycled or the fuel cell is self humidifying,  $\Psi$  will be zero and Equation 6.34 can be rewritten into Equation 6.36.

$$p_{exit} = \frac{(\lambda + 0.210)p_{W_{outlet}}}{0.420} \quad (6.36)$$

The ratio between water and air mass flows is defined as the specific humidity  $\omega$  as shown in Equation 6.37, where  $P_W$  and  $P_{air}$  are the partial pressures of water and air. It is important to note that the mass of any species in a mixture is proportional to the product of the species partial pressure and its molecular mass. It is usually not easy to obtain the dry air partial pressure and the total pressure  $p$  is used instead. Therefore, Equation 6.37 is re-arranged to calculate the water's mass as shown in Equation 6.38. With the air mass flow rate, the total pressure and the desired inlet water pressure specified, the amount of water supply  $\dot{m}_w$  required can be calculated. Moreover, since the exhaust gases are reused, Equation 6.13 can be used to calculate how much water is produced and therefore how much water is leaving the fuel cell. The humidifier will thus have to extract the required supply of water from the leaving water vapour.

$$\omega = \frac{\dot{m}_w}{\dot{m}_{air}} = \frac{18 \times p_{W_{inlet}}}{28.97 \times P_{air}} = 0.622 \frac{p_{W_{inlet}}}{P_{air}} \quad (6.37)$$

$$\dot{m}_w = 0.622 \frac{p_{W_{inlet}}}{p_{inlet} - p_{W_{inlet}}} \dot{m}_{air,inlet} \quad [kg/s] \quad (6.38)$$

$$\dot{m}_{water,prod} = 9.34E - 8 \times \frac{P_{total}}{V_{cell}} \quad (6.13 \text{ revisited})$$

For the fuel cell model used, the amount of air used during cruise is around 0.326 kg/s and the fuel cell inlet pressure is 250 kPa. Moreover, the preferred inlet relative humidity and temperature are 40% and 353 K respectively. Using the methodology addressed earlier, the fuel cell exit pressure is equal to 192.70 kPa and the amount of water needed to keep the fuel cell membrane humid is equal to 0.017 kg/s. It should be noted that the amount of water produced during cruise is around 0.043 kg/s. Therefore, the amount of water that should be recycled is approximately 39% of the produced water, given an inlet relative humidity of 40%.

### 6.2.3. THERMAL MANAGEMENT SYSTEM

The master's thesis made by Vonhoff [78] looked into the design of a thermal management system for PEM fuel cells in some detail. His design methodology is based on Chapman et al. [83] work, where they presented the thermal system design of a turboelectric 15 passenger VTOL aircraft. The thermal management system covered is using liquid cooling and a ducted heat exchanger as means of heat transfer. In this work, the focus is directed towards determining the power of the cooling fan and coolant pump which can be calculated using Equation 6.39 and Equation 6.40. These equations are adapted on correlations made by Chapman et al. [83]. The wasted heat  $\dot{Q}_{wastedheat}$  from the fuel cell is calculated using Equation 6.41 in which it considers the fuel cell stack efficiency. It must be pointed out that a zero fuel cell stack power means that there will be no wasted heat. However, a zero wasted heat does not result in a zero cooling system power which is not possible in reality since the fuel cell is not operating. This is fixed by implementing a conditional if statement which equates the cooling system power to zero if the fuel cell power is generating no power.

$$P_{CS} = (0.371\dot{Q}_{wastedheat} + 1.33) f_{dt} \quad (6.39)$$

$$f_{dt} = 0.0038 \left( \frac{T_0}{(T_{FC} - T_0)} \right)^2 + 0.0352 \left( \frac{T_0}{(T_{FC} - T_0)} \right) + 0.1817 \quad (6.40)$$

$$\dot{Q}_{wastedheat} = \left( \frac{E_0}{V_{cell}} - 1 \right) P_{FC} = \left( \frac{1}{\eta_{voltage}} - 1 \right) P_{FC} \quad (6.41)$$

### 6.3. THE TURBO-GENERATOR

#### 6.3.1. GAS TURBINE SIMULATION PROGRAM (GSP)

GSP is an object oriented component based modelling environment developed by the TU Delft and NLR [84, 79]. It is used to simulate and analyse the turbo-shaft engine performance used in this work. The turbo-shaft model is composed of sub-component models stacked together, in which each component is represented by a set of performance maps and characteristic equations. The model as illustrated in Figure 6.6 has an inlet, compressor, combustor, turbine and exhaust. The two additional components are the fuel control where the fuel type and mass flow rate are adjusted and the operating envelope scheduler. The operating envelope in Figure 6.7 is adjusted according to the eVTOL flight envelope, where the minimum altitude, minimum and maximum speeds are set to 4000 m, 40 m/s and 160 m/s as shown in Figure 6.8. This component also allows the addition of a series of data points from the control's input transient table as shown in Figure 6.9 where the turbine inlet temperature  $T_4$  is varied from 500 to 1250 K. At each flight and ambient condition, simulations are run for all the power settings specified which in this case the turbine inlet temperature range. These data points are marked with a star in Figure 6.7.

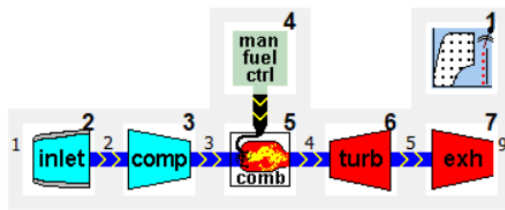


Figure 6.6: GSP model of a turbo-shaft [85].

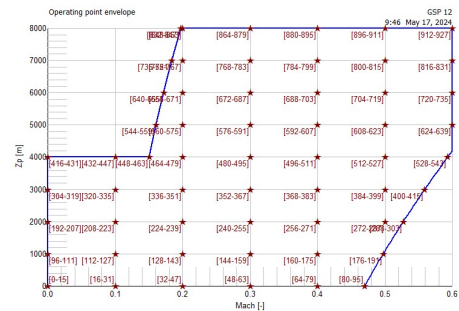


Figure 6.7: Flight envelope including the series data points used as an input for GSP

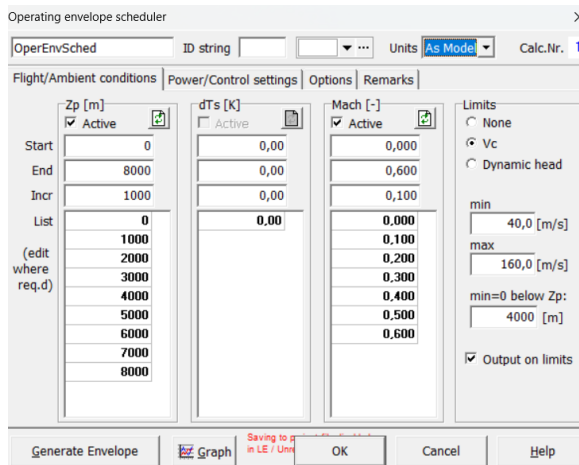


Figure 6.8: The Flight/Ambient condition tab sheet with the altitude, Mach number and limits specified.

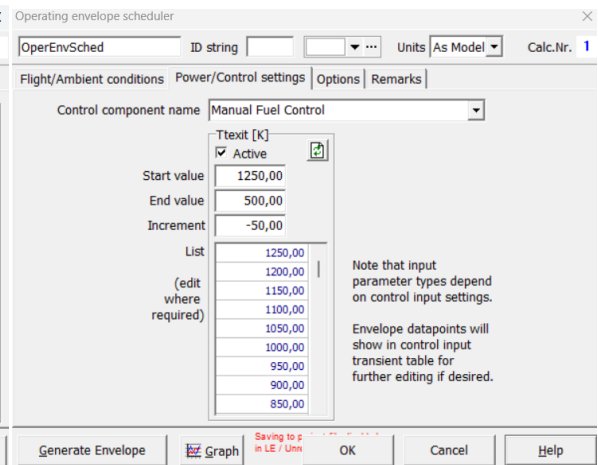


Figure 6.9: The Power/Control settings tab sheet with the turbine inlet temperature values presented.

Now that the boundary conditions are specified, steady state operating point calculations are carried out. GSP starts with an initial guess for the engine's operating point, including variables such as air mass flow rate, pressure ratios, and temperatures at key stations within the engine [79]. The model is composed of a set of non-linear differential equations representing the mass, energy, and momentum conservation across the engines components. It should be noted that since the simulations carried out are steady-state, the time derivatives are set to zero.

The set of equations are solved in an iterative numerical manner employing the multi-variable Newton-Raphson method in which it tries to find a solution with the minimum residual error. For each iteration, the components state variables  $S$  are updated, using the inverse Jacobian matrix derived from the partial deriva-

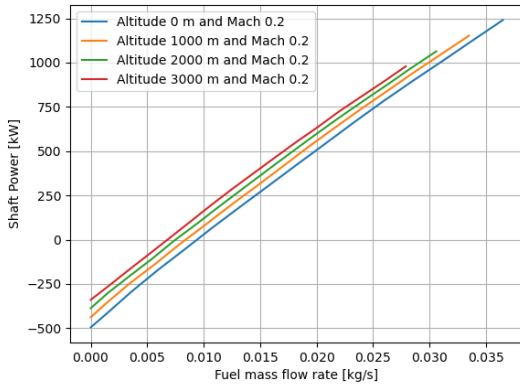


tives of the components error variables  $E$ , with respect to the state variables  $S$  as shown in Equation 6.42. Equation 6.43 is used to update the components state variables until all elements in the error vector are smaller than the tolerance level which are in the range of  $10^{-6}$  and  $10^{-8}$ . It is worth noting that  $f$  in Equation 6.43 is a factor used to limit the magnitude of the correction steps and therefore improve the convergence stability. In case the simulation did not converge, GSP usually adjusts the initial guess or modify the component performance characteristics to stabilize the solution.

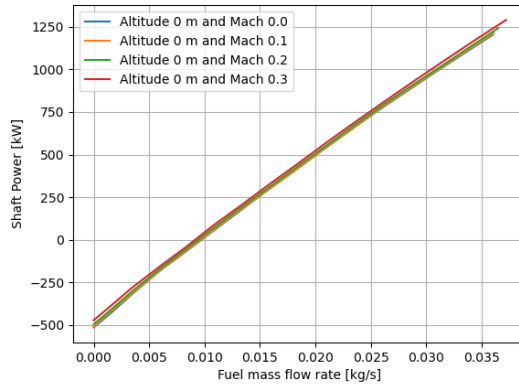
$$\Delta E = J \cdot \Delta S \quad (6.42)$$

$$S_{i+1} = S_i + f \cdot J^{-1} \cdot E_i \quad (6.43)$$

### 6.3.2. GSP OUTPUT DATA PROCESSING



**Figure 6.10:** Shaft power output vs fuel mass flow rate for altitudes 0 to 3000 m. The mach number is kept constant to show the effect of increasing the altitude.



**Figure 6.11:** Shaft power output vs fuel mass flow rate for 0 m altitude. The mach number is varied to show the effect it has on the line trend.

2D graphs are produced to clearly show the effect of increasing the altitude and Mach number. Figure 6.10 shows that increasing the altitude reduces the amount of fuel used and the available shaft power. This is due to the reduction of the air density, pressure and temperature which decrease the engine shaft power and fuel consumption. Figure 6.11, illustrates the effect of increasing the Mach number on the available shaft power and fuel consumption. The available shaft power and fuel consumption increases because of the increase in the ingested air mass flow rate, pressure and density.

Fuel mass flow rate  $\dot{m}_f$  and shaft power  $P_{TG}$  are the required output of this simulation in which it requires three input variables: Altitude  $h$ , Mach number  $M$  and turbine inlet temperature  $T_4$ . The discrete data points can be interpolated such that it can be used to predict  $\dot{m}_f$  and  $P_{TG}$  given any set of values for  $h$ ,  $M$  and  $T_4$ . In that case 3D linear interpolation is used which is carried out using the Scipy python library. The 3D linear interpolation in the library uses an extension version of 2D Delaunay triangulation. Delaunay triangulation (DT) creates a mesh of triangles from the data points and these triangles are used to interpolate the values within its area as illustrated in Figure 6.12. The reason behind the usage of the Delaunay triangulation (DT) method is that it tries to create equatorial triangles as shown on the left of Figure 6.12. In this way, the interpolated results will be closer to the vertices of the triangles and thus producing more accurate results. This interpolation method is considered a weighted-average interpolation method which uses Equation 6.44 where  $w_i(x)$  is the weight of the neighbour points  $p_i$  and  $a_i$  are the attributes of these points  $p_i$  [86].

$$f(x) = \frac{\sum_{i=1}^k w_i(x) \cdot a_i}{\sum_{i=1}^k w_i(x)} \quad (6.44)$$

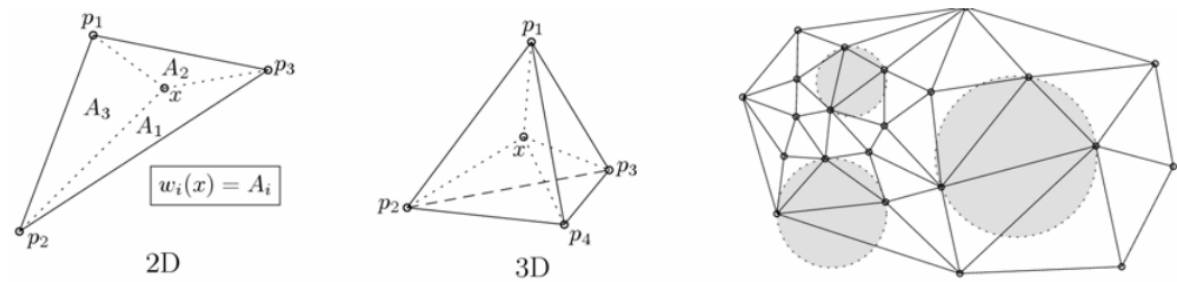


Figure 6.12: Left: barycentric coordinates in 2D and 3D planes. Right: Delaunay triangulation in 2D [86].

#### 6.4. PRELIMINARY WEIGHT ESTIMATION

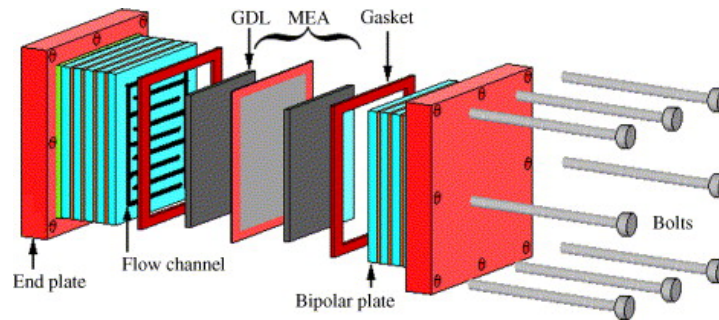


Figure 6.13: The PEM fuel cell in an exploded view [87]

The methodology used by Daniel Juschus [88] is used to estimate the fuel cell stack mass. The structure of the PEM FC can be visualised in Figure 6.13. The membrane electrode assembly is sandwiched between the anode and cathode electrode and it is considered the core of the FC. The membrane is assumed to be made of Nafion with an area density of  $0.2 \text{ kg/m}^2$  [89]. The MEA is positioned between gas diffusion layers to help distribute the gasses evenly. The bipolar plates has a similar function but it is also used to provide structural support and remove the heat generated during the reaction, by facilitating the flow of the coolant through its channels. Moreover, the gaskets main goal is to ensure that the pressure within the stack remains the same and that the reactant gasses do not leak out of the flow channels. The stack is compressed with end plates and bolts to further improve the structural integrity of the stack and prevent any leakage between other components.

The material used for all components except the MEA is assumed to be 304L Stainless Steel with a density of  $8000 \text{ kg/m}^3$  [90]. Furthermore, the thickness for bipolar plates and end plates are assumed to be equal to 0.2 and 0.25 mm respectively based on values from literature [91, 92, 93]. For simplicity the mass of the bolts and gaskets are neglected. The values used to estimate the mass is summarized and presented in Table 6.1. Equation 6.45, Equation 6.46 and Equation 6.47 are therefore when combined can estimate the fuel cell stack mass as shown in Equation 6.48. Furthermore to estimate the hydrogen gas tank, the study made by P. Muthukumar et al.[94] is used. They studied three tank types in their which are depicted in Table 6.2 together with the turbo-generator power density.

MEA Area Density [89]	$0.2 \text{ kg/m}^2$
Material for Other Components [90]	304L Stainless Steel
Density of 304L Stainless Steel [90]	$8000 \text{ kg/m}^3$
Bipolar Plates Thickness [91, 92, 93]	0.2 mm
Endplates Thickness [91, 92, 93]	25 mm

Table 6.1: Summary of variables and materials used to estimate the fuel cell stack mass

$$m_{MEA} = \rho_{MEA} \cdot A_{cell} \quad (6.45)$$

$$m_{BP} = n_{cells} \cdot A_{cell} \cdot t_{BP} \cdot \rho_{BP} \quad (6.46)$$

$$m_{EP} = 2 \cdot A_{cell} \cdot t_{EP} \cdot \rho_{EP} \quad (6.47)$$

$$m_{stack} = A_{cell} [n_{cells} (t_{BP} \rho_{BP} + \rho_{MEA}) + 2t_{EP} \rho_{EP}] \quad [kg] \quad (6.48)$$

Parameter	Value
Turbogenerator power density [21] [ $MW/kg$ ]	0.00435
Type-I Hydrogen gas tank capacity at 200 bar [94] [ $kg_{H_2}/kg$ ]	1.1
Type-II Hydrogen gas tank capacity at 300 bar [94] [ $kg_{H_2}/kg$ ]	2.1
Type-III Hydrogen gas tank capacity at 700 bar [94] [ $kg_{H_2}/kg$ ]	4.21
Type-IV Hydrogen gas tank capacity at 700 bar [94] [ $kg_{H_2}/kg$ ]	5.7

**Table 6.2:** Relevant densities used in preliminary estimate the HPS weight

## CASE STUDY IMPLEMENTATION & RESULTS

This chapter will cover the application of the methodology outlined in previous chapters to an electrical vertical take off and landing (eVTOL) aircraft with the flight mission shown in Figure 5.1. The flight data used is summarised in Table B.1. In section 7.1, an initial fuel cell design point is used to simulate the hybrid propulsion system (HPS) operation through the flight mission. In section 7.2, multiple design points are tried and the model is used to find an optimum payload mass based on the resultant HPS mass.

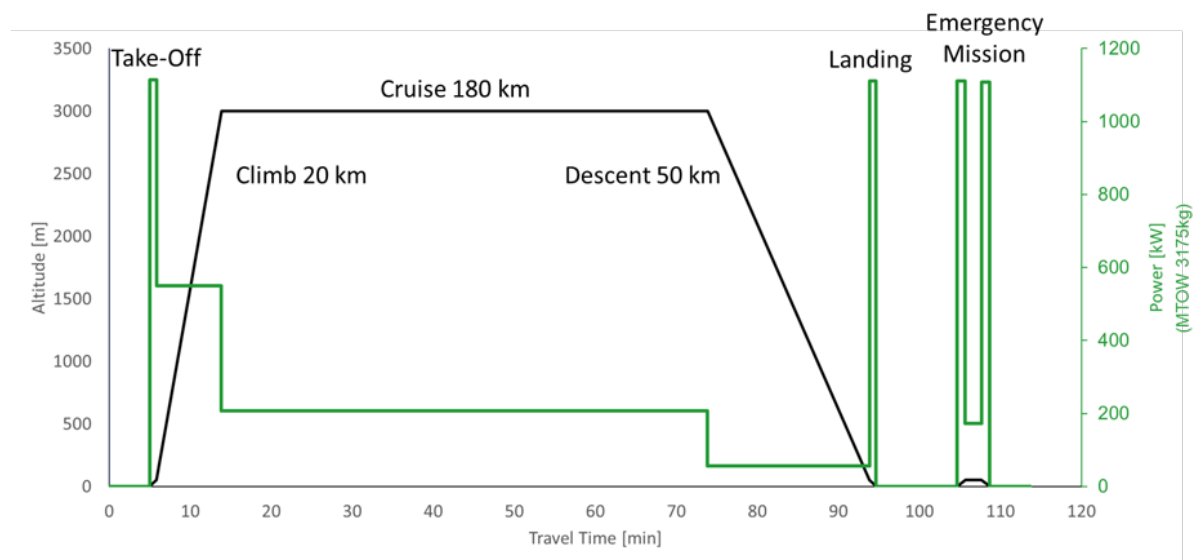


Figure 5.1 revisited, eVTOL flight mission and power requirement versus time.

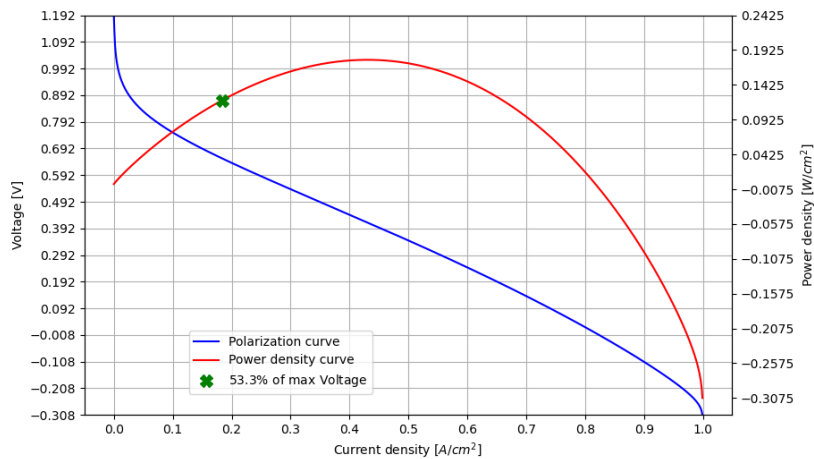
### 7.1. MISSION ANALYSIS WITH INITIAL DESIGN ASSUMPTION

#### 7.1.1. FUEL CELL ON DESIGN MODEL

**eVTOL cruise design point requirements**

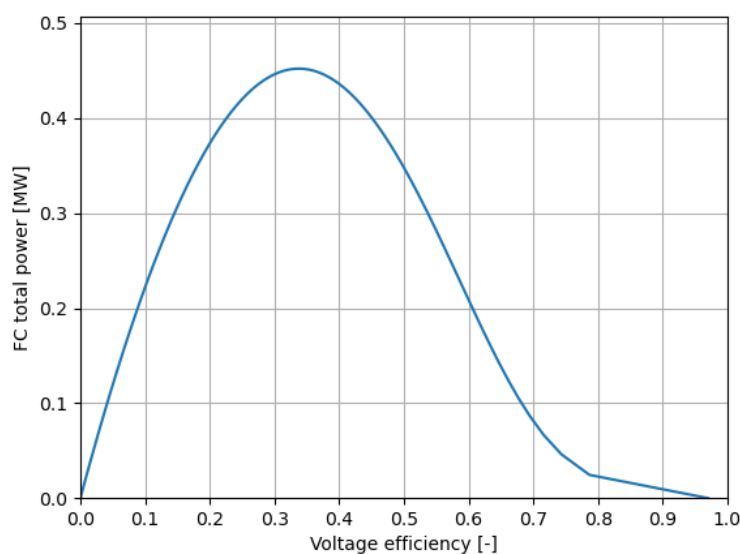
Required power [kW]	206.6
Fuel cell operating temperature [K] [3]	353.15
Fuel cell cathode operating pressure [bar] [4]	2.5
Fuel cell anode operating pressure [bar] [4]	2.53
Water mole ratio to membrane sulfonic group $\lambda_{water}$ [82]	14
Air stoichiometric condition [4]	2
Compressor pressure ratio	3.5
Number of stacks	2
Fuel cell membrane thickness [cm][95]	0.08
Total voltage requirement [V]	800

Table 7.1: The fuel cell design point requirements used to estimate the needed number of cells and fuel cell area.



**Figure 7.1:** The fuel cell operating point during cruise at 53% voltage efficiency

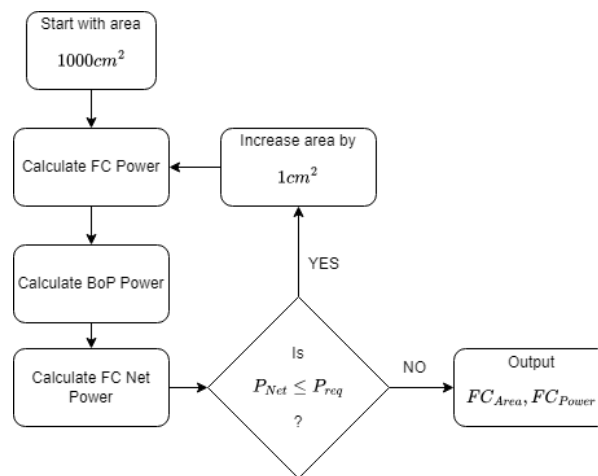
The fuel cell system is designed to operate during cruise because it is the longest in the flight mission. Parameters included in [Table 7.1](#) are the input used by the tool to guide the initial design of the fuel cell system. The tool initially draw the fuel cell polarization curve as depicted in [Figure 7.1](#) which, is used as a map to obtain the cell voltage  $V_{cell}$  and the corresponding  $P_{dens}$  at a known voltage efficiency. [Figure 7.2](#) is also used to determine the corresponding fuel cell power given an operating fuel cell voltage efficiency. [Equation 7.1](#) is then used to calculate the number of cells required to deliver the required voltage. With the number of cells  $n_{cells}$ , the cell area is determined by using the loop shown in [Figure 7.3](#). The loop starts with an initial area and use [Equation 7.2](#) to calculate the fuel cell gross power. However, the calculated gross power might not be enough to support both the propulsion system and the balance of plant components. It is also worth noting that the power for the balance of plant components is dependant on the calculated gross power, therefore new power for the balance of plant components is calculated and subtracted from he gross power. The loop keeps adding  $1 \text{ cm}^2$  to the area until the computed net power is more than or equal to the required power specified by the user.



**Figure 7.2:** Fuel cell total power versus the fuel cell voltage efficiency.

$$n_{cells} = V_{req}/V_{cell}/n_{stack} \quad (7.1)$$

$$P_{FC} = P_{dens} \times A \times n_{cells} \times n_{stacks} \quad (7.2)$$



**Figure 7.3:** The loop used to calculate the required fuel cell area for the on design model

**Table 7.2** provides a comprehensive overview of the performance and characteristics of a fuel cell system. The first section details the mass flow rates of the different species involved in the electrochemical reaction process. Inlet air enters the system at a rate of  $0.3176 \text{ kg/s}$ , while the fuel cell consumes oxygen and hydrogen at rates of  $0.03684 \text{ kg/s}$  and  $0.00464 \text{ kg/s}$ , respectively. After the reaction,  $0.2807 \text{ kg/s}$  of air exits the system, and  $0.04148 \text{ kg/s}$  of water is produced as a byproduct.

The second section outlines the power consumption and generation of the system components. The cooling system, which is essential for maintaining optimal operating temperatures, consumes  $32.759 \text{ kW}$  of the power. The compressor, which supplies air to the fuel cell, requires  $51.556 \text{ kW}$ . The fuel cell itself generates a gross power of  $290.941 \text{ kW}$ ; however, after accounting for the power consumed by auxiliary components, the net power available for use is  $206.626 \text{ kW}$ . Additionally, the system produces  $265.916 \text{ kW}$  of waste heat, which must be managed to ensure efficiency and safety.

The third section describes the fuel cell's physical and operational parameters. The cell operates at a voltage of  $0.6547 \text{ V}$  and has a power density of  $0.1204 \text{ W/cm}^2$ . The system comprises 611 individual cells with a total active area of  $1976.5 \text{ cm}^2$ . The fuel cell achieves a voltage efficiency of  $53.3\%$ , a thermodynamic efficiency of  $83.0\%$ , and a total efficiency of  $44.2\%$ , reflecting the overall effectiveness of the fuel cell in converting hydrogen and oxygen into electrical energy. It is worth noting that the fuel utilisation efficiency  $\epsilon_{fuel}$  is assumed to be  $100\%$ .

Reactants/Products Mass flow rate [kg/s]	
Inlet Air	0.3176
Oxygen used	0.03684
Hydrogen used	0.00464
Outlet Air	0.2807
Water produced	0.04148
Components Power [kW]	
Cooling system	32.759
Compressor	51.556
Fuel cell gross power	290.941
Fuel cell net power	206.626
Fuel cell wasted heat	265.916
Fuel cell sizing	
Cell voltage [V]	0.6547
Power density [ $W/cm^2$ ]	0.1204
Number of cells [-]	611
Area [ $cm^2$ ]	1976.5
Fuel cell voltage efficiency [%]	53.3
Fuel cell thermodynamic efficiency [%]	83.0
Fuel cell total efficiency [%]	44.2

**Table 7.2:** On design characteristics at 53 % voltage efficiency

### 7.1.2. INITIAL POINT OFF-DESIGN MISSION SIMULATION

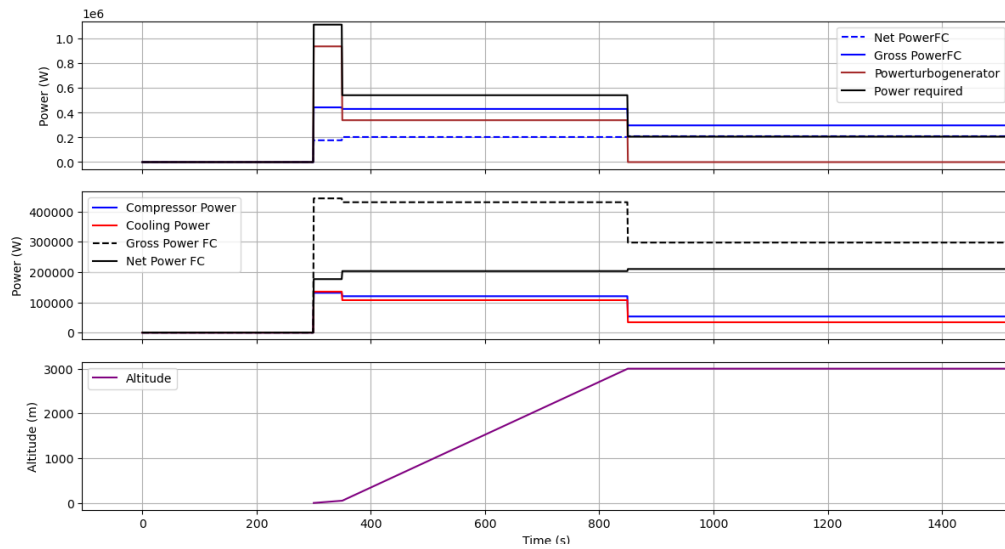
The designed fuel cell is used in the off-design performance simulation of the eVTOL hybrid propulsion system. The off-design scenarios follow the mission flight phases in Figure 5.1.

Figure 7.4 and Figure 7.5 shows the power division between the fuel cell and the turbogenerator through the flight to deliver the required power output. There are two fuel cell power lines: gross and net power. The fuel cell net power represented by the dashed line is the net power delivered to the electric motor of the propeller whereas the sharp line represent the fuel cell total power which includes the power delivered to the balance of plant components: the compressor and the cooling system. There is a sharp increase of around 1.1 MW in the power required during take-off which in result increased the demand from the fuel cell. The fuel cell net power during take-off can only deliver up to 0.18 MW and consequently the turbo-generator had to deliver the power deficiency of approximately 0.92 MW. For climb, the power demand drops to around 0.55 MW. The fuel cell net power only supply around 0.2 MW without supplying the total required power and the turbo-generator is used once again to supply the power deficiency of approximately 0.35 MW. For cruise, the power demand drops to approximately 0.2 MW which is the target power for the on-design. Therefore, the fuel cell managed to deliver the power demand as expected. During descent the required power drops to around 0.08 MW which was also doable by the fuel cell alone. During landing and loitering a similar scenario to the take-off is repeated and the turbo-generator was turned on to supply the power deficiency.

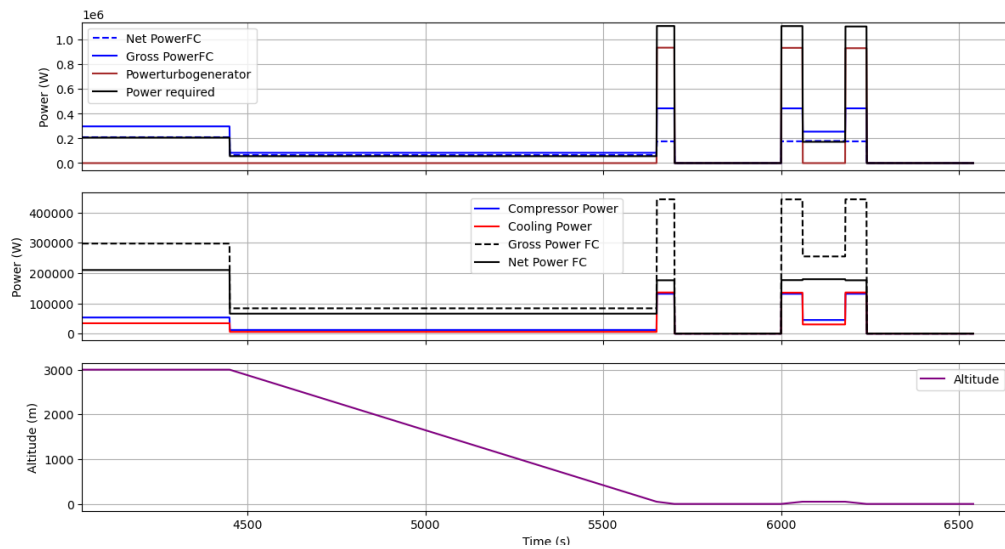
Figure 7.4 and Figure 7.5 also further detailed the power division, where the power consumed by the compressor and cooling system are included in a separate graph. As mentioned earlier in the previous paragraph, the fuel cell gross power included the fuel cell net power, the compressor and the cooling system power. The compressor power  $P_{comp}$  is affected by the ambient temperature, pressure and flight speed. It is the highest at lower altitudes because of the higher ambient temperature. The cooling system power  $P_{CS}$  is dependant on the temperature gradient between the ambient and fuel cell operating temperature. The closer the ambient temperature to the fuel cell operating temperature, the higher the required cooling power. It is also dependant on the fuel cell wasted heat which increases by increasing the fuel cell power output.

The current variation and the species mass flow rates which includes hydrogen and oxygen consumption throughout the mission are plotted in Figure 7.6 and Figure 7.7. In off-design, the model chooses the lowest and nearest current that produces a fuel cell net power closer to the required input power. If the model reaches the maximum power density corresponding current, and the required power is not yet reached, power deficiency is calculated. The model then calculates the turbogenerator power that can deliver the power deficiency. In Figure 7.6 and Figure 7.7, the maximum power density corresponding current is reached during take-off, climb and the emergency mission.

It is worth noting that the mass flow rates are directly proportional to the current which explains the



**Figure 7.4:** Detailed power division of the hybrid propulsion system during take off and descend, including the balance of plant components power demands.



**Figure 7.5:** Detailed power division of the hybrid propulsion system during approach and emergency mission, including the balance of plant components power demands.

similarity in the trend between both graphs. The total amount of hydrogen used will be further analysed in [section 7.2](#) to calculate the tank mass.



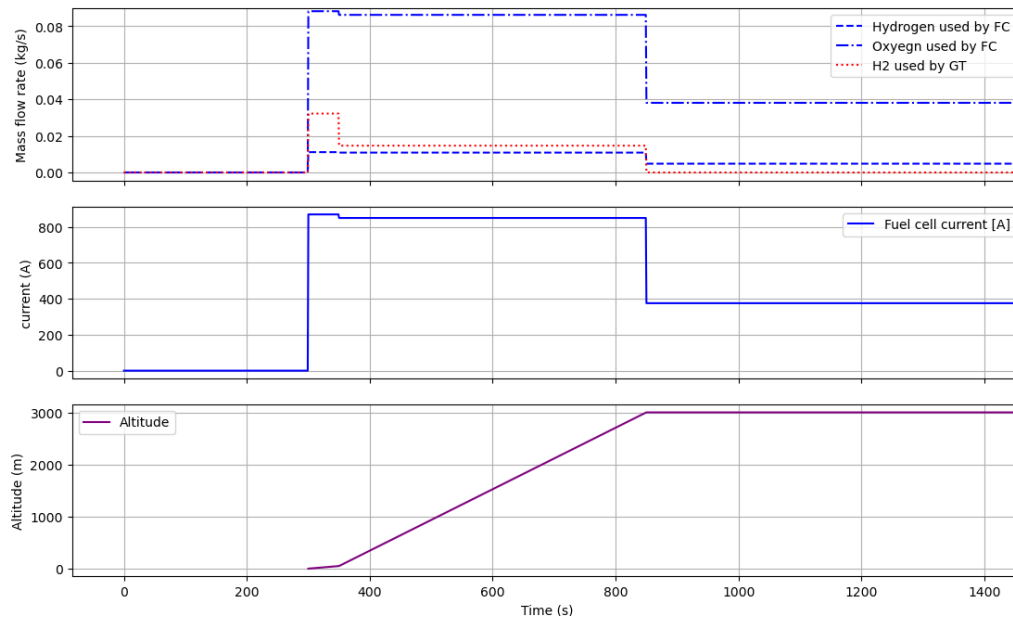


Figure 7.6: Species mass flow rates during the eVTOL take-off, descent and the corresponding current variation.

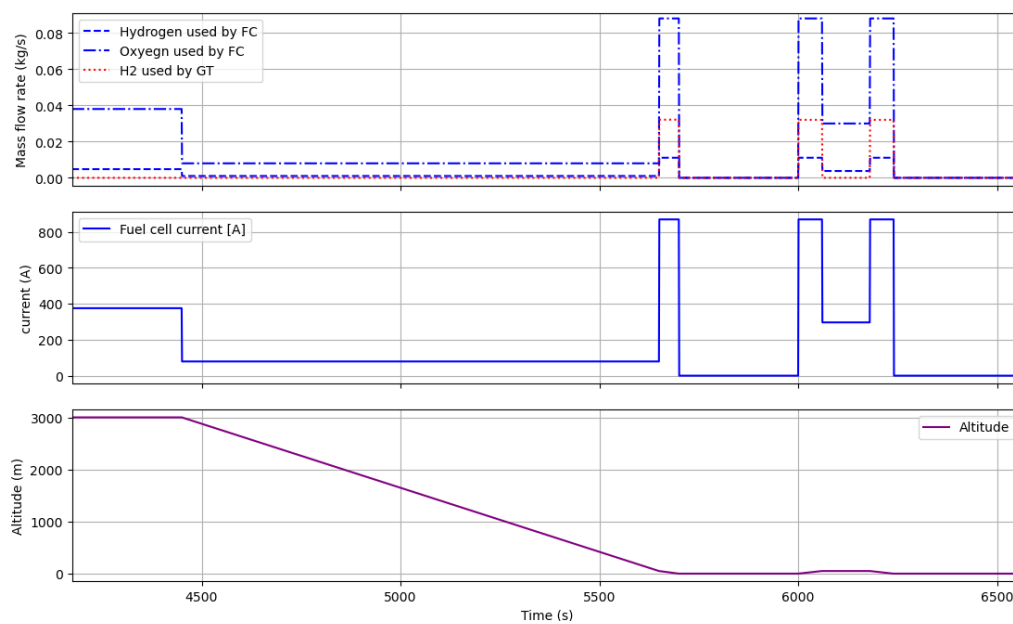
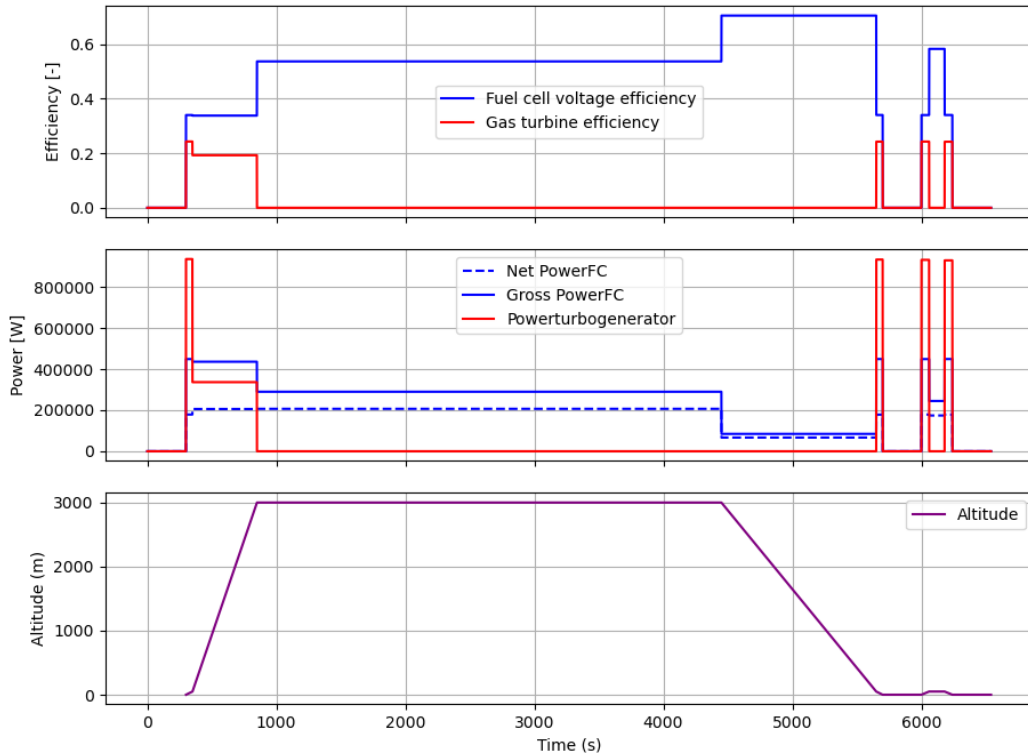
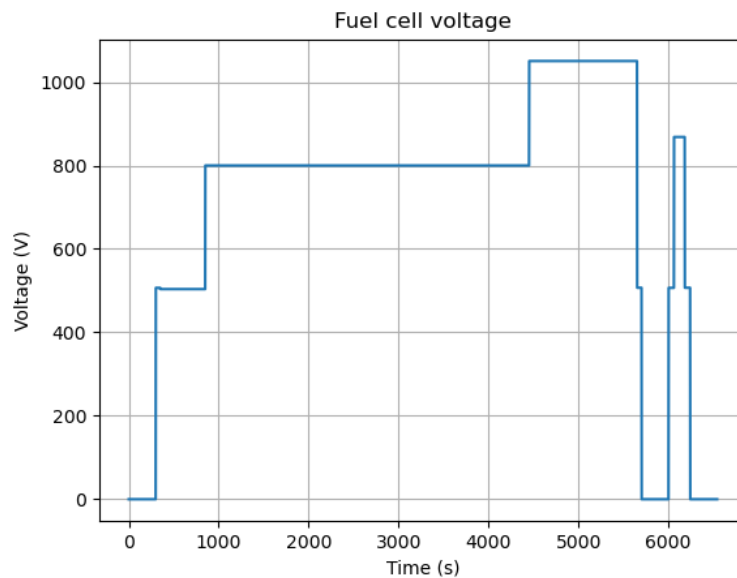


Figure 7.7: Species mass flow rates during the eVTOL approach, emergency mission and the corresponding current variation.

The fuel cell voltage and turbo-generator efficiencies are illustrated in Figure 7.8, corresponding to the power division between both propulsion systems. It is important to note that the fuel cell voltage and turbo-generator efficiencies are taken as zero when they are not operating.



**Figure 7.8:** eVTOL flight mission power requirement HPS fuel cell & gas turbine efficiencies. It should be noted that efficiencies are zero when the corresponding power requirement is zero.

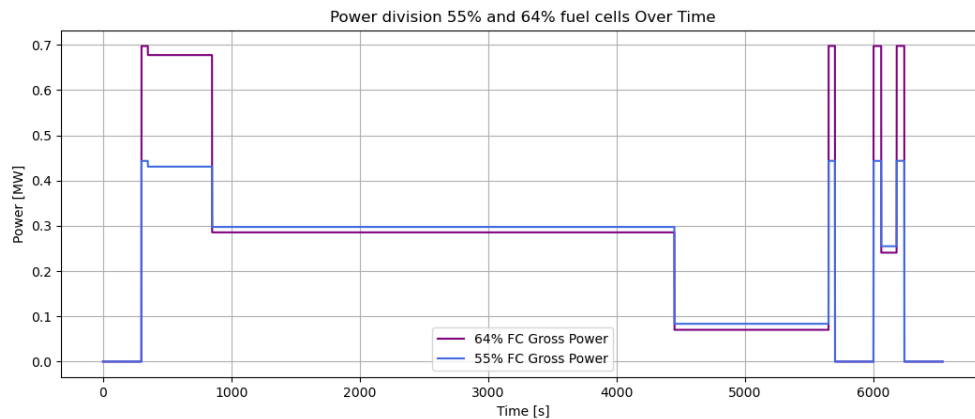


**Figure 7.9:** Fuel cell voltage variation with respect to the flight mission. Voltage is taken as zero when the power requirement is zero.

The fuel cell was designed to supply the propulsion system with 800 volts. This can be seen in [Figure 7.9](#) where the 800 volts are met during cruise. This is a method of verification to ensure that the tool is working as intended in the off-design condition. Moreover, the voltage is highest during descent, because of the low power requirement at this stage of the flight. As explained earlier, based on the power requirement, the current is varied, changing the fuel cell voltage as a result.

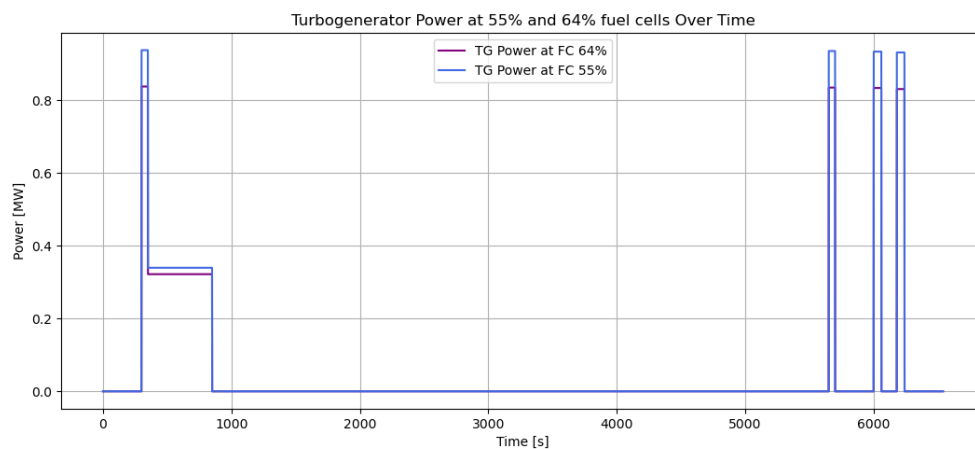
It can be concluded from this section that, the power calculated from the fuel cell depends on a lot of variables such as the power requirement, the fuel cell area, number of cells, compressor, cooling system power, and altitude. The fuel cell net power during cruise supply the power demand without the need for the turbogenerator assistance. However, it reached its maximum power at some of the mission phases of the eVTOL such as take-off and descent in the beginning of the flight mission. This raises the question of what if the fuel cell was designed at a different design point than the 53% point chosen in that section? This will be answered in [section 7.2](#).

## 7.2. CASE STUDY AND DESIGN OPTIMISATION

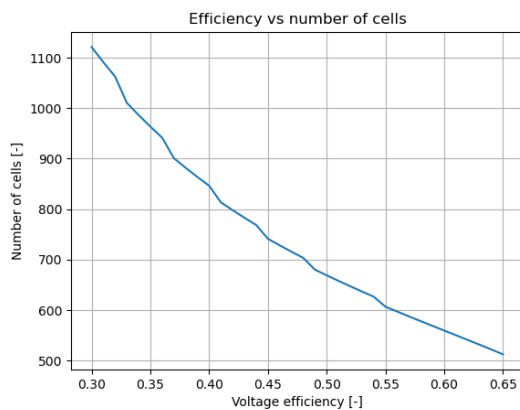


**Figure 7.10:** The resultant fuel cell gross power when designing the fuel cell at voltage efficiencies of 55% and 64%.

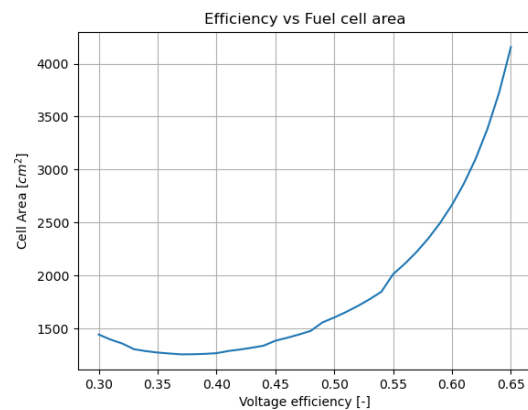
In this section, the fuel cell is designed at different voltage efficiencies, which changes the point location on the power density, polarization curve. Changing the design point, will change the fuel cell area and the number of cells. As a result, the fuel cell power throughout the flight mission will change, changing the turbogenerator produced power as a result. This effect is illustrated in [Figure 7.10](#), where a comparison is made between the fuel cell total power when designed at 64% and 55%. It is worth noting that operating at 64% means that the point is further back on the polarization curve. The fuel cell area and number of cells at 55% are  $2013 \text{ cm}^2$  and 607 cells respectively whereas at 64% they are  $3726 \text{ cm}^2$  and 523 cells respectively. This has an effect on the power produced by the turbogenerator as shown in [Figure 7.11](#) where the maximum power produced by the turbogenerator decreased from 0.94 MW to 0.84 MW when the fuel cell area increased.



**Figure 7.11:** Turbogenerator power when operating with fuel cell at voltage efficiencies of 55% and 64%.



**Figure 7.12:** Number of cells at different design points.



**Figure 7.13:** Fuel cell at different design points.

The fuel cell weight is dependant on cell area and the number of cells. By changing the fuel cell voltage efficiency, the cell area and the number of cells required to deliver the required power changes. Therefore, the model is used to run through various design points where the fuel cell voltage efficiency is varied between 30% and 66%. [Figure 7.12](#) depicts the correlation between voltage efficiency and the number of cells in the fuel cell stack. The trend in the graph indicates a clear inverse relationship: as voltage efficiency increases, the number of cells decreases. This suggests that higher efficiencies can be achieved with fewer cells, due to the higher operating voltages, reducing the need for additional cells to meet voltage requirements. Furthermore, [Figure 7.12](#) shows a steady decline in the number of cells as efficiency increases from 0.30 to 0.65, highlighting how advancements in cell efficiency can lead to a more compact fuel cell. [Figure 7.13](#) on the other hand, illustrates the relationship between voltage efficiency and the fuel cell's surface area. As observed in the graph, there is a notable trend where the cell area initially decreases slightly as the voltage efficiency increases, reaching a minimum at an efficiency around 0.35 to 0.40. Beyond this point, the cell area begins to increase significantly with rising voltage efficiency, showing an exponential growth pattern. This suggests that higher voltage efficiencies require a substantially larger cell area to maintain, due to the need for more active surface area to support efficient chemical reactions within the fuel cell.

With the dependency of the number of cells and cell area on the voltage efficiency analysed, it is possible to estimate the fuel cell, gas turbine (GT) masses and thus the payload at different fuel cell voltage efficiencies. From [Figure 7.14](#), it is found that the possible minimum fuel cell is around 477 kg at a fuel cell operating voltage efficiency of approximately 43%. The corresponding mas of the gas turbine is equal to 221.53 kg resulting in a total available payload of 174.03 kg. At efficiencies lower than 43%, the fuel cell mass is higher because of the large area and high number of cells. The fuel cell gets heavier at efficiencies larger than 43% due to the increase in the area and the power demand from the balance of plant components as shown in [Figure 7.15](#). It is also observed that at efficiencies between approximately 40% and 45%, the gas turbine mass is the heaviest. In general, the gas turbine weight is inversely proportional to the fuel cell weight curve, since it has to compensate for the lower power produced by the fuel cell. The bottom graph examines the available payload mass, which is not only affected by the fuel cell and gas turbine mass but also the compressor, cooling system, hydrogen fuel used and gaseous hydrogen tanks. At efficiencies between 30% and approximately 34% , the available payload mass is negative, indicating an excess weight beyond the eVTOL's capacity. As efficiency improves, the available payload mass increases, becoming positive around an efficiency of 36%, and peaks at 174.03 kg at 43% efficiency. Beyond this peak, the available payload mass decreases, approaching negative values again at the highest efficiencies.

To better understand the trends in [Figure 7.14](#), [Figure 7.15](#) is generated. . [Figure 7.15](#) depicts how the voltage efficiency impacts the maximum power of the various components of the HPS, with voltage efficiency values ranging from 30% to 65% and maximum power measured in megawatts (MW). At 43% the fuel cell is producing the lowest power, which concurs with [Figure 7.13](#) result, where the fuel cell area is the lowest. During take-off, the turbogenerator is producing its maximum output which decreases at fuel cell voltage efficiencies more than approximately 47%. This is because at take-off the fuel cell delivers more power at high efficiencies and thus the turbogenerator does get smaller. The components weight of the hybrid propulsion system is estimated and depicted in [Figure 7.16](#), where it shows the result of varying the voltage efficiency on the mass of the compressor and cooling system at different design points.

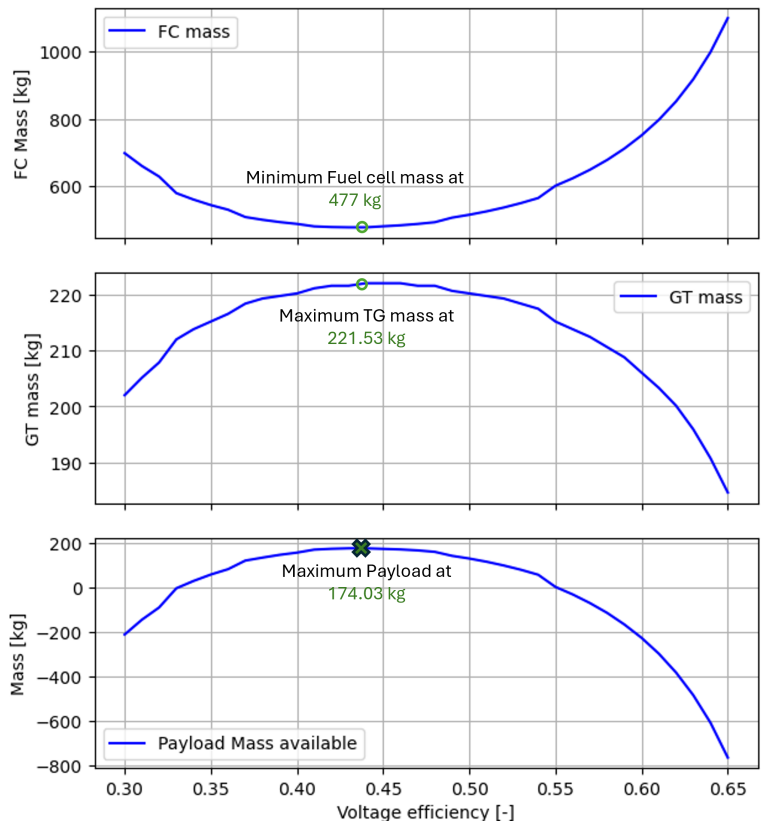


Figure 7.14: Fuel cell, turbogenerator mass and the resultant free payload mass.

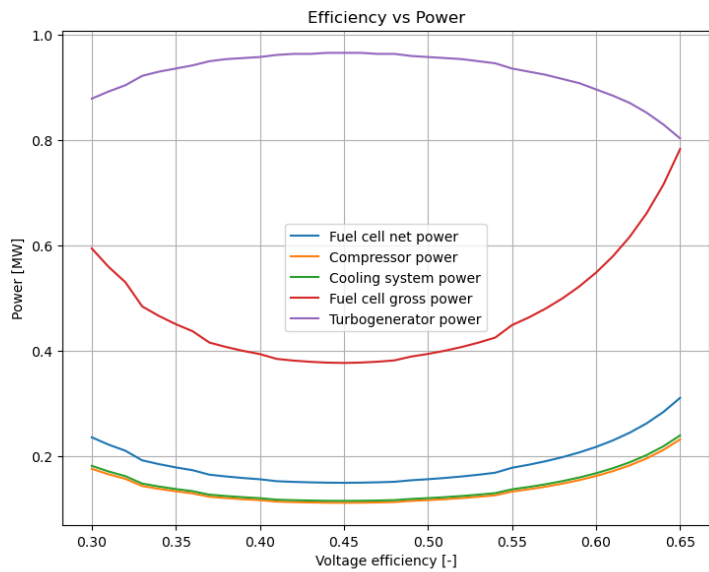


Figure 7.15: Power division of the fuel cell, compressor, cooling system and turbogenerator at different design points.

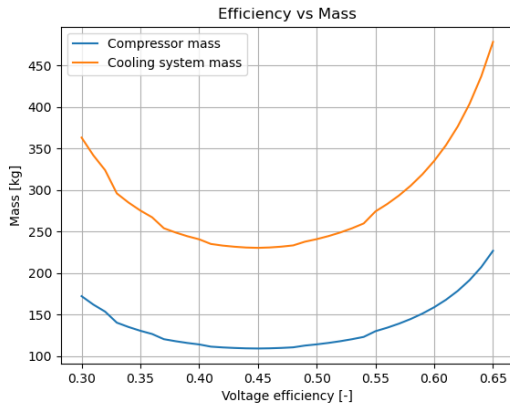


Figure 7.16: The compressor and cooling system weights at different design points.

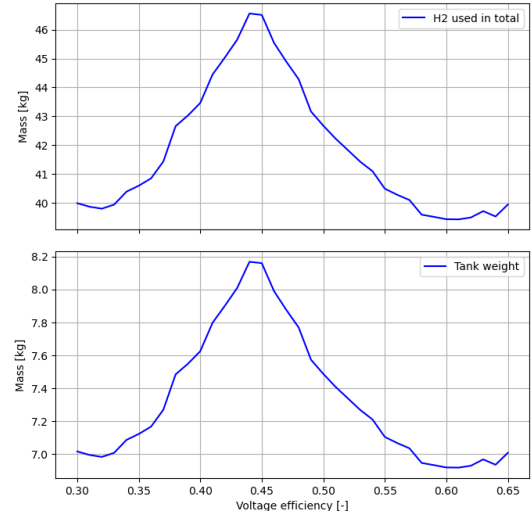


Figure 7.17: Hydrogen consumed and the corresponding tank weight.

The overall trend in Figure 7.15 and Figure 7.16 can be explained by considering that at efficiencies below 43%, the fuel cell produces power comparable to higher efficiencies but requires a higher number of cells. This is due to the power density in the polarization curve declining after reaching a maximum, necessitating an overdesign of the fuel cell with more cells to maintain the required voltage of 800 V. The hydrogen consumed and the resultant fuel tank masses are depicted in Figure 7.17. The hydrogen tank used is the type-IV tank mentioned in Table 6.2, which is the lightest option.

It is important to draw attention to the effect the compressor isentropic efficiency  $\eta_{comp}$  has on the available payload weight. Increasing the compressor efficiency will not only result in a lower compressor exit temperature which will require less cooling, but also an increase in the overall payload weight and vice versa. This is explained by the decrease in the compressor weight and power, decreasing as a result the required fuel cell gross power. The decrease in the fuel cell gross power means more power will be available from the fuel cell to the propeller and thus the turbo generator maximum power and mass will decrease. Moreover, the decrease in the compressor efficiency will result in a small decrease in the amount of fuel used. Table 7.3 shows two cases where two compressor efficiencies were used 70% and 85% and the corresponding weights are tabulated. It should be noted that the maximum take off weight and the structural weight of the eVTOL are 3175 and 1905 kg respectively.

	Fuel cell	Compressor	Cooling System	Turbogenerator	Fuel used	Fuel IV tank	Payload
Weight at compressor of 85 % [kg]	466.55	101.27	226.76	221.08	46.51	8.16	199.67
Weight at compressor of 70 % [kg]	504.74	133.04	245.32	223.36	47.11	8.27	108.17
Change	8.19 %	31.37 %	8.19 %	1.03%	1.30%	1.30%	-45.82%

Table 7.3: Component weight variation at compressor efficiencies of 70% and 85%

# 8

## CONCLUSIONS

The thesis explores a critical challenge in the aviation industry's pursuit of sustainability. Electric propulsion systems are seen as a potential solution to the industry's sustainability issues, and a hybrid propulsion system combining a fuel cell and a turbogenerator, as studied in this thesis, could meet the needs of potential eVTOL customers. AYED-ENGINEERING plans to develop a turbogenerator capable of supplying electric power to eVTOLs or small aircraft. Turbogenerators enable multirotor configurations, facilitating vertical take-off and landing for eVTOLs, while also supporting long cruise flights. However, using a kerosene-powered turbogenerator is not sustainable. Hydrogen could offer a sustainable alternative, but it introduces a new problem: turbogenerators are highly inefficient, leading to high fuel consumption. While this is manageable with kerosene, using hydrogen would require larger, heavier fuel tanks due to its low energy density. To address this, the thesis proposes coupling the turbogenerator with a fuel cell, which is more efficient for long cruise flights. The goal of the thesis is to develop a tool that can calculate the performance of this hybrid propulsion system and provide preliminary estimates of the system's weight.

To develop this tool, a literature study was made on the hybrid propulsion concepts and fuel cell modelling methods. Hybrid architectures are divided into series, parallel, and series parallel. The series is the simplest, most efficient, and lightest. However, it is not as compact as the parallel. Both architectures can be used further when connecting the propulsion system components.

The polymer electrolyte membrane fuel cell (PEMFC) will be used instead of the solid oxide fuel cell due to its lower operating temperature, faster start time, and lighter weight. The fuel cell is expected to operate at higher altitudes where the pressure is not optimal, hence the need of a compressor is essential. The preferred operating pressure of the fuel cell is  $0.25 \text{ MPa}$  and it can be supplied by a radial/centrifugal compressor which is run by an electric motor. It is important to note that the compressor exhaust is cooled down by an intercooler to match the fuel cell operating temperature. Moreover, the heat exchanger will be important to control the temperature of the fuel cell which should be between  $60^\circ \text{ C}$  and  $80^\circ \text{ C}$ . There are many thermal management techniques available in literature, and their application depends on the fuel cell stack output power. Fuel cells with higher output power is better cooled using liquid cooling technique. This method will be best suited to the fuel cell in the propulsion system since it is expected to deliver more than  $300 \text{ kW}$ . Beside a heat exchanger, a humidifier will be needed to manage the fuel humidity levels which should be between 80 % and 100 %. For that, a membrane based humidifier is chosen due to its simplicity, light weight and good performance. The hybrid propulsion system components can thus be concluded and visualised in [Figure 8.1](#).

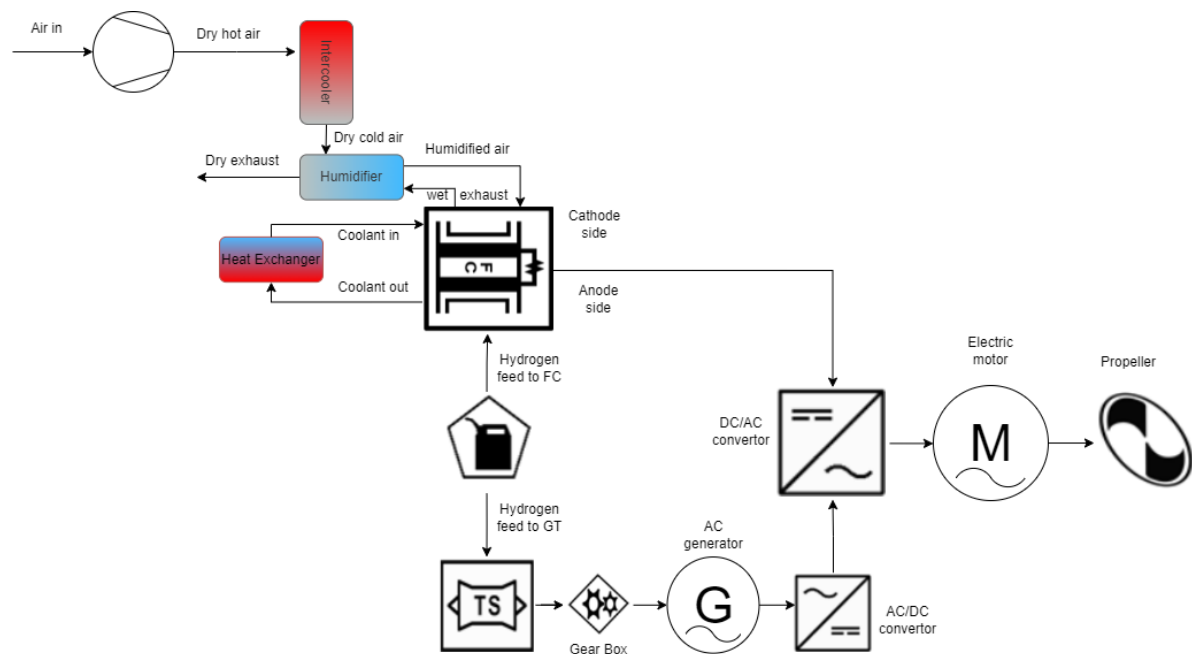


Figure 8.1: Simplified flow diagram example of the hybrid propulsion system.

The tool main input variables are the aircraft altitude, speed, power and voltage requirements. Fuel cell operating conditions such as water content ratio, operating temperature and pressure found in literature are also used as an input to the model. The model border variables are represented by external key variables such as the air ambient conditions and the components power output. The open source PEMFC simulation (OPEM) is used to carry out the fuel cell simulations. It uses Semi-empirical relations used by Amphlett which model the fuel cell in a steady state mode. It is the most precise and detailed when compared to other models such as Larminie-Dicks, and Chamberline-Kim models. This is because it handles the fuel cell physical parameters such as the fuel cell membrane thickness and ionic conductivity. For the balance of plant components (BoP), compressor and humidifier, simple formulas were used that depends on species mass flow rates and the fuel cell operating conditions. The heat exchanger equations were used to determine the power of the cooling fan and coolant pump. These equations are adapted on correlations made by Chapman et al. [83]. In general, the BoP models main output was their power consumption which was used to estimate their weights.

The gas turbine was modelled using the gas turbine simulation program (GSP). The simulation input was based on the eVTOL flight envelope and a range of turbine inlet temperatures which ranged between 500 to 1250 K. 3D linear interpolation was used to interpolate the simulation output data. The functions created by the interpolation takes the mach number, altitude and turbine inlet temperature as an input and gives the fuel mass rate and shaft power as an output.

The fuel cell can be divided into layers of different materials and thicknesses. Using the densities of these materials, the layers thicknesses, the area and number of cells, the fuel cell mass can be estimated. The weight of the rest of the components of the propulsion system was estimated using the calculated power consumption and power densities obtained from literature.

The model is then applied to a case study flight mission of an eVTOL aircraft. The findings indicate that the optimal fuel cell voltage efficiency for maximum payload capacity is around 43%, which effectively balances system weight and efficiency. It is concluded that while higher voltage efficiencies benefit fuel consumption, they also result in heavier fuel cells and related components, thereby affecting overall system weight and aircraft performance. The component weights at 43% voltage efficiency, including those for the fuel cell, compressor, cooling system, and gas turbine, are documented in Table 8.1. The heaviest component is the fuel cell with a weight of 477.23 kg which matches a fuel cell with a power density of approximately  $0.8 \text{ kW/kg}$ . The maximum payload capacity for the mission is found to equal to 174.03 kg, which is low when compared with payloads carried by other eVTOLs such as lilium which can carry around 700 kg<sup>1</sup>.

<sup>1</sup><https://lilium.com/newsroom-detail/certifying-the-lilium-jet-and-its-operations>



<b>Component weights at 43 % voltage efficiency [kg]</b>		
<b>MTOW</b>	3175	
<b>eVTOL Structure</b>	1905	(-)
<b>Fuel cell</b>	477.23	(-)
<b>Compressor</b>	110.06	(-)
<b>Cooling system</b>	231.95	(-)
<b>Gas turbine</b>	221.53	(-)
<b>Fuel used</b>	46.96	(-)
<b>Fuel tank</b>	8.24	(-)
<b>Payload</b>	174.03	

**Table 8.1:** The hybrid propulsion system weights and the resultant maximum payload weight

The model serves as a preliminary tool and there are some recommendations to improve the tool's output. The compressor and cooling system weights were based on linear power to weight ratios found in literature and their weight estimations can be further improved. The compressor can be sized by using velocity triangles, duty coefficients such as the degree of reaction, work and flow coefficients and their relations with the rotor dimensions. Conduction and convection relations can be used to size the intercooler and the cooling system. Moreover, the liquid cooling thermal management system effectiveness should be determined to estimate the efficiency of the cooling system. It is also important to note that humidity calculations uses water partial pressure, fuel cell inlet and outlet air pressures. It did not consider the area of the membrane and thus the amount of water needed to hydrate the membrane can be improved. It was found in this work that increasing the compressor efficiency from 80% to 85% can increase the payload from 174 kg to 199.67 kg. Therefore, it is important to accurately determine the compressor efficiency by using compressor maps. Moreover, one interesting concept to further investigate in future studies is to use the compressor of the gas turbine to also compress the air for the fuel cell instead of the usage of an auxiliary compressor, which will result in an increase in the available payload mass. By getting rid of the compressor, the fuel cell will be lighter since it will not need to deliver power to the auxiliary compressor motor anymore.

Overall, this study provides a framework for modeling and preliminary optimizing hybrid electric propulsion systems for eVTOL aircraft, offering valuable insights that contribute to the advancement of sustainable aviation technologies.

# ACKNOWLEDGMENTS

I am not sure how much does this thesis cost me in terms of money, and mental health. All what I can say is its a journey with more downs than ups. Actually thinking about it, it was all down. I am exhausted mentally but yeah as they say what does not kill you makes you stronger. I am still alive and hopefully I graduate from here. There are people I want to thank, so I have dedicated some time writing this...

Naijb Jahjah, you have been with me the entire masters trip through the downs before the ups. Failing MDO twice together is not the best but it is just a proof that we were together in dark times. Thank you for everything man, and I know you will do an amazing job with your thesis. I am sure of it. Stay strong and remember I am always here by your side.

The W12, Arjun (mijn liefje), Alexi, Easwaar (Eash), Eleni, Keval, Leti (Dr Teez), Tommaso, Luc, and Preateek (P). Thank you guys. Ok thanks is not enough. Without you guys, I dont think I would be able to fit here in the Netherlands. I have learned a lot from you and you have contributed a lot in my development as a person. May God guide you to his righteous path. Thank you again!

Amran, Zayid, Adam, Salma, Ibrahim (mon amie), Yasser, Mohamed Alfie, Savier, Lourd, Zeyaad, Linda, Lilien, Mohamed yaakoub, Mohamed khalifa, Youssef farah and to all my friends and my students who I mentored or I was their TA. You were part of this journey so thank you for everything.

Baroo, also known as akhoya el wazeer. I have been a terrible brother I know. Despite that, you have not abandoned or gave up on me. You took care of me and acted as a second father. Thank you my brother for everything.

The three princesses, Nahar (el dactara), Jana (Bakalezo) and Woud (el atafel). You have been through a lot after our father's loss. I have not been by your sides since then but you add colors to my life. I am very lucky to have you as my sisters and I promise I will do my absolute best to treat you as queens. I love you !

Mama, my true legendary hero. You have sacrificed everything for me and my siblings. Words can not describe you I am afraid. You took care of five orphans, and you never complained or gave up. Thanks for being my mom. Despite the immense financial stress, you kept supporting me and I ask God to provide for you in the afterlife before this life. May God grant you alfiridaws ala'la men elganna, ya ommy!

Papa, 6 years has passed without you around. You never left my mind. I know you are resting now, lucky you but I truly miss you. These years were honestly not the easiest, but they made me a stronger man who you can be proud of. I wish we meet in paradise in the future, but before then, I will keep your promise and take care of my mother and siblings. May god have mercy on your soul !

I can continue writing but I am crying now so later..

# BIBLIOGRAPHY

- [1] . *Vertical Aeropsace: eVTOL company*. 2021. URL: <https://newatlas.com/aircraft/vertical-aerospace-vx4-evtol-reveal/> (visited on 03/19/2024).
- [2] European Environment Agency. *European aviation environmental report 2022*. European Union Aviation Safety Agency EASA, 2022. DOI: [10.2822/129746](https://doi.org/10.2822/129746).
- [3] Ryan O’hayre et al. *Fuel cell fundamentals*. John Wiley & Sons, 2016.
- [4] James Larminie, Andrew Dicks, and Maurice S McDonald. *Fuel cell systems explained*. Vol. 2. J. Wiley Chichester, UK, 2003.
- [5] Rendón et. al. “Aircraft hybrid-electric propulsion: Development trends, challenges and opportunities”. In: *Journal of Control, Automation and Electrical Systems* 32.5 (2021), pp. 1244–1268.
- [6] Christopher Courtin et al. “Feasibility study of short takeoff and landing urban air mobility vehicles using geometric programming”. In: *2018 Aviation Technology, Integration, and Operations Conference*. 2018, p. 4151.
- [7] Roland Berger. “Aircraft electrical propulsion-onwards and upwards”. In: *Think: Act*. July (2018).
- [8] Luke Shadbolt. “Technical study: Electric Aviation in 2022”. In: *HDI Global Specialty SE, April (2022)*. URL: [https://www.hdi.global/globalassets/\\_local/international/downloads/specialty/aviation/e-flight\\_whitepaper.pdf](https://www.hdi.global/globalassets/_local/international/downloads/specialty/aviation/e-flight_whitepaper.pdf).
- [9] Christopher Courtin and R John Hansman. “Safety considerations in emerging electric aircraft architectures”. In: *2018 aviation technology, integration, and operations conference*. 2018, p. 4149.
- [10] Benjamin J Brelje and Joaquim RRA Martins. “Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches”. In: *Progress in Aerospace Sciences* 104 (2019), pp. 1–19.
- [11] Shamsheer S Chauhan and Joaquim RRA Martins. “Tilt-wing eVTOL takeoff trajectory optimization”. In: *Journal of aircraft* 57.1 (2020), pp. 93–112. DOI: <https://doi.org/10.2514/1.C035476>.
- [12] Roland Berger. “Urban air mobility: The rise of a new mode of transportation”. In: *Roland Berger: Focus* (2018).
- [13] NEW ATLAS. *Lilium’s weird, energy-hungry “small fan” design could be a hidden ace*. URL: <https://newatlas.com/aircraft/lilium-interview-small-fans-pros-cons/>.
- [14] Jeremy Kariuki. *Vertical Aerospace Secures \$205 Million Toward eVTOL Development*. URL: <https://www.flyingmag.com/vertical-205m/>.
- [15] Volocopter. *Volocopter urban air mobility*. URL: <https://www.volocopter.com/en/urban-air-mobility>.
- [16] Nicholas Polaczyk et al. “A review of current technology and research in urban on-demand air mobility applications”. In: *8th biennial autonomous VTOL technical meeting and 6th annual electric VTOL symposium*. 2019, pp. 333–343.
- [17] P Nathen et al. “Architectural performance assessment of an electric vertical take-off and landing (eVTOL) aircraft based on a ducted vectored thrust concept”. In: *Lilium GmbH, Claude-Dornier StraeSse, Weßling, Germany, Tech. Rep* (2021).
- [18] Alessandro Bacchini and Enrico Cestino. “Key aspects of electric vertical take-off and landing conceptual design”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 234.3 (2020), pp. 774–787.
- [19] David G Ullman et al. “Comparing electric sky taxi visions”. In: DOI: <https://doi.org/10.13140/RG2.14819.50729> (2017).
- [20] Hidenori Ben, Shinrak Park. *Honda eVTOL vol.2 Hybrid-electric propulsion system “Gas Turbine Hybrid System”*. URL: [https://global.honda/en/tech/eVTOL\\_gas\\_turbine\\_hybrid\\_system/](https://global.honda/en/tech/eVTOL_gas_turbine_hybrid_system/).

- [21] Alessandro Bacchini et al. "Electric VTOL preliminary design and wind tunnel tests". PhD thesis. PhD thesis, Politecnico di Torino, 29.02, 2020.
- [22] Rolls Royce. *Rolls-Royce advances hybrid-electric flight with new technology to lead the way in Advanced Air Mobility*. URL: <https://www.rolls-royce.com/media/press-releases/2022/22-06-2022-rr-advances-hybrid-electric-flight-with-new-technology.aspx>.
- [23] Andreas W. Schäfer et. al. "Technological, economic and environmental prospects of all-electric aircraft". In: *nature energy* 4 (2019), pp. 160–166. URL: <https://doi.org/10.1038/s41560-018-0294-x>.
- [24] Chris Mi and M Abul Masrur. *Hybrid electric vehicles: principles and applications with practical perspectives*. John Wiley & Sons, 2017.
- [25] Engineering National Academies of Sciences, Medicine, et al. *Commercial aircraft propulsion and energy systems research: reducing global carbon emissions*. National Academies Press, 2016. DOI: <https://doi.org/10.17226/23490>.
- [26] John T et. al Economou. "Design of a distributed hybrid electric propulsion system for a light aircraft based on genetic algorithm". In: *AIAA Propulsion and Energy 2019 Forum*. 2019, p. 4305.
- [27] Larissa Lorenz et al. *Hybrid power trains for future mobility*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV Bonn, Germany, 2014.
- [28] Lukic et al. "Effects of drivetrain hybridization on fuel economy and dynamic performance of parallel hybrid electric vehicles". In: *IEEE transactions on vehicular technology* 53.2 (2004), pp. 385–389.
- [29] Isikveren et al. "Pre-design strategies and sizing techniques for dual-energy aircraft". In: *Aircraft Engineering and Aerospace Technology: An International Journal* 86.6 (2014), pp. 525–542.
- [30] XIE Ye, Al Savvarisal, and Tsourdos et. al. "Review of hybrid electric powered aircraft, its conceptual design and energy management methodologies". In: *Chinese Journal of Aeronautics* 34.4 (2021), pp. 432–450. URL: <https://doi.org/10.1016/j.cja.2020.07.017>.
- [31] Mehrdad Ehsani et al. *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018.
- [32] Teresa Donateo, Antonio Ficarella, and Luigi Spedicato. "A method to analyze and optimize hybrid electric architectures applied to unmanned aerial vehicles". In: *Aircraft Engineering and Aerospace Technology* 90.5 (2018), pp. 828–842.
- [33] Ching Chuen Chan. "The state of the art of electric, hybrid, and fuel cell vehicles". In: *Proceedings of the IEEE* 95.4 (2007), pp. 704–718.
- [34] Tyler J Wall and Richard T Meyer. "Hybrid electric aircraft switched model optimal control". In: *Journal of Propulsion and Power* 36.4 (2020), pp. 488–497.
- [35] Joachim Schömann. "Hybrid-electric propulsion systems for small unmanned aircraft". PhD thesis. Technische Universität München, 2014.
- [36] Tyler S Dean, Gabrielle E Wroblewski, and Phillip J Ansell. "Mission analysis and component-level sensitivity study of hybrid-electric general-aviation propulsion systems". In: *Journal of Aircraft* 55.6 (2018), pp. 2454–2465.
- [37] Marty Bradley. "Identification and descriptions of fuel cell architectures for aircraft applications". In: *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*. IEEE. 2022, pp. 1047–1050. DOI: <https://doi.org/10.1109/ITEC53557.2022.9814063>.
- [38] FM.Mulder et al. *SET3085: Hydrogen Technology course material*. The Technical University of Delft. June 2022.
- [39] Arun K Sehra and Woodrow Whitlow Jr. "Propulsion and power for 21st century aviation". In: *Progress in aerospace sciences* 40.4-5 (2004), pp. 199–235.
- [40] Vansh Malik et al. "Comparative study and analysis between Solid Oxide Fuel Cells (SOFC) and Proton Exchange Membrane (PEM) fuel cell—A review". In: *Materials Today: Proceedings* 47 (2021), pp. 2270–2275.
- [41] Stefano Campanari et al. "Performance assessment of turbocharged PEM fuel cell systems for civil aircraft onboard power production". In: (2008).
- [42] Hao Wu. "Mathematical modeling of transient transport phenomena in PEM fuel cells". In: (2009).

- [43] Felix N Büchi and Supramaniam Srinivasan. "Operating proton exchange membrane fuel cells without external humidification of the reactant gases: Fundamental aspects". In: *Journal of the Electrochemical Society* 144.8 (1997), p. 2767.
- [44] Xingwang Tang, Yujia Zhang, and Sichuan Xu. "Experimental study of PEM fuel cell temperature characteristic and corresponding automated optimal temperature calibration model". In: *Energy* 283 (2023), p. 128456.
- [45] Lin Wang et al. "A parametric study of PEM fuel cell performances". In: *International journal of hydrogen energy* 28.11 (2003), pp. 1263–1272.
- [46] Alhussein Albarbar and Mohamad Alrweq. *Proton exchange membrane fuel cells: Design, modelling and performance assessment techniques*. Springer, 2017.
- [47] JE Dawes et al. "Three-dimensional CFD modelling of PEM fuel cells: an investigation into the effects of water flooding". In: *Chemical Engineering Science* 64.12 (2009), pp. 2781–2794.
- [48] Fengge Gao, Benjamin Blunier, and Abdellatif Miraoui. *Proton exchange membrane fuel cells modeling*. John Wiley & Sons, 2013.
- [49] Frano Barbir. *PEM fuel cells: theory and practice*. Academic press, 2012.
- [50] Thomas E Springer, TA Zawodzinski, and Shimshon Gottesfeld. "Polymer electrolyte fuel cell model". In: *Journal of the electrochemical society* 138.8 (1991), p. 2334.
- [51] Thomas E Springer, Mahlon S Wilson, and Shimshon Gottesfeld. "Modeling and experimental diagnostics in polymer electrolyte fuel cells". In: *Journal of the Electrochemical Society* 140.12 (1993), p. 3513.
- [52] Dawn M Bernardi and Mark W Verbrugge. "Mathematical model of a gas diffusion electrode bonded to a polymer electrolyte". In: *AIChE journal* 37.8 (1991), pp. 1151–1163.
- [53] Dawn M Bernardi and Mark W Verbrugge. "A mathematical model of the solid-polymer-electrolyte fuel cell". In: *Journal of the Electrochemical Society* 139.9 (1992), p. 2477.
- [54] John C Amphlett et al. "Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell: I. Mechanistic model development". In: *Journal of the Electrochemical Society* 142.1 (1995), p. 1.
- [55] John C Amphlett et al. "A model predicting transient responses of proton exchange membrane fuel cells". In: *Journal of Power sources* 61.1-2 (1996), pp. 183–188.
- [56] Ronald F Mann et al. "Development and application of a generalised steady-state electrochemical model for a PEM fuel cell". In: *Journal of power sources* 86.1-2 (2000), pp. 173–180.
- [57] Cesar A Luongo et al. "Next generation more-electric aircraft: A potential application for HTS superconductors". In: *IEEE Transactions on applied superconductivity* 19.3 (2009), pp. 1055–1068.
- [58] David Blanding. "Subsystem design and integration for the more electric aircraft". In: *5th international energy conversion engineering conference and exhibit (IECEC)*. 2007, p. 4828.
- [59] Ehab Sayed et al. "Review of electric machines in more-/hybrid-/turbo-electric aircraft". In: *IEEE Transactions on Transportation Electrification* 7.4 (2021), pp. 2976–3005.
- [60] Junbo Hou et al. "Control logics and strategies for air supply in PEM fuel cell engines". In: *Applied energy* 269 (2020), p. 115059.
- [61] Im Mo Kong et al. "Experimental study on the self-humidification effect in proton exchange membrane fuel cells containing double gas diffusion backing layer". In: *Applied Energy* 145 (2015), pp. 345–353.
- [62] Ramesh K Shah and Dusan P Sekulic. *Fundamentals of heat exchanger design*. John Wiley & Sons, 2003.
- [63] Yicheng Huang et al. "Thermal management of polymer electrolyte membrane fuel cells: A critical review of heat transfer mechanisms, cooling approaches, and advanced cooling techniques analysis". In: *Energy Conversion and Management* 254 (2022), p. 115221.
- [64] Guangsheng Zhang and Satish G Kandlikar. "A critical review of cooling techniques in proton exchange membrane fuel cell stacks". In: *international journal of hydrogen energy* 37.3 (2012), pp. 2412–2429.
- [65] Saman Rashidi et al. "Progress and challenges on the thermal management of electrochemical energy conversion and storage technologies: Fuel cells, electrolyzers, and supercapacitors". In: *Progress in Energy and Combustion Science* 88 (2022), p. 100966.

- [66] Ebrahim Afshari, Nabi Jahantigh, and Seyed Ali Atyabi. "Configuration of proton exchange membrane fuel cell gas and cooling flow fields". In: *PEM Fuel Cells*. Elsevier, 2022, pp. 429–463.
- [67] Syed Mushahid Hussain Hashmi. "Cooling strategies for PEM FC stacks". PhD thesis. Universitätsbibliothek der HSU/UniBwH, 2010.
- [68] Rupinder Singh, Amandeep Singh Oberoi, and Talwinder Singh. "Factors influencing the performance of PEM fuel cells: A review on performance parameters, water management, and cooling techniques". In: *International Journal of Energy Research* 46.4 (2022), pp. 3810–3842.
- [69] Ashley Fly and RH Thring. "Temperature regulation in an evaporatively cooled proton exchange membrane fuel cell stack". In: *international journal of hydrogen energy* 40.35 (2015), pp. 11976–11982.
- [70] Huy Quoc Nguyen and Bahman Shabani. "Proton exchange membrane fuel cells heat recovery opportunities for combined heating/cooling and power applications". In: *Energy Conversion and Management* 204 (2020), p. 112328.
- [71] Qin Chen et al. "Thermal management of polymer electrolyte membrane fuel cells: A review of cooling methods, material properties, and durability". In: *Applied Energy* 286 (2021), p. 116496.
- [72] Jiamin Xu et al. "Progress and perspectives of integrated thermal management systems in PEM fuel cell vehicles: A review". In: *Renewable and Sustainable Energy Reviews* 155 (2022), p. 111908.
- [73] Yafei Chang et al. "Humidification strategy for polymer electrolyte membrane fuel cells—A review". In: *Applied energy* 230 (2018), pp. 643–662.
- [74] Eiji Endoh et al. "Degradation study of MEA for PEMFCs under low humidity conditions". In: *Electrochemical and Solid-State Letters* 7.7 (2004), A209.
- [75] Mengbo Ji and Zidong Wei. "A review of water management in polymer electrolyte membrane fuel cells". In: *Energies* 2.4 (2009), pp. 1057–1106.
- [76] Ryan Huizing et al. "Design methodology for membrane-based plate-and-frame fuel cell humidifiers". In: *Journal of Power Sources* 180.1 (2008), pp. 265–275.
- [77] Colleen Spiegel. *PEM fuel cell modeling and simulation using MATLAB*. Elsevier, 2011.
- [78] G.L.M. Vonhoff. "Conceptual Design of Hydrogen Fuel Cell Aircraft. Flying on hydrogen for a more sustainable future". MA thesis. Technical University of Delft, 2021.
- [79] Wilhelmus Petrus Jozef Visser. "Generic Analysis Methods for Gas Turbine Engine Performance: The Development of the Gas Turbine Simulation Program GSP". Proefschrift ter verkrijging van de graad van doctor aan de Technische Universiteit Delft. PhD Thesis. Delft, The Netherlands: Technische Universiteit Delft, Jan. 6, 2015. ISBN: 978-94-6259-492-0. URL: <https://doi.org/10.4233/uuid:f95da308-e7ef-47de-abf2-aedbfa30cf63>.
- [80] A Saadi et al. "Comparison of proton exchange membrane fuel cell static models". In: *Renewable Energy* 56 (2013), pp. 64–71.
- [81] Sepand Haghighi et al. "OPEM : Open Source PEM Cell Simulation Tool". In: *Journal of Open Source Software* 3.27 (July 2018), p. 676. DOI: [10.21105/joss.00676](https://doi.org/10.21105/joss.00676). URL: <https://doi.org/10.21105/joss.00676>.
- [82] SM Sharifi Asl, S Rowshanzamir, and MH Eikani. "Modelling and simulation of the steady-state and dynamic behaviour of a PEM fuel cell". In: *Energy* 35.4 (2010), pp. 1633–1646.
- [83] Jeffryes W Chapman, Sydney L Schnulo, and Michael P Nitzsche. "Development of a Thermal Management System for Electrified Aircraft". In: *AIAA SciTech Forum*. E-19786. 2020.
- [84] North Atlantic Treaty Organisation NATO RTO. *Performance Prediction and Simulation of Gas Turbine Engine Operation, RTO technical report 44*. 2002.
- [85] NLR & TU Delft. *GSP Turbohaft Engine Model*. Brightspace TU Delft page.
- [86] Hugo Ledoux and Christopher Gold. "Spatial Interpolation: From Two to Three Dimensions". In: *Proceedings 11th International Symposium on Spatial Data Handling*. Leicester, UK: GIS Research Centre, School of Computing, University of Glamorgan, 2004, pp. 97–108.
- [87] P Zhou, CW Wu, and GJ Ma. "Influence of clamping force on the performance of PEMFCs". In: *Journal of Power Sources* 163.2 (2007), pp. 874–881. DOI: [doi.org/10.1016/j.jpowsour.2006.09.068](https://doi.org/10.1016/j.jpowsour.2006.09.068).



- [88] Daniel Juschus. "Preliminary Propulsion System Sizing Methods for PEM Fuel Cell Aircraft". MA thesis. Technical University of Delft, 2021.
- [89] Thomas Kadyk et al. "Analysis and design of fuel cell systems for aviation". In: *Energies* 11.2 (2018), p. 375. DOI: [doi.org/10.3390/en11020375](https://doi.org/10.3390/en11020375).
- [90] Kivanc Karacan et al. "Investigation of formability of metallic bipolar plates via stamping for light-weight PEM fuel cells". In: *International Journal of Hydrogen Energy* 45.60 (2020), pp. 35149–35161.
- [91] S Asghari, MH Shahsamandi, and MR Ashraf Khorasani. "Design and manufacturing of end plates of a 5 kW PEM fuel cell". In: *International journal of hydrogen energy* 35.17 (2010), pp. 9291–9297.
- [92] Shahram Karimi et al. "A review of metallic bipolar plates for proton exchange membrane fuel cells: materials and fabrication methods". In: *Advances in Materials Science and Engineering 2012* (2012).
- [93] Mostafa Habibnia, Mohammadreza Shirkhani, and Peyman Ghasemi Tamami. "Optimization of proton exchange membrane fuel cell's end plates". In: *SN Applied Sciences* 2 (2020), pp. 1–8.
- [94] P Muthukumar et al. "Review on large-scale hydrogen storage systems for better sustainability". In: *International Journal of Hydrogen Energy* (2023).
- [95] Chrysovalantou Ziogou et al. "Modeling, simulation and experimental validation of a PEM fuel cell system". In: *Computers & Chemical Engineering* 35.9 (2011), pp. 1886–1900.
- [96] . *The Engineering ToolBox*. URL: [https://www.engineeringtoolbox.com/water-vapor-saturation-pressure-d\\_599.html](https://www.engineeringtoolbox.com/water-vapor-saturation-pressure-d_599.html) (visited on 05/14/2023).

# A

## SATURATION VAPOUR PRESSURE VS TEMPERATURE TABLE

Temperature [K]	Temperature °C	Water saturation pressure kPa
273.16	0.01	0.61165
275.15	2	0.70599
277.15	4	0.81355
283.15	10	1.2282
287.15	14	1.599
291.15	18	2.0647
293.15	20	2.3393
298.15	25	3.1699
303.15	30	4.247
307.15	34	5.3251
313.15	40	7.3849
317.15	44	9.1124
323.15	50	12.352
327.15	54	15.022
333.15	60	19.946
343.15	70	31.201
353.15	80	47.414
363.15	90	70.182
369.15	96	87.771
373.15	100	101.42
383.15	110	143.38
393.15	120	198.67
403.15	130	270.28
413.15	140	361.54
423.15	150	476.16
433.15	160	618.23
453.15	180	1002.8
473.15	200	1554.9
493.15	220	2319.6
513.15	240	3346.9
533.15	260	4692.3
553.15	280	6416.6
573.15	300	8587.9
593.15	320	11284
613.15	340	14601
633.15	360	18666
643.15	370	21044

**Table A.1:** Saturated vapour pressure of water at selected temperatures [96]



# B

## EVTOL DATA

	Time [s]	Distance [m]	Altitude [m]	Power [kW]	Energy [kWh]	v_x [m/s]	v_z [m/s]
<b>Idle</b>	300	0	0.1	0	0	0	0.000333333
<b>Take Off</b>	50	0	50	1114.324938	15.47673525	0	0.998
<b>Descend</b>	500	20000	3000	542.196876	75.30512167	40	5.9
<b>Cruise</b>	3600	180000	3000	206.5937151	206.5937151	50	0
<b>Approach</b>	1200	50000	50	56.12451584	18.70817195	41.66666667	-2.458333333
<b>Take Down</b>	50	0	0.1	1111.099415	15.43193632	0	-0.998
<b>Idle</b>	300	0	0.1	0	0	0	0
<b>Take Off R</b>	60	0	50	1110.555441	18.50925736	0	0.831666667
<b>Cruise R</b>	120	5000	50	172.3899564	5.74633188	41.66666667	0
<b>Take Down R</b>	60	0	0.1	1107.329714	18.45549524	0	-0.831666667
<b>Idle</b>	300	0	0.1	0	0	0	0

**Table B.1:** eVTOL aircraft mission data used for the case study

# C

## CODE FILES PROJECT

### C.1. FLIGHT DATA READER

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from ambiance import Atmosphere
5
6 def read_data(file_name):
7     df_flight = pd.read_csv(file_name)
8
9     flight_condition = list(df_flight[df_flight.columns[0]])
10    Zh = list(df_flight['Altitude [m]'])
11    Pw_r = list(df_flight['Power [kW]'])
12    Time = list(df_flight['Time [s]'])
13    timeF = list(np.cumsum(Time))
14    v_x = list(df_flight['v_x [m/s]'])
15    v_z = list(df_flight['v_z [m/s]'])
16
17    TimeP = np.linspace(0, timeF[-1], int(timeF[-1]))
18
19    # Initialize the new power list
20    Pw_P = []
21
22    # Initialize the index for the timeF and Pw_r lists
23    index = 0
24
25    # For each time in TimeP
26    for time in TimeP:
27        # If the time is greater than or equal to the current time in timeF
28        if index < len(timeF) - 1 and time >= timeF[index]:
29            # Move to the next time and power
30            index += 1
31        # Add the current power to the new power list
32        Pw_P.append(Pw_r[index])
33
34    # Convert the new power list to a numpy array
35    Pw_P = np.array(Pw_P)
36
37    return timeF, Zh, Pw_r, TimeP, Pw_P, flight_condition, v_x, v_z, df_flight, Time
38
39 def plot_data(timeF, Zh, Pw_r, TimeP, Pw_P):
40    fig, ax1 = plt.subplots()
41
42    ax1.plot(timeF, Zh, 'b-')
43    ax1.set_xlabel('Time [s]')
44    ax1.set_ylabel('Altitude [m]', color='b')
45    ax1.tick_params('y', colors='b')
46
47    ax2 = ax1.twinx()
48    ax2.plot(TimeP, Pw_P, 'r-')
49    ax2.set_ylabel('Power [kW]', color='r')
50    ax2.tick_params('y', colors='r')
51
52    plt.title('Time vs Altitude and Power')
53    plt.show()
54
```

```

55
56 def get_flying_speed(flight_phase, df_flight):
57     # Filter the dataframe based on the flight phase
58     df_filtered = df_flight[df_flight[df_flight.columns[0]] == flight_phase]
59
60     # Get the flying speed v_x
61     v_x = df_filtered['v_x [m/s]'].values
62
63     # If v_x is not empty, return the first value as an integer
64     if v_x.size > 0:
65         return float(v_x[0])
66     # If v_x is empty, return None
67     else:
68         return None
69
70 def get_altitude(flight_phase, df_flight):
71     # Filter the dataframe based on the flight phase
72     df_filtered = df_flight[df_flight[df_flight.columns[0]] == flight_phase]
73
74
75     Zh = df_filtered['Altitude [m]'].values
76
77     if Zh.size > 0:
78         return float(Zh[0])
79     else:
80         return None
81
82 def get_power(flight_phase, df_flight):
83     # Filter the dataframe based on the flight phase
84     df_filtered = df_flight[df_flight[df_flight.columns[0]] == flight_phase]
85
86
87     Pw = df_filtered['Power [kW]'].values
88
89     if Pw.size > 0:
90         return float(Pw[0])
91     else:
92         return None
93
94 def get_flying_Mach(Zh, vx):
95
96     atmosphere = Atmosphere(Zh)
97     # Calculate the speed of sound
98     a0 = atmosphere.speed_of_sound[0]
99
100    # Calculate the Mach number
101    Mach = vx / a0
102
103    return Mach
104
105 def calculate_air_properties(altitude, Mach):
106     atmosphere = Atmosphere(altitude)
107     kappa_air = 1.4 # ratio of specific heats for air
108     # Calculate the speed of sound
109     a0 = atmosphere.speed_of_sound[0]
110
111     # Calculate the static temperature
112     T0 = atmosphere.temperature[0]
113
114     # Calculate the total temperature
115     Tt0 = T0*(1 + ((kappa_air-1)*0.5*Mach**2))
116
117     # Calculate the static pressure
118     p0 = atmosphere.pressure[0]
119
120     # Calculate the total pressure
121
122     Pt0 = p0 * (1 + Mach**2*0.5*(kappa_air-1))**(kappa_air/(kappa_air-1))
123
124     # Calculate the density
125     rho0 = atmosphere.density[0]

```

```

126
127 # Calculate the dynamic pressure
128 q0 = 0.5 * rho0 * (Mach * a0)**2
129
130 return T0, Tt0, p0, Pt0, rho0, q0

```

Listing C.1: Script for functions used to read the flight data

## C.2. MODEL FUNCTIONS

```

1 from math import *
2 import matplotlib.pyplot as plt
3 import sys
4 # from jinja2 import Environment, FileSystemLoader
5 from scipy.interpolate import griddata, InterpolatedUnivariateSpline
6 import numpy as np
7 import pandas as pd
8 import subprocess
9 import csv
10 import os
11 import numpy as np
12 import csv
13
14 def fc(T, ph2, po2, j, gamma):
15     n = 2 # [-]
16     F = 96485 # [C / mol]
17     deltas = -44.34 # [J / mol. k]
18     e0 = 1.223 # [ V ] This is the maximum thermodynamic voltage derived from the gibbs free energy
19     deltaG/2F = 237000/2F
20     T0 = 298.15 # [K]
21     p0 = 101325 # [Pa]
22     R = 8.314 # J / K mol
23     j_leak = 1e-2 # A / cm2
24     alphah2 = 1 # [-] #From HY4 parameter optimisation
25     alphao2 = 0.3 # [-] #From HY4 parameter optimisation
26     c = 0.02 # [V] #Obtained from O'Hayre
27     j_0o2 = 2.45e-8 * ((po2/p0)**0.54) * exp((67e3/R)*(1/T0 - 1/T))
28     # j_0o2 = 8e-4 #A/cm2
29     j_0h2 = 0.27 * exp((16e3 / R) * (1 / T0 - 1 / T))
30     # j_0h2 = 40 #A/cm2
31     asr = 7.5e-2 # ohm/ cm2
32     # j_1 = 2 # A / cm2
33     j_1 = 2
34     "Thermodynamic Voltage"
35     E = e0 + ((R*T)/(n*F)) * log(ph2*sqrt(po2)) + (deltas/(n*F)) * (T - T0) # [V]
36     "Losses"
37     etaact = ((R * T) / (4 * alphao2 * F)) * log((j + j_leak) / j_0o2) + (R * T) / (n * alphah2 * F) * log
38     ((j + j_leak) / j_0h2) # Activation loss at both the anode and the cathode
39     "etaact from amphlett"
40     zi1 = -0.948
41     ch2 = (ph2)/(1.09e6*exp(77/T))
42     co2 = (po2)/(5.08e6*exp(-498/T))
43     zi2 = 0.00286 + 0.0002*log(1) + 4.3e-5*log(ch2)
44     zi3 = 7.6e-5
45     zi4 = -1.93e-4
46     etaactA = -(zi1 + zi2*T + zi3*T*log(co2) + zi4*T*log(j))
47     "etaohm from amphlett"
48     # gamma = 23 # This is to measure humidity in the fuel cell
49     l = 0.02
50     A = 1
51     rhom = (181.6*(1 + 0.03*j)+0.062*((T/303)**2) * j**2.5) / ((gamma - 0.634 - 3*j) * exp(4.18*((T-303)/
52     T)))
53     rproton = rhom * l / A
54     etaohmA = j * rproton
55     "etaohm from Ohayrae"
56     etaohm = j * asr # Ohmic losses
57     # (1 + (1 / alphah2))
58     etaconc = ((R * T) / (n * F)) * log(j_1 / (j_1 - (j + j_leak)))
59     etaconcA = c * log(j_1 / (j_1 - (j + j_leak)))
60     eta = etaact + etaohmA + etaconc
61     V = E - eta

```

```

59     P = V*j
60     return V, P
61
62 def plot(x, y, y1, label1, label2, xlabel1, ylabel1, ylabel2, color1, color2, ax1=None, ax2=None, fig=None):
63     if not ax1:
64         fig, ax1 = plt.subplots()
65         ax2 = ax1.twinx()
66     ax1.plot(x,y,color1,label = label1)
67     ax2.plot(x,y1,color2,label = label2)
68
69
70     ax1.set_xlabel(xlabel=xlabel1)
71     ax1.set_ylabel(ylabel=ylabel1)
72     ax2.set_ylabel(ylabel=ylabel2)
73     # ax1.set_xlim([0,2])
74     # ax1.set_ylim([0.0, 1.2])
75     # ax2.set_ylim([0, 1.5])
76
77     # ax1.grid()
78     # plt.grid(True)
79     # plt.grid(axis = 'x')
80     fig.legend()
81     #plt.show()
82     return ax1,ax2,fig,plt
83
84 "Calculate relative humidity, ref using larmin fuel cell system explained book"
85 "T is the fuel cell operating temperature, pexit is the exit pressure of the fuel cell, stoichiometric_air
86   is the stoichiometric air ratio"
87 "dry is a boolean value to determine if the inlet air is dry or humid, pinlet is the inlet pressure of the
88   fuel cell, pwinlet is the inlet pressure of the water, a is the relative humidity phi"
89
90 def interpolatepsat(T):
91     df = pd.read_csv('model\psat.csv')
92     Tsat = np.array(df['Temperature [K]'])
93     psat = np.array(df['Water saturation pressure kPa'])
94     # Create a spline of the data
95     spline = InterpolatedUnivariateSpline(Tsat, psat)
96
97     # Use the spline to interpolate/extrapolate
98     Psat = spline(T)
99
100    return Psat
101
102 def calc_pw(phi, T):
103     # df = pd.read_csv('model\psat.csv')
104     # Tsat = list(df['Temperature [K]'])
105     # psat = list(df['Water saturation pressure kPa'])
106     # for temp in Tsat:
107     #     if temp == T:
108     #         Psat = psat[Tsat.index(temp)]
109     #         break
110     #     elif abs(T - temp) <= 5:
111     #         if T > temp:
112     #             Psat = psat[Tsat.index(temp)]
113     #             break
114     #         elif T < temp:
115     #             Psat = psat[Tsat.index(temp)]
116     #     else:
117     #         Psat = 361.54
118     Psat = interpolatepsat(T)
119     Pw = phi * (Psat*10**3)
120     return Pw
121
122 def calc_lamda(T, pexit, stoichiometric_air, dry, pinlet,
123               pwinlet=None):
124     # lamda = None
125     df = pd.read_csv('model\psat.csv')
126     Tsat = list(df['Temperature [K]'])
127     psat = list(df['Water saturation pressure kPa'])
128     for temp in Tsat:
129         if temp == T:

```

```

128     Psat = psat[Tsat.index(temp)]
129     break
130     elif abs(T - temp) <= 5:
131         if T > temp:
132             Psat = psat[Tsat.index(temp)]
133             break
134         elif T < temp:
135             Psat = psat[Tsat.index(temp)]
136     if dry is True:
137         Pwout = (0.420 * pexit) / (stoichiometric_air + 0.210)
138     else:
139         Phi = pwinlet / (pinlet - pwinlet)
140         Pwout = ((0.420 + Phi * stoichiometric_air) * pexit) / ((1 + Phi)*stoichiometric_air + 0.210) #
141         This is the exit pressure of water
142         a = Pwout / (Psat*10**3) # Relative humidity phi
143         # https://doi.org/10.1016/j.energy.2009.12.010
144         # Calculation of the water content fo teh memebarane lambda
145         if 0 < a <= 1:
146             lamda = 0.0043 + 17.81 * a - 39.85 * a**2 + 36 * a**3
147         elif 1 < a < 3:
148             lamda = 14 + 1.4 * (a - 1)
149         elif a == 3:
150             lamda = 16.8
151         elif a > 3:
152             lamda = 22
153     return lamda, a, Pwout
154
155 def amph(A, jmax, Ncells, ph2, po2, ropt, T, istart, istep, iend, l, lamda, filename):
156     lst = [6, 'p', filename, A, jmax, Ncells, ph2, po2, ropt, T, istart, istep, iend, l, lamda] # Input
157     list for OEM
158     lst2 = []
159     for x in lst:
160         lst2.append(str(x) + "\n")
161     lst = lst2
162     with open('orderN.txt', 'w') as f:
163         f.writelines(lst)
164
165     return subprocess.run('OPEM-1.3.exe < orderN.txt', shell=True)
166
167 def mdots(power, voltage, lamda=None):
168     if lamda is None:
169         lamda = 2 #Typical stoichiometric ratio [Enough air to supply oxygen to the FC]
170     F = 96485 # Faraday constant [C/mol]
171     o2MolarMass = 31.9988e-3 # [kg/mol]
172     h2MolarMass = 2.01588e-3 # [kg/mol]
173     h2oMolarMass = 18.01528e-3 # [kg/mol]
174     airmolarMass = 28.9647e-3 # [kg/mol]
175     mdot_air_in = (airmolarMass / (0.21*4*F)) * (power/voltage) * lamda #kg/s
176     o2_used = (o2MolarMass / (4*F)) * (power/voltage) #kg/s
177     mdot_air_out = mdot_air_in - o2_used #kg/s
178     h2o_produced = (h2oMolarMass / (2 * F)) * (power/voltage) #kg/s
179     h2_used = (h2MolarMass / (2 * F)) * (power/voltage) #kg/s
180     return mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used
181
182 def plot_data(x, y, label, color, dual_axis=False, x2=None, y2=None, label2=None, color2=None):
183     """
184     Plot data with an option for dual axes.
185
186     Parameters:
187     - x: x values for the first plot
188     - y: y values for the first plot
189     - label: label for the first plot
190     - dual_axis: True if you want to plot on two axes, False otherwise
191     - x2: x values for the second plot (required if dual_axis is True)
192     - y2: y values for the second plot (required if dual_axis is True)
193     - label2: label for the second plot (required if dual_axis is True)
194     """
195     plt.figure(figsize=(8, 6))
196

```

```

197 # Plotting the first set of data
198 plt.plot(x, y, color ,label=label)
199
200 if dual_axis:
201     if x2 is None or y2 is None or label2 is None:
202         raise ValueError("For dual axis, x2, y2, and label2 must be provided.")
203
204     # Creating a second y-axis
205     ax2 = plt.gca().twinx()
206
207     # Plotting the second set of data on the second axis
208     ax2.plot(x2, y2, color2, label=label2)
209
210 plt.xlabel("X-axis")
211 plt.ylabel("Y-axis")
212
213 # Display the grid
214
215 plt.grid()
216
217 # Display legend
218 plt.legend()
219
220 # Display the plot
221 plt.show()
222 return plt
223
224 def HEX_Power(T_op, T_static, V_cell, P_needed):
225     st = 1 # [-] # This is the anode stoichiometric ratio, and it is 100% fuel utilization
226     # I had 0.95 multiplied by V_cell for some reason but I took it out because it does not make sense to
227     # me
228     E0 = 241.83e3 / (2 * 96485) # J/mol
229     q_dot_heat = ((E0*st/(V_cell)) - 1) * P_needed # From the fuel cell system explained by larmine book
230     # q_dot_heat = ((1.48*st/(V_cell)) - 1) * P_needed # The equation is from the fuel cell fundamentals
231     # book considering water as liquid Conservative
232     f_dt = 0.0038 * (T_static/(T_op - T_static))**2 + 0.0352 * (T_static / (T_op - T_static)) + 0.1817
233     P_cs = (0.371 * q_dot_heat + 1.33) * f_dt
234     return P_cs , q_dot_heat #I am switching these two values around for now !!! [I brought it back to
235     normal]
236
237
238
239 def compressor_power(beta, mdot, T1, eta_m, eta_c): # realistically eta_m is 0.9 and eta_c is 0.8
240     gamma = 1.4 # [-]
241     cp = 1004 # J/kgK
242     Power = mdot * cp * (T1/(eta_m * eta_c)) * (((beta)**((gamma -1)/gamma)) - 1)
243     return Power
244
245
246
247 def Temp_aft_compress(Compressor_Pi, T0_upcoming, eta_compressor, kappa):
248     T0_aft = T0_upcoming*(1 + (1/eta_compressor)*(Compressor_Pi**((kappa-1)/kappa)-1))
249     return T0_aft
250
251
252
253 def delete_file(file_path):
254     if os.path.exists(file_path):
255         os.remove(file_path)
256         print(f"{file_path} has been deleted.")
257     else:
258         print(f"The file {file_path} does not exist.")
259
260
261
262 def plot_energy_balance(volt, power, P_compressor, P_cooling, effective_power, Wasted_power):
263     # Calculate total available power
264     E0 = 241.83e3 / (2 * 96485) # J/mol
265     total_power = (E0 / volt) * power
266
267     # Create a bar for the total power
268     plt.bar('Available total power', total_power, color='k', width=0.15)
269
270     # Write the total power value on the bar

```

```

264 plt.text('Available total power', total_power / 2, f'{total_power:.2f}', ha='center', va='center',
265         color='white')
266
267 # Adjust the order of power values
268 power_values = [effective_power, P_compressor, P_cooling, Wasted_power]
269
270 # Adjust the order of labels
271 labels = ['Effective Power', 'Compressor Power', 'Cooling System Power', 'Wasted Power']
272
273 # Adjust the order of colors
274 colors = ['g', 'y', 'b', 'r']
275
276 # Create the bars for all powers
277 for i in range(4): # Changed to 4 because the total power is already plotted
278     plt.bar('Energy Balance', power_values[i], bottom=sum(power_values[:i]), color=colors[i], width
279           =0.15)
280
281     # Write the power value on the bar
282     plt.text('Energy Balance', sum(power_values[:i]) + power_values[i] / 2, f'{power_values[i]:.2f}',
283           ha='center', va='center', color='white')
284
285 # Add a title and labels
286 plt.title('Energy Balance')
287 plt.xlabel('Component')
288 plt.ylabel('Power (W)')
289
290 # Add a legend
291 plt.legend(['Total Power'] + labels)
292
293 # Display the chart
294 plt.show()
295
296 ## Thesis by daniel is used here to estimate the fuel cell mass
297 def fc_mass(n_stacks_series, n_cells, area_cell):
298     """
299     Calculate mass of stack(s) in the FC system.
300
301     :param n_stacks_series: Number of stacks in series in FC system
302     :param volt_req: Voltage to be delivered by FC system in V
303     :param volt_cell: Nominal cell voltage in V
304     :param power_req: Electrical power to be delivered by stacks (bigger than propulsive output power of
305     FC system)
306     :param power_dens_cell: Nominal cell power density in W/m^2
307     :return: mass of stack(s) in kg
308     """
309     # # constants
310     # bipolar plate
311     t_bp_dim = 2e-4 # m - thickness for determination of complete dimensions
312     t_bp = 2e-4 # m
313     rho_bp = 8e3 # kg/m3 - SS304L
314
315     # endplate
316     t_ep = 2.5e-2 # m
317     rho_ep = 8e3 # kg/m3 - same as bp
318
319     # bolts - remove those?
320     n_bolt = 10 # see Dey 2019
321     rho_bolt = 8e3
322
323     # MEA
324     rho_mea = 0.2 # kg/m2 - Kadyk 2018
325
326     # # # calculations - per stack
327     # n_cells = volt_req / volt_cell / n_stacks_series # number of cells per stack
328     # area_cell = power_req / n_stacks_series / n_cells / power_dens_cell # area of single cell
329
330     m_bp = n_cells*t_bp*rho_bp*area_cell # mass of bipolar plates of one stack
331     m_ep = 2*t_ep*rho_ep*area_cell # mass of endplates of one stack
332     m_mea = rho_mea*area_cell*n_cells # mass of MEA of one stack
333
334     d_bolt = sqrt(area_cell/600) # diameter of bolts - see Dey 2019

```



```

331 l_bolt = 2*t_ep + n_cells*t_bp # length of bolts
332 m_bolts = n_bolt*pi*(d_bolt/2)**2*l_bolt*rho_bolt # mass of bolts of one stack
333
334 m_tot = m_bp+m_ep+m_mea # mass of a single stack
335 # print("BP: {}, EP: {}, Bolts: {}, MEA: {}".format(m_bp, m_ep, m_bolts, m_mea ))
336
337 dim = [np.sqrt(area_cell),np.sqrt(area_cell),l_bolt]
338 # , dim, [n_cells, area_cell]
339 return m_tot*n_stacks_series
340
341
342
343 def write_lists_to_csv(filename, **kwargs):
344 # Open (or create) a CSV file and write lists to it
345 with open(filename, 'w', newline='') as file:
346     writer = csv.writer(file)
347
348     # Write the headers
349     headers = kwargs.keys()
350     writer.writerow(headers)
351
352     # Write the data
353     for values in zip(*kwargs.values()):
354         writer.writerow(values)
355
356 def calc_payload(fcmass, TG_maxpower, MIOW, fuelused, compressor_power, cooling_power):
357     tg_pd = 0.00435e6 #W/kg
358     comp_pd = 11/11250 #kg/W
359     hex_pd = 11.1 / (1.84e3 + 3.72e3) #kg/W
360     tankIV_pd = 5.7 #kgh2/kg
361     # bop = 51.1 #kg
362     structure = 1905 #kg
363     payload = MIOW - fcmass - TG_maxpower/tg_pd - compressor_power*comp_pd - cooling_power*hex_pd -
364     structure - fuelused/tankIV_pd - fuelused
365     return payload
366
367 def calculate_water(reli_in, rel_out, T_in, T_out, p_in, stoichiometric_air, mdotair):
368 # in and out means inlet and outlet of the fuel cell
369 # outlet relative humidity is dependant on the fuel cell water content number
370 # inlet relative humidity is an input by the user
371 psat_in = interpolatepsat(T_in)
372 psat_out = interpolatepsat(T_out)
373 pwout = rel_out * psat_out
374 pwin = reli_in * psat_in
375 psi = (pwin / (p_in - pwin))
376
377 #Calculate exit pressure using eq 4.6 in fuel cell system explained book by larmine
378 p_out = (pwout*((1 + psi)*stoichiometric_air + 0.210)) / (0.420 + psi*stoichiometric_air )
379
380 # Calculate the amount of water required to be injected into the fuel cell by the humidifier using eq
381 4.3
382 mdotwr = 0.622 * (pwout / (p_in - pwout)) * mdotair
383
384 return p_out, mdotwr

```

Listing C.2: Script for all functions used in the model

### C.3. GSP DATA PROCESSING

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from mpl_toolkits.mplot3d import Axes3D
5 from scipy.interpolate import griddata
6
7
8 def read_GSP(filename, rows_to_skip):
9     df = pd.read_csv(filename, encoding='unicode_escape', header=[1], skiprows=rows_to_skip, decimal=',')
10    return df
11
12 def get_data(df):

```

```

13 Altitude = df[ 'Zp\r\n[m]' ].astype(float)
14 Mach = df[ 'Macha\r\n[-]' ].astype(float)
15 Tt4 = df[ 'Tt4\r\n[K]' ].astype(float)
16 Tt3 = df[ 'Tt3\r\n[K]' ].astype(float)
17 PWshaft = df[ 'PWshaft\r\n[kW]' ].astype(float)
18 WF = df[ 'WF\r\n[kg/s]' ].astype(float)
19 W = df[ 'W\r\n[kg/s]' ].astype(float)
20 return Altitude, Mach, Tt4, PWshaft, WF, W, Tt3
21
22
23 rows_to_skip = list(range(2, 989, 17)) + [989, 990, 991, 992] # NaN rows excluding the design point
24
25 df = read_GSP('GTG\DATA_GT3.CSV', rows_to_skip)
26
27 # print(df)
28
29 Altitude, Mach, Tt4, PWshaft, WF, W, Tt3 = get_data(df)
30
31
32 # # plotting for mach = 0.2 and 1250
33
34 # m1250 = []
35 # A1250 = []
36 # p1250 = []
37 # wf1250 = []
38
39 # m1 = []
40 # A1 = []
41 # p1 = []
42 # wf1 = []
43
44 # m2 = []
45 # A2 = []
46 # p2 = []
47 # wf2 = []
48
49 # m0 = []
50 # A0 = []
51 # p0 = []
52 # wf0 = []
53
54 # for i in range(0, len(Altitude),1):
55 #     if Altitude[i] == 0 and Mach[i] == 0.0:
56 #         m0.append(Mach[i])
57 #         A0.append(Altitude[i])
58 #         p0.append(PWshaft[i])
59 #         wf0.append(WF[i])
60 #     if Altitude[i] == 0 and Mach[i] == 0.1:
61 #         m1.append(Mach[i])
62 #         A1.append(Altitude[i])
63 #         p1.append(PWshaft[i])
64 #         wf1.append(WF[i])
65 #     if Altitude[i] == 0 and Mach[i] == 0.2:
66 #         m2.append(Mach[i])
67 #         A2.append(Altitude[i])
68 #         p2.append(PWshaft[i])
69 #         wf2.append(WF[i])
70 #     if Altitude[i] == 0 and Mach[i] == 0.3:
71 #         m1250.append(Mach[i])
72 #         A1250.append(Altitude[i])
73 #         p1250.append(PWshaft[i])
74 #         wf1250.append(WF[i])
75
76 # for i in range(320,336,1):
77 #     m1250.append(Mach[i])
78 #     A1250.append(Altitude[i])
79 #     p1250.append(PWshaft[i])
80 #     wf1250.append(WF[i])
81
82 # print(m1250)
83 # print(A1250)

```

```

84 # print(p1250)
85
86
87 # plt.plot(wf0, p0, label='Altitude 0 m and Mach 0.0')
88 # plt.plot(wf1, p1, label='Altitude 0 m and Mach 0.1')
89 # plt.plot(wf2, p2, label='Altitude 0 m and Mach 0.2')
90 # plt.plot(wf1250, p1250, label='Altitude 0 m and Mach 0.3')
91 # plt.xlabel('Fuel mass flow rate [kg/s]')
92 # plt.ylabel('Shaft Power [kW]')
93 # plt.grid()
94 # plt.legend()
95 # plt.show()
96
97
98
99 # fig = plt.figure()
100 # ax = fig.add_subplot(111, projection='3d')
101
102 # sc = ax.scatter(PWshaft, WF, Altitude, c=Mach, cmap='viridis')
103
104 # # Create a meshgrid for the plane
105 # z = np.array([min(Altitude), max(Altitude)])
106 # y = np.array([min(Tt4), max(Tt4)])
107 # z, y = np.meshgrid(z, y)
108
109 # # # Set z to 0 to create a plane that crosses the x-axis at 0
110 # # x = np.zeros_like(z)
111
112 # # Plot the plane
113 # # ax.plot_surface(x, y, z, color='grey', alpha=0.7, rstride=1, cstride=1)
114 # ax.set_xlabel('Shaft Power [kW]', fontsize=15)
115 # ax.set_ylabel('PWshaft [kW]')
116 # ax.set_ylabel('Fuel flow [kg/s]', fontsize=15)
117 # ax.set_zlabel('Altitude [m]', fontsize=15)
118
119 # # Add a colorbar
120 # fig.colorbar(sc, label='Mach number')
121
122 # plt.show()
123
124 # Prepare data for interpolation
125 points = df[['Mach\r\n[-]', 'Tt4\r\n[K]', 'Zp\r\n[m]']].values
126 valuesWf = df['WF\r\n[kg/s]'].values
127 valuesPWshaft = df['PWshaft\r\n[kW]'].values
128 valuesTt3 = df['Tt3\r\n[K]'].values
129
130
131
132
133
134
135 def interpolate_wf(Mach, Tt4, Altitude):
136     return griddata(points, valuesWf, (Mach, Tt4, Altitude), method='linear')
137
138 def interpolate_Tt3(Mach, Tt4, Altitude):
139     return griddata(points, valuesTt3, (Mach, Tt4, Altitude), method='linear')
140
141 def interpolate_PWshaft(Mach, Tt4, Altitude):
142     return griddata(points, valuesPWshaft, (Mach, Tt4, Altitude), method='linear')
143
144 def lookup_values(Mach_target, Altitude_target, PWshaft_target):
145     # Create a DataFrame from the data
146     data = pd.DataFrame({
147         'Mach': Mach,
148         'Altitude': Altitude,
149         'PWshaft': PWshaft,
150         'WF': WF,
151         'Tt4': Tt4
152     })
153
154     # Calculate the Euclidean distance between the target point and all points in the data

```

```

155 distances = np.sqrt((data['Mach'] - Mach_target)**2 + (data['Altitude'] - Altitude_target)**2 + (data[
156     'PWshaft'] - PWshaft_target)**2)
157
158 # Find the index of the minimum distance
159 index = distances.idxmin()
160
161 # Return the corresponding WF and Tt4
162 return data.loc[index, 'WF'], data.loc[index, 'Tt4'], data.loc[index, 'Mach'], data.loc[index, '
163     Altitude'], data.loc[index, 'PWshaft']
164
165 def calc_gt_efficiency(flight_mach, Tt4, altitude, Tt3, PW_GT):
166     fuelgtflow = interpolate_wf(flight_mach, Tt4, altitude)
167     gt_eff = PW_GT / (fuelgtflow * 120*10**6)
168     return gt_eff
169
170 # Use the lookup function
171 # # Mach_target = 0.32 # Replace with your target Mach number
172 # # Altitude_target = 2000 # Replace with your target altitude
173 # # PWshaft_target = 500 # Replace with your target shaft power value
174 # # WF, Tt4, Mach, Altitude, PWshaft = lookup_values(Mach_target, Altitude_target, PWshaft_target)
175 # # print('Fuel Flow:', WF, 'Tt4:', Tt4, 'Mach:', Mach, 'Altitude:', Altitude, 'PWshaft:', PWshaft)
176
177 # # Interpolate the data
178 # PW = interpolate_PWshaft(0.35, 1200, 2000)
179 # Wf = interpolate_wf(0.4, 1200, 1000)
180 # WF, Tt4, Mach, Altitude, PWshaft = lookup_values(0.35, 2000, PW)
181 # print('Fuel Flow:', WF, 'Tt4:', Tt4, 'Mach:', Mach, 'Altitude:', Altitude, 'PWshaft:', PWshaft)
182 # print(PW, Wf)

```

Listing C.3: Script used to process GSP data

## C.4. PROPULSION SYSTEM INTEGRATION

```

1 import numpy as np
2 from fc import *
3 import pandas as pd
4 from flightdata_reader import calculate_air_properties
5 import matplotlib.pyplot as plt
6
7 def on_design(Altitude, flight_mach, OCs, OCrs):
8     # Calculate the air properties at the given altitude
9
10    T0, Tt0, p0, pt0, _, _ = calculate_air_properties(Altitude, flight_mach)
11
12    # Air properties after compressor
13
14    Tt1 = Temp_aft_compress(OCrs['beta'], Tt0, 0.8, 1.4) # Total temperature after compressor
15
16    air_properties = {'T0': T0, 'Tt0': Tt0, 'p0': p0, 'pt0': pt0, 'Tt1': Tt1}
17
18    A = 1 # For amphlett model to have everything per unit area cm^2
19    n = 1 # Number of cells for the amphlett model to have all the calculated values per cell
20    jmax = 1 # Maximum current density in A/cm^2
21
22    ##### Humidity calculations
23
24    # pexit = 0.4 * OCs['Pt1']
25    # phi_inlet = 0.25
26    # pwinlet = calc_pw(phi_inlet, Tt1) # NOTE: I am not using an intercooler and therefore the
27    # temperature of the air is the same as the temperature after the compressor
28    # OCs['lambda'], phi_outlet, pwout = calc_lambda(OCs['T_oc'], pexit, OCs['st_air'], dry=False, pinlet=
29    # OCs['Pt1'], pwinlet=pwinlet)
30
31    amph(A, jmax, n, OCs['p_H2'], OCs['p_O2'], 0, OCs['T_oc'], 0, 0.001, 1, OCrs['1'], OCs['lambda'], 'FC
32    ')
33
34    df = pd.read_csv('Amphlett/FC.csv')

```

```

35 voltage = list(df['Vcell (V)'])
36 current = list(df['I (A)'])
37 power_FC = list(df['Power (W)'])
38
39 n_fac = 0.8 # Factor to calculate n% of the maximum power_FC
40 voltage_efficiency = OCrs['eta_v'] # Voltage efficiency of the fuel cell
41
42 power_nth_percent = n_fac * max(power_FC)
43 voltage_at_efficiency = voltage_efficiency * max(voltage)
44
45 # Find the value in Voltage_FC that is closest to voltageclose
46 closest_voltage = min(voltage, key=lambda x: abs(x - voltage_at_efficiency))
47
48 # Get the index of the closest voltage value
49 closest_voltage_index = voltage.index(closest_voltage)
50
51 closest_power = None
52 closest_power_index = None
53
54 # Find the index of the maximum power_FC
55 max_power_index = power_FC.index(max(power_FC))
56
57 # Iterate over the power values
58 for index, power in enumerate(power_FC[:max_power_index + 1]):
59     # If it's the first power value or it's closer to 80% of the maximum power
60     if closest_power is None or abs(power - power_nth_percent) < abs(closest_power - power_nth_percent
61 ):
62         closest_power = power
63         closest_power_index = index
64         # If the power is greater than or equal to 80% of the maximum power, stop the loop
65         if power >= power_nth_percent:
66             break
67
68 # Size the fuel cell based on the cell voltage and power density [Changed it to take the voltage index
69 # instead of the power index]
70 V_cell = voltage[closest_voltage_index] # cell coltage at nth% of max power
71 P_dens = power_FC[closest_voltage_index] # Power density at nth% of max power density
72
73 p_needed = OCrs['P_req'] # W [Power needed by the user for cruise from the excel sheet]
74
75 n_cells = OCrs['V_req'] / V_cell / OCrs['n_stacks']
76
77 # The new convergence loop impmented here
78
79 P_effective = 0
80 area = 0
81 while P_effective < OCrs['P_req']:
82     area += 0.5
83     P_FC = P_dens * area * n_cells * OCrs['n_stacks']
84     mdot_air_in, _, _, _ = mdots(P_FC, V_cell) #Calculate the new mdot_air_in
85     P_cs, _ = HEX_Power(OCs['T_oc'], T0, V_cell, P_FC) #initial compressor power in W
86     P_comp = compressor_power(OCrs['beta'], mdot_air_in, Tt0, 0.9, 0.8)
87     P_effective = P_FC - P_comp - P_cs
88
89 #Calculate new mass flow rates
90 mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used = mdots(P_FC, V_cell) #Calculate the new
91 #mdot_air_in
92 P_comp = compressor_power(OCrs['beta'], mdot_air_in, Tt0, 0.9, 0.8) #Calculate the new compressor
93 #power
94 P_cs, qheat = HEX_Power(OCs['T_oc'], T0, V_cell, P_FC) # Power required by the cooling system
95
96 P_effective = P_FC - P_comp - P_cs
97
98 # Water required to keep the humidity constant
99
100 # mwr = 0.622 * ( pwinlet / (pexit - pwinlet) ) * mdot_air_in
101
102 # humidity_output = [pwinlet, mwr, pwout, phi_outlet, OCs['lambda']]
103
104 # A = P_new/OCrs['n_stacks']/n_cells/P_dens

```

```

102 # Calculate the efficiencies related to the fuel cell
103 delta_g = (-61.12/2 - 38.96 ) + 306.69 # kJ/mol
104 delta_h = 285.83 # kJ/mol High heating value of water is used [HHV is liquid and LHV is gas because
105 you can still get energy from the condensation of water]
106 E0 = -237e3 / (2*96485) # V
107 eta_thermo = delta_g / delta_h # Thermo efficiencyThermodynamic efficiency and also the maximum
108 theoretical efficiency
109 eta_voltage = V_cell / abs(E0) # Voltage efficiency
110 eta_fuel = 1 # Fuel utilisation efficiency where 2 is the stoichiometric ratio supplying more fuel
111 than required in this case air ? set it to 1
112 eta_total = eta_thermo * eta_voltage * eta_fuel # Total efficiency
113
114 # delete_file('Amphlett/FC.csv')
115 # delete_file('Amphlett/FC.html')
116 # delete_file('Amphlett/FC.opem')
117
118 return area, n_cells, P_effective, P_FC, P_cs, P_comp, mdot_air_in, o2_used, mdot_air_out,
119 h2o_produced, h2_used, \
120 V_cell, P_dens, eta_total, eta_voltage, voltage, current, power_FC, closest_voltage_index,
121 eta_voltage, qheat, Tt1
122
123 def plot_polarisation(current, voltage, power, index, nfac):
124
125     nfacnew = round(float(nfac*100), 1)
126
127     fig, ax1 = plt.subplots()
128
129     ax1.set_ylim(bottom=0)
130
131     ax2 = ax1.twinx()
132
133     # ax2.set_ylim(bottom=0)
134     ax1.plot(current, voltage, 'b-', label='Polarization curve')
135     ax2.plot(current, power, 'r-', label='Power density curve')
136     ax2.plot(current[index], power[index], 'X', label= str(nfacnew) + '$\%$ of max Voltage', markersize
137 =10, color='g')
138     fig.legend(loc=(0.25, 0.15))
139
140 # Set smaller intervals for x and y axes
141 ax1.xaxis.set_ticks(np.arange(min(current), max(current)+0.1, 0.1)) # Change 0.5 to your desired
142 interval for x-axis
143 ax1.yaxis.set_ticks(np.arange(min(voltage), max(voltage)+0.1, 0.1)) # Change 0.1 to your desired
144 interval for y-axis
145 ax2.yaxis.set_ticks(np.arange(min(power), max(power)+0.1, 0.05)) # Change 0.1 to your desired interval
146 for y-axis
147
148 # Set labels and title
149 ax1.set_xlabel('Current density [A/cm^2$]')
150 ax1.set_ylabel('Voltage [V]')
151 ax2.set_ylabel('Power density [W/cm^2$]')
152 ax1.grid()
153 plt.show()
154
155 def off_design(Altitude, flight_mach, Area, n_cells, OCs, OCrs, POC):
156 # Calculate the air properties at the given altitude
157
158 T0, Tt0, p0, pt0, _, _ = calculate_air_properties(Altitude, flight_mach)
159
160 # Air properties after compressor
161
162 Tt1 = Temp_aft_compress(OCrs['beta'], Tt0, 0.8, 1.4) # Total temperature after compressor
163 pt1 = pt0 * OCrs['beta'] # Total pressure after compressor
164 # A condition to correct the temperature after the compressor in case no power is required by the user
165 if POC == 0:
166     Tt1 = Tt0
167
168 A = 1 # For amphlett model to have everything per unit area cm^2
169 n = 1 # Number of cells for the amphlett model to have all the calculated values per cell

```

```

164 jmax = 1 # Maximum current density in A/cm^2
165
166 PO2 = 0.21 * pt1 / 101325 # Partial pressure of oxygen in atm
167
168 # ##### Humidity calculations
169
170 # pexit = 0.4 * pt1
171 # phi_inlet = 0.25
172 # pwinlet = calc_pw(phi_inlet, Tt1) # NOTE: I am not using an intercooler and therefore the
173 # temperature of the air is the same as the temperature after the compressor
174 # OCs['lambda'], phi_outlet, pwout = calc_lambda(OCs['T_oc'], pexit, OCs['st_air'], dry=False, pinlet=
175 # OCs['Pt1'], pwinlet=pwinlet)
176
177 # # The OEM model does not work for lambda values less than 14 and it will take it as 22 instead
178 # which is not right
179 # if OCs['lambda'] < 14:
180 #     OCs['lambda'] = 14
181
182 amph(A, jmax, n, OCs['p_H2'], PO2, 0, OCs['T_oc'], 0, 0.01, 1, OCrs['1'], OCs['lambda'], 'FC')
183
184 df = pd.read_csv('Amphlett/FC.csv')
185 voltage = list(df['Vcell (V)'])
186 current = list(df['I (A)'])
187 power_FC = list(df['Power (W)'])
188
189 # Initialize last values
190 last_P_effective = None
191 last_P_dens = None
192 last_V_cell = None
193 last_Power = None
194 last_Voltage_fc = None
195 last_mdot_air_in = None
196 last_o2_used = None
197 last_mdot_air_out = None
198 last_h2o_produced = None
199 last_h2_used = None
200 last_P_comp = None
201 last_P_cs = None
202 last_qheat = None
203 last_closest_voltage_index = None
204 last_I = None
205 smallest_diff = None
206
207 for i in current[:power_FC.index(max(power_FC))+1]:
208     P_dens = power_FC[current.index(i)]
209     V_cell = voltage[current.index(i)]
210     Power = P_dens * Area * n_cells * OCrs['n_stacks']
211     Voltage_fc = n_cells * OCrs['n_stacks'] * V_cell
212     mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used = mdots(Power, V_cell) #Calculate
213     mdot_air_in
214     P_comp = compressor_power(OCrs['beta'], mdot_air_in, Tt0, 0.90, 0.8) #Calculate the new compressor
215     power
216     if POC == 0:
217         P_cs, qheat, Voltage_fc = 0, 0, 0
218     else:
219         P_cs, qheat = HEX_Power(OCs['T_oc'], T0, V_cell, Power) # Power required by the cooling system
220     P_effective = Power - P_comp - P_cs
221     closest_voltage_index = voltage.index(V_cell)
222     I = i * Area
223
224     diff = abs(POC - P_effective)
225
226 # Save the current values as the last values
227 last_P_effective = P_effective
228 last_P_dens = P_dens
229 last_V_cell = V_cell
230 last_Power = Power
231 last_Voltage_fc = Voltage_fc
232 last_mdot_air_in = mdot_air_in
233 last_o2_used = o2_used

```

```

230     last_mdot_air_out = mdot_air_out
231     last_h2o_produced = h2o_produced
232     last_h2_used = h2_used
233     last_P_comp = P_comp
234     last_P_cs = P_cs
235     last_qheat = qheat
236     last_closest_voltage_index = closest_voltage_index
237     last_I = I
238
239     # If this is the first iteration, or if this difference is smaller than the smallest so far,
update smallest_diff and the best values
240     if smallest_diff is None or (diff < smallest_diff and P_effective >= POC):
241         smallest_diff = diff
242         best_P_effective = P_effective
243         best_P_dens = P_dens
244         best_V_cell = V_cell
245         best_Power = Power
246         best_Voltage_fc = Voltage_fc
247         best_mdot_air_in = mdot_air_in
248         best_o2_used = o2_used
249         best_mdot_air_out = mdot_air_out
250         best_h2o_produced = h2o_produced
251         best_h2_used = h2_used
252         best_P_comp = P_comp
253         best_P_cs = P_cs
254         best_qheat = qheat
255         best_closest_voltage_index = closest_voltage_index
256         best_I = I
257     # If the difference starts to increase, check if the best_P_effective is higher than POC before
breaking the loop
258     elif diff > smallest_diff:
259         if best_P_effective is not None and best_P_effective >= POC:
260             break
261         else:
262             continue
263
264     # If the loop finished and best_P_effective is not higher than POC, use the last values
265     if best_P_effective is None or best_P_effective < POC:
266         best_P_effective = last_P_effective
267         best_P_dens = last_P_dens
268         best_V_cell = last_V_cell
269         best_Power = last_Power
270         best_Voltage_fc = last_Voltage_fc
271         best_mdot_air_in = last_mdot_air_in
272         best_o2_used = last_o2_used
273         best_mdot_air_out = last_mdot_air_out
274         best_h2o_produced = last_h2o_produced
275         best_h2_used = last_h2_used
276         best_P_comp = last_P_comp
277         best_P_cs = last_P_cs
278         best_qheat = last_qheat
279         best_closest_voltage_index = last_closest_voltage_index
280         best_I = last_I
281
282     # for i in current[:power_FC.index(max(power_FC))+1]:
283     #     P_dens = power_FC[current.index(i)]
284     #     V_cell = voltage[current.index(i)]
285     #     Power = P_dens * Area * n_cells * OCrs['n_stacks']
286     #     Voltage_fc = n_cells * OCrs['n_stacks'] * V_cell
287     #     mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used = mdots(Power, V_cell) #Calculate
mdot_air_in
288     #     P_comp = compressor_power(OCrs['beta'], mdot_air_in, Tt0, 0.94, 0.95) #Calculate the new
compressor power
289     #     if POC == 0:
290     #         P_cs, qheat, Voltage_fc = 0, 0, 0
291     #     else:
292     #         P_cs, qheat = HEX_Power(OCrs['T_oc'], T0, V_cell, Power) # Power required by the cooling
system
293     #     P_effective = Power - P_comp - P_cs
294     #     closest_voltage_index = voltage.index(V_cell)
295     #     I = i * Area

```



```

296 # smallest_diff = None
297 # best_P_effective = None
298
299
300 # diff = abs(POC - P_effective)
301 # if smallest_diff is None or diff < smallest_diff:
302 #     smallest_diff = diff
303 #     best_P_effective = P_effective
304 # # If the difference starts to increase, break the loop
305 # elif diff > smallest_diff:
306 #     break
307 # if Voltage_fc == 800 :
308 #     break
309 # To target the closest power to the required power
310 # if POC <= 60e3:
311 #     error_margin = 0.01
312 #     if POC<= P_effective <= POC+POC*error_margin:
313 #         break
314 # error_margin = 0.1
315 # if POC<= P_effective <= POC+POC*error_margin:
316 #     break
317
318
319
320
321 # n_fac = 0.8 # Factor to calculate n% of the maximum power_FC
322 # voltage_efficiency = 0.56 # Voltage efficiency of the fuel cell
323
324 # voltage_at_efficiency = voltage_efficiency * max(voltage)
325
326 # # Find the value in Voltage_FC that is closest to voltageclose
327 # closest_voltage = min(voltage, key=lambda x: abs(x - voltage_at_efficiency))
328
329 # # Get the index of the closest voltage value
330 # closest_voltage_index = voltage.index(closest_voltage)
331
332 # power_nth_percent = n_fac * max(power_FC)
333
334 # closest_power = None
335 # closest_power_index = None
336
337 # # Find the index of the maximum power_FC
338 # max_power_index = power_FC.index(max(power_FC))
339
340 # # Iterate over the power values
341 # for index, power in enumerate(power_FC[:max_power_index + 1]):
342 #     # If it's the first power value or it's closer to 80% of the maximum power
343 #     if closest_power is None or abs(power - power_nth_percent) < abs(closest_power -
344 # power_nth_percent):
345 #         closest_power = power
346 #         closest_power_index = index
347 #     # If the power is greater than or equal to 80% of the maximum power, stop the loop
348 #     if power >= power_nth_percent:
349 #         break
350
351 # # Size the fuel cell based on the cell voltage and power density [Changed it to take the voltage
352 # index instead of the power index]
353 # V_cell = voltage[closest_voltage_index] # cell coltage at nth% of max power
354 # P_dens = power_FC[closest_voltage_index] # Power density at nth% of max power density
355
356 # #Calculate the power based on the n_cells and the area
357
358 # Power = OCrs['n_stacks'] * n_cells * Area * P_dens
359 # Voltage_fc = n_cells * OCrs['n_stacks'] * V_cell
360
361
362 # #Calculate new mass flow rates
363 # mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used = mdots(Power, V_cell) #Calculate the new
364 # mdot_air_in
365 # P_comp = compressor_power(OCrs['beta'], mdot_air_in, Tt0, 0.94, 0.95) #Calculate the new compressor

```

```

364     power
365     # P_cs, qheat = HEX_Power(OCs['T_oc'], T0, V_cell, Power) # Power required by the cooling system
366
367     # P_effective = Power - P_comp - P_cs
368
369     # Water required to keep the humidity constant
370
371     # mwr = 0.622 * ( pwinlet / (pexit - pwinlet) ) * mdot_air_in
372
373     # humidity_output = [pwinlet, mwr, pwout, phi_outlet, OCs['lambda']]
374
375     # Calculate the efficiencies related to the fuel cell
376     delta_g = (-61.12/2 - 38.96 ) + 306.69 # kJ/mol
377     delta_h = 285.83 # kJ/mol High heating value of water is used [HHV is liquid and LHV is gas because
378     # you can still get energy from the condensation of water]
379     E0 = -237e3 / (2*96485) # V
380     eta_thermo = delta_g / delta_h # Thermo efficiency Thermodynamic efficiency and also the maximum
381     # theoretical efficiency
382     eta_voltage = best_V_cell / abs(E0) # Voltage efficiency
383     eta_fuel = 1 # Fuel utilisation efficiency where 2 is the stoichiometric ratio supplying more fuel
384     # than required in this case air 0.5 ?
385     eta_total = eta_thermo * eta_voltage * eta_fuel # Total efficiency
386
387     # delete_file('Amphlett/FC.csv')
388     # delete_file('Amphlett/FC.html')
389     # delete_file('Amphlett/FC.opem')
390
391     return best_I, best_Voltage_fc, best_P_effective, best_Power, best_P_cs, best_P_comp, best_mdot_air_in
392     , best_o2_used, best_mdot_air_out, best_h2o_produced, best_h2_used, \
393     best_V_cell, best_P_dens, eta_total, eta_voltage, voltage, current, power_FC,
394     best_closest_voltage_index, eta_voltage, best_qheat, Tt1
395
396 def create_plots(TimeP, NEW_power_effectivelist, NEW_Powerturbogeneratorlist, Pw_P, timeF, Zh,
397 NEW_PowerFClist, NEW_h2_usedlist, NEW_o2_usedlist, NEW_fuelGTlist):
398     # Create a figure and a set of subplots
399     fig, ax1 = plt.subplots(figsize=(10, 6))
400
401     # Plot PowerFC, Powerturbogenerator, and h2_used on the first y-axis
402     ax1.plot(TimeP, NEW_power_effectivelist, label='PowerFC', color='blue')
403     ax1.plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
404     ax1.plot(TimeP, Pw_P, label='Power required', color='black')
405     ax1.set_xlabel('Time (s)')
406     ax1.set_ylabel('Power (W)')
407     ax1.tick_params('y', colors='r')
408     ax1.legend(loc='upper left')
409
410     # Create a second y-axis
411     ax2 = ax1.twinx()
412
413     # Plot altitude on the second y-axis
414     ax2.plot(timeF, Zh, label='Altitude', color='purple')
415     ax2.set_ylabel('Altitude (m)')
416     ax1.tick_params('y', colors='b')
417     ax2.legend(loc='upper right')
418
419     plt.grid(True)
420     plt.show()
421
422     # Create a figure and a set of subplots
423     fig, axs = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
424
425     # Plot PowerFC, Powerturbogenerator, and Power required on the first subplot
426     axs[0].plot(TimeP, NEW_power_effectivelist, label='PowerFC Effective', color='blue')
427     axs[0].plot(TimeP, NEW_PowerFClist, 'b--', label='PowerFC total')
428     axs[0].plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
429     axs[0].plot(TimeP, Pw_P, label='Power required', color='black')
430     axs[0].set_ylabel('Power (W)')
431     axs[0].legend()
432     axs[0].grid(True)

```

```

428 # Plot h2_used on the second subplot
429 axs[1].plot(TimeP, NEW_h2_usedlist, label='Hydrogen used by FC', color='green')
430 axs[1].plot(TimeP, NEW_o2_usedlist, label='Oxyegn used by FC', color='blue')
431 axs[1].plot(TimeP, NEW_fuelGTlist, label='Hydrogen used by GT', color='black')
432 axs[1].set_ylabel('Mass flow rate (kg/s)')
433 axs[1].legend()
434 axs[1].grid(True)
435
436 # Plot altitude on the third subplot
437 axs[2].plot(timeF, Zh, label='Altitude', color='purple')
438 axs[2].set_xlabel('Time (s)')
439 axs[2].set_ylabel('Altitude (m)')
440 axs[2].legend()
441 axs[2].grid(True)
442
443 plt.tight_layout(pad=3.0)

```

Listing C.4: Script used to integrate the propulsion system

## C.5. MAIN FILE

```

1 import numpy as np
2 import pandas as pd
3 from fc import *
4 import matplotlib.pyplot as plt
5 from flightdata_reader import read_data, plot_data, get_altitude, get_flying_speed, get_flying_Mach, \
6     calulate_air_properties, get_power
7 from propulsion_system import on_design, plot_polarisation, off_design
8 from GSP_data_postprocessing import read_GSP, get_data, interpolate_wf, interpolate_PWshaft,
9     interpolate_Tt3, \
10     calc_gt_efficiency
11 from system import propulsion_system
12 # from pypsa import statistics
13
14
15 # Read the flight data of Odonata VTOL aircraft
16 timeF, Zh, Pw_r, TimeP, Pw_P, flight_condition, v_x, v_z, flight_data, TimeD = read_data('Flight_Data.csv'
17 )
18
19 # Multiply all the values in Pw_r by 1000
20 Pw_r = [value * 1000 for value in Pw_r]
21 Pw_P = [value * 1000 for value in Pw_P]
22
23 # plot_data(timeF, Zh, Pw_r, TimeP, Pw_P)
24
25 Altitude = get_altitude('Cruise', flight_data)
26 flight_speed = get_flying_speed('Cruise', flight_data)
27 flight_power = get_power('Cruise', flight_data)
28 flight_mach = get_flying_Mach(Altitude, flight_speed)
29
30 # Calculate the air properties at the cruise altitude
31
32 T0, Tt0, p0, pt0, _, _ = calulate_air_properties(Altitude, flight_mach)
33
34 OCs = {
35     'T_oc': 80 + 273.15, # [K] Fuel cell operating temperarure
36     'p_H2': 2.5, # [atm] Hydrogen partial pressure
37     'Pt1': 0.25e6, # [Pa] Required pressure for FC optimum operation
38     'p_O2': 0.21*0.25e6 / 101325, # [atm] Oxygen partial pressure
39     'st_air': 2, # [-] Stoichiometric air condition
40     'lambda': 14, # [-] Maximum water content of the fuel cell
41 }
42
43 OCrs = {
44     'n_stacks': 2, # [-] Number of stacks
45     'l': 0.08, # [cm] Membrane thickness >>>https://doi.org/10.1016/j.compchemeng.2011.03.013
46     'V_req': 800, # [V] Required voltage
47     'P_req': flight_power*10**3, # [W] Required power during cruise

```

```

48     'beta' : OCs['Pt1'] / pt0, # [-] Pressure ratio
49     'eta_v': 0.55,           # [-] Voltage efficiency
50 }
51
52 # etav = 0.35
53
54 # system = propulsion_system(Altitude, flight_mach, OCs, OCrs)
55
56 # system.calculate_air_properties()
57
58 # system.print_air_properties()
59
60
61
62
63 print(f'Pt0: {pt0} Pa')
64 print(f'Tt0: {Tt0} K')
65 print(f'T0: {T0} Pa')
66 print(f'p0: {p0} K')
67
68
69
70 print(f'Mach: {flight_mach}')
71
72
73
74
75 # # print(OCrs['beta'])
76
77 A, n_cells, P_effective, P_new, P_cs, P_comp, mdot_air_in, o2_used, mdot_air_out, h2o_produced, h2_used, \
78 V_cell, P_dens, eta_total, eta_voltage, \
79 voltage, current, power_FC, closest_power_index, n_fac, qheaton, Tt1on = on_design(Altitude, flight_mach,
      OCs, OCrs)
80
81 print(eta_voltage)
82 print(f'A: {A}')
83 print(f'n_cells: {n_cells}')
84 print(f'P_effective: {P_effective}')
85 print(f'P_new: {P_new}')
86 print(f'P_cs: {P_cs}')
87 print(f'P_comp: {P_comp}')
88 print(f'mdot_air_in: {mdot_air_in}')
89 print(f'o2_used: {o2_used}')
90 print(f'mdot_air_out: {mdot_air_out}')
91 print(f'h2o_produced: {h2o_produced}')
92 print(f'h2_used: {h2_used}')
93 print(f'V_cell: {V_cell}')
94 print(f'P_dens: {P_dens}')
95 print(f'eta_total: {eta_total}')
96 print(f'eta_voltage: {eta_voltage}')
97 print(f'Heat Lost: {qheaton}')
98 print(f'FC system efficiency: {P_effective/P_new}')
99 print(OCrs['P_req'])
100 print(flight_power*10**3 - P_effective)
101
102 print(Tt1on)
103
104 print(A)
105 plot_polarisation(current, voltage, power_FC, closest_power_index, n_fac)
106
107 print(h2_used*119*10**6 - (P_comp + P_cs + P_effective + qheaton))
108
109 plot_energy_balance(V_cell, P_new, P_comp, P_cs, P_effective, qheaton)
110
111
112
113 _, Voltage_fc, P_effective_off, Poweroff, P_csoff, P_compoft, mdot_air_in_off, o2_used_off, mdot_air_out_off,
      h2o_produced_off, h2_used_off, \
114     V_cell_off, P_dens_off, eta_total_off, eta_voltage_off, voltage_off, \
115     current_off, power_FC_off, closest_power_index_off, n_fac_off, qheaton, Tt1off = off_design(
      Altitude, flight_mach, A, n_cells, OCs, OCrs, OCrs['P_req'])

```

```

116
117 print(Tt1on, Tt1off)
118
119 print(f'Voltage_fc: {Voltage_fc} ')
120 print(f'P_effectiveoff: {P_effectiveoff} ')
121 print(f'Poweroff: {Poweroff} ')
122 print(f'P_csoff: {P_csoff} ')
123 print(f'P_compoft: {P_compoft} ')
124 print(f'mdot_air_in_off: {mdot_air_in_off} ')
125 print(f'o2_used_off: {o2_used_off} ')
126 print(f'mdot_air_out_off: {mdot_air_out_off} ')
127 print(f'h2o_produced_off: {h2o_produced_off} ')
128 print(f'h2_used_off: {h2_used_off} ')
129 print(f'V_cell_off: {V_cell_off} ')
130 print(f'P_dens_off: {P_dens_off} ')
131 print(f'eta_total_off: {eta_total_off} ')
132 print(f'eta_voltage_off: {eta_voltage_off} ')
133 print(f'closest_power_index_off: {closest_power_index_off} ')
134 print(f'n_fac_off: {n_fac_off} ')
135 print(f'Heat lost: {qheato} ')
136 print(f'FC system efficiency: {P_effectiveoff/Poweroff} ')
137
138
139 print(f'A: {A} ')
140 print(f'n_cells: {n_cells} ')
141 print(f'P_effective: {P_effective} ')
142 print(f'P_new: {P_new} ')
143 print(f'P_cs: {P_cs} ')
144 print(f'P_comp: {P_comp} ')
145 print(f'mdot_air_in: {mdot_air_in} ')
146 print(f'o2_used: {o2_used} ')
147 print(f'mdot_air_out: {mdot_air_out} ')
148 print(f'h2o_produced: {h2o_produced} ')
149 print(f'h2_used: {h2_used} ')
150 print(f'V_cell: {V_cell} ')
151 print(f'P_dens: {P_dens} ')
152 print(f'eta_total: {eta_total} ')
153 print(f'eta_voltage: {eta_voltage} ')
154 print(f'FC system efficiency: {P_effective/P_new} ')
155 print( flight_mach)
156 plot_polarisation(current_off, voltage_off, power_FC_off, closest_power_index_off, n_fac_off)
157
158 # # # Read the GSP data
159
160 rows_to_skip = list(range(2, 989, 17)) + [989, 990, 991, 992] # NaN rows excluding the design point
161
162 df = read_GSP('GTG\DATA_GT3.CSV', rows_to_skip)
163
164 Altitude, Mach, Tt4, PWshaft, WF, W, tt3 = get_data(df)
165
166 # etavs = np.arange(0.30, 0.66, 0.01)
167 etavs = [0.64]
168
169 incremenets = [0.7]
170
171 fcvoltlist = []
172 PowerFClist = []
173 power_effectivelist = []
174 Powerdeficiencylist = []
175 Powerturbogeneratorlist = []
176 coolingpowerlist = []
177 compressorpowerlist = []
178 mdot_air_inlist = []
179 o2_usedlist = []
180 mdot_air_outlist = []
181 h2o_producedlist = []
182 h2_usedlist = []
183 fuelGTlist = []
184 compTt1 = []
185 TiT = []
186 Tt3lst = []

```

```

187 current_list = []
188 efficiency_list = []
189 gtefficiency_list = []
190 vefficiency_list = []
191
192 h2_listfc = []
193 h2Gtlist = []
194 battery_powerlst = []
195 mwrlst = []
196 h2olst = []
197 total_mwrlst = []
198
199 #Getting the payload mass for a range of efficiencies
200
201 Payloadmasslst = []
202 fueltanklst = []
203 fuelcellmasslst = []
204 fuelusedlst = []
205 GTmasslst = []
206 preqinc = []
207 TGmaxpower = []
208 FCmaxpower = []
209 fcnetmaxpower = []
210 compressor_mass= []
211 compressormax = []
212 coolingmax = []
213 coolingsystem_mass = []
214 As = []
215 test = []
216 ns = []
217
218 # for inc in incremenets:
219 for eta in etavs:
220     flight_power = get_power('Cruise', flight_data)
221     # OCrs['P_req'] = flight_power*10**3*inc
222     # preqinc.append(OCrs['P_req'])
223     Altitude = get_altitude('Cruise', flight_data )
224     flight_speed = get_flying_speed('Cruise', flight_data)
225     flight_mach = get_flying_Mach(Altitude, flight_speed)
226     T0, Tt0, p0, pt0, _, _ = calculate_air_properties(Altitude, flight_mach)
227     OCrs['eta_v'] = eta
228
229     A, n_cells, P_effective, P_new, P_cs, P_comp, mdot_air_in, o2_used, mdot_air_out, h2o_produced,
230     h2_used, \
231     V_cell, P_dens, eta_total, eta_voltage, \
232     voltage, current, power_FC, closest_power_index, n_fac, qheaton, Tt1on = on_design(Altitude,
233     flight_mach, OCs, OCrs)
234     As.append(A)
235     ns.append(n_cells)
236     # Pw_r[3] = flight_power*10**3*inc
237     test.append(Pw_r)
238     h2_listfc.clear()
239     h2Gtlist.clear()
240     coolingpowerlist.clear()
241     compressorpowerlist.clear()
242     Powerturbogeneratorlist.clear()
243     PowerFclist.clear()
244     power_effectivelist.clear()
245     mwrlst.clear()
246     h2olst.clear()
247     for altitude, v_x_V, power_flight_data, t in zip(Zh, v_x, Pw_r, TimeD):
248         # Convert v_x to Mach number
249
250         flight_mach = get_flying_Mach(altitude, v_x_V)
251
252         _, Tt0, _, _, _ = calculate_air_properties(altitude, flight_mach)
253         # OCrs['beta'] = OCs['Pt1'] / pt0
254         # Run the off_design function
255         Current, vfc, P_effectiveoff, power_FC_off, power_cooling, power_compressor, mdot_air_in, \
256         o2_used, mdot_air_out, h2o_produced, h2_used, _, _, eta_total, eta_voltage, _, _, _, _, \
257         Tt1off1 = off_design(altitude, flight_mach, A, n_cells, OCs, OCrs, power_flight_data)

```

```

255     if power_flight_data == 0:
256         Tt1off1 = Tt0
257
258     h2ofc = h2o_produced*t
259     h2olst.append(h2ofc)
260     # mwrlst.append(humidoffm[1]*t)
261
262     h2fc = h2_used*t
263     h2_listfc.append(h2fc)
264
265
266     fcvoltlist.append(vfc)
267     PowerFClist.append(power_FC_off)
268     power_effectivelist.append(P_effectiveoff)
269     coolingpowerlist.append(power_cooling)
270     compressorpowerlist.append(power_compressor)
271     mdot_air_inlist.append(mdot_air_in)
272     o2_usedlist.append(o2_used)
273     mdot_air_outlist.append(mdot_air_out)
274     h2o_producedlist.append(h2o_produced)
275     h2_usedlist.append(h2_used)
276     compTt1.append(Tt1off1)
277     current_list.append(Current)
278     efficiency_list.append(eta_total)
279     vefficiency_list.append(eta_voltage)
280
281
282     # Calculate the power deficiency
283     power_deficiency = power_flight_data - P_effectiveoff
284
285     Powerdeficiencylist.append(power_deficiency)
286
287     if power_deficiency > 0:
288         for Tt4 in range(900, 1600, 1):
289             PW_GT = interpolate_PWshaft(flight_mach, Tt4, altitude)* 10**3 * 0.9 # shaft power in KW
290             taking into consideration generator efficiency !!!!
291             if PW_GT >= power_deficiency:
292                 Powerturbogeneratorlist.append(PW_GT)
293                 TiT.append(Tt4)
294                 Tt3 = interpolate_Tt3(flight_mach, Tt4, altitude)
295                 Tt3lst.append(Tt3)
296                 GTfuels = interpolate_wf(flight_mach, Tt4, altitude)
297                 fuelGTlist.append(GTfuels)
298                 GTfuel = GTfuels*t
299                 h2Gtlist.append(GTfuel)
300                 # Calculating efficiency at the point
301                 gt_eff = calc_gt_efficiency(flight_mach, Tt4, altitude, Tt3, PW_GT)
302                 gtefficiency_list.append(gt_eff)
303                 battery_powerlist.append(0)
304                 break
305             elif Tt4 == 1550 and PW_GT < power_deficiency:
306                 battery_power = power_deficiency - PW_GT
307                 Powerturbogeneratorlist.append(PW_GT)
308                 TiT.append(Tt4)
309                 Tt3 = interpolate_Tt3(flight_mach, Tt4, altitude)
310                 Tt3lst.append(Tt3)
311                 GTfuels = interpolate_wf(flight_mach, Tt4, altitude)
312                 fuelGTlist.append(GTfuels)
313                 GTfuel = GTfuels*t
314                 h2Gtlist.append(GTfuel)
315                 gt_eff = calc_gt_efficiency(flight_mach, Tt4, altitude, Tt3, PW_GT)
316                 gtefficiency_list.append(gt_eff)
317                 battery_powerlist.append(battery_power)
318             else:
319                 print('Turbogenerator cannot provide enough power')
320         else:
321             Powerturbogeneratorlist.append(0)
322             fuelGTlist.append(0)
323             TiT.append(Tt0)
324             gtefficiency_list.append(0)
325             # Tt3 = interpolate_Tt3(flight_mach, Tt4, altitude)

```

```

325         Tt3lst.append(Tt0)
326         battery_powerlst.append(0)
327         FCmaxpower.append(max(PowerFClist))
328         fcnetmaxpower.append(max(power_effectivelist))
329         total_fuel = sum(h2_listfc)+sum(h2Gtlist)
330         # total_mwr = sum(mwrlst) - sum(h2olst)
331         # total_mwrlst.append(total_mwr)
332         TGmaxpower.append(max(Powerturbogeneratorlist))
333         fuelusedlst.append(total_fuel)
334         Mass_FC = fc_mass(OCrs['n_stacks'], n_cells, A*10**-4)
335         # Mass_FC = max(PowerFClist)/0.00435e6
336         fuelcellmasslst.append(Mass_FC)
337         GTmass = max(Powerturbogeneratorlist) / 0.00435e6
338         GTmasslst.append(GTmass)
339         tankmass = total_fuel/5.7
340         fueltanklst.append(tankmass)
341         compressor_mass.append(max(compressorpowerlist)*(11/11250))
342         compressormax.append(max(compressorpowerlist))
343         coolingmax.append(max(coolingpowerlist))
344         coolingsystem_mass.append(max(coolingpowerlist) * (11.1 / (1.84e3 + 3.72e3)))
345         payload = calc_payload(Mass_FC, max(Powerturbogeneratorlist), 3175, total_fuel, max(
346             compressorpowerlist), max(coolingpowerlist))
347         Payloadmasslst.append(payload)
348
349         print(f'Altitude: {altitude}, Power (flight data): {power_flight_data}, Power (off design): {
350             P_effectiveoff}, Power deficiency: {power_deficiency}')
351
352     write_lists_to_csv('outputf2.csv', etavs=etavs, As=As, ns=ns, Payloadmasslst=Payloadmasslst,
353         fuelcellmasslst=fuelcellmasslst, fuelusedlst=fuelusedlst, GTmasslst=GTmasslst,
354         TGmaxpower=TGmaxpower, FCmaxpower=FCmaxpower, fueltanklst=fueltanklst,
355         coolingsystem_mass=coolingsystem_mass, compressor_mass=compressor_mass, coolingmax=
356         coolingmax, compressormax=compressormax,
357         fcnetmaxpower=fcnetmaxpower)
358
359     plt.plot(etavs, As)
360     plt.ylabel('Area')
361     plt.xlabel('Voltage efficiency [-]')
362     plt.show()
363
364     print(test)
365     print(As)
366     print(preqinc)
367     print(Payloadmasslst)
368
369     plt.plot(etavs, Payloadmasslst)
370     plt.ylabel('Payload mass [kg]')
371     plt.xlabel('Voltage efficiency [-]')
372     plt.grid()
373     plt.show()
374
375     plt.plot(etavs, fuelcellmasslst)
376     plt.ylabel('Fuel cell mass [kg]')
377     plt.xlabel('Voltage efficiency [-]')
378     plt.grid()
379     plt.show()
380
381     plt.plot(etavs, TGmaxpower, label='Turbogenerator max power')
382     plt.plot(etavs, FCmaxpower, label='FC max power')
383     plt.ylabel('Max power [W]')
384     plt.xlabel('Voltage efficiency [-]')
385     plt.grid()
386     plt.legend()
387     plt.show()
388
389     fig, axs = plt.subplots(2, 1, figsize=(10, 18), sharex=True)
390     axs[0].plot(etavs, fuelusedlst, label='H2 used in total', color='blue')
391     axs[0].set_ylabel('Mass [kg]')
392     axs[0].legend()
393     axs[0].grid(True)

```



```

393
394
395 axes[1].plot(etavs, fueltanklst, label='Tank weight', color='blue')
396 axes[1].set_ylabel('Mass [kg]')
397 axes[1].legend()
398 axes[1].grid(True)
399 axes[1].set_xlabel('Voltage efficiency [-]')
400
401 plt.tight_layout(pad=5.0)
402 plt.show()
403
404
405 fig, axes = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
406 axes[0].plot(etavs, fuelcellmasslst, label='FC mass', color='blue')
407 axes[0].set_ylabel('FC Mass [kg]')
408 axes[0].legend()
409 axes[0].grid(True)
410
411
412 axes[1].plot(etavs, GTmasslst, label='GT mass', color='blue')
413 axes[1].set_ylabel('GT mass [kg]')
414 axes[1].legend()
415 axes[1].grid(True)
416
417 axes[2].plot(etavs, Payloadmasslst, label='Payload Mass available', color='blue')
418 axes[2].set_ylabel('Mass [kg]')
419 axes[2].legend()
420 axes[2].grid(True)
421
422 axes[2].set_xlabel('Voltage efficiency [-]')
423
424 plt.tight_layout(pad=5.0)
425 plt.show()
426
427 print(current_list)
428 print(Pw_r)
429 print(power_effectivelist)
430 print(PowerFclist)
431 print(Powerdefficiencylist)
432 print(Powerturbogeneratorlist)
433 print(len(Powerturbogeneratorlist), len(timeF))
434
435 New_fcvolt = []
436 NEW_PowerFclist = []
437 NEW_power_effectivelist = []
438 NEW_Powerdefficiencylist = []
439 NEW_Powerturbogeneratorlist = []
440 NEW_coolingpowerlist = []
441 NEW_compressorpowerlist = []
442 NEW_mdot_air_inlist = []
443 NEW_o2_usedlist = []
444 NEW_mdot_air_outlist = []
445 NEW_h2o_producedlist = []
446 NEW_h2_usedlist = []
447 NEW_fuelGTlist = []
448 NEW_compTtl = []
449 NEW_TiT = []
450 NEW_battery_powerlst = []
451 New_current_lst = []
452 New_efficiency_lst = []
453 New_gtefficiency_lst = []
454 New_tt3 = []
455 New_vefficiency_lst = []
456
457 index = 0
458 for time in TimeP:
459     # If the time is greater than or equal to the current time in timeF
460     if index < len(timeF) - 1 and time >= timeF[index]:
461         # Move to the next time and power
462         index += 1
463     NEW_PowerFclist.append(PowerFclist[index])

```

```

464     NEW_power_effectivelist.append(power_effectivelist[index])
465     NEW_Powerdeficiencylist.append(Powerdeficiencylist[index])
466     NEW_Powerturbogeneratorlist.append(Powerturbogeneratorlist[index])
467     NEW_coolingpowerlist.append(coolingpowerlist[index])
468     NEW_compressorpowerlist.append(compressorpowerlist[index])
469     NEW_mdot_air_inlist.append(mdot_air_inlist[index])
470     NEW_o2_usedlist.append(o2_usedlist[index])
471     NEW_mdot_air_outlist.append(mdot_air_outlist[index])
472     NEW_h2o_producedlist.append(h2o_producedlist[index])
473     NEW_h2_usedlist.append(h2_usedlist[index])
474     NEW_fuelGTlist.append(fuelGTlist[index])
475     NEW_compTt1.append(compTt1[index])
476     NEW_TtT.append(TtT[index])
477     NEW_battery_powerlst.append(battery_powerlst[index])
478     New_current_lst.append(current_list[index])
479     New_efficiency_lst.append(eficiency_list[index])
480     New_tt3.append(Tt3lst[index])
481     New_gtefficiency_lst.append(gtefficiency_list[index])
482     New_fcvolt.append(fcvoltlist[index])
483     New_vefficiency_lst.append(vefficiency_list[index])
484
485
486     print(f'PowerFList: {PowerFList}')
487     print(f'power_effectivelist: {power_effectivelist}')
488     print(f'Powerturbogeneratorlist: {Powerturbogeneratorlist}')
489     print(f'battery power: {battery_powerlst}')
490     print(f'h2 used by FC: {h2_usedlist}')
491     print(f'fuelGTlist: {fuelGTlist}')
492     print(h2_listfc, h2Gtlist)
493     print(TimeD)
494     print(sum(h2_listfc)+sum(h2Gtlist))
495     # Plot the power deficiency
496
497
498     # Loop to filter out high voltage efficiency
499
500     for v in New_vefficiency_lst:
501         if v > 0.8:
502             New_vefficiency_lst[New_vefficiency_lst.index(v)] = 0
503
504
505     for vi in New_efficiency_lst:
506         if vi >0.8:
507             New_efficiency_lst[New_efficiency_lst.index(vi)] = 0
508
509     write_lists_to_csv('outputfv2.csv', TimeP=TimeP, NEW_power_effectivelist=NEW_power_effectivelist,
510                       NEW_PowerFList=NEW_PowerFList, NEW_Powerturbogeneratorlist=
511                       NEW_Powerturbogeneratorlist, NEW_fuelGTlist=NEW_fuelGTlist, NEW_mdot_air_outlist =
512                       NEW_mdot_air_outlist,
513                       NEW_mdot_air_inlist=NEW_mdot_air_inlist, NEW_h2_usedlist=NEW_h2_usedlist,
514                       NEW_o2_usedlist=NEW_o2_usedlist,
515                       NEW_compTt1=NEW_compTt1, New_current_lst=New_current_lst, NEW_TtT=NEW_TtT,
516                       New_vefficiency_lst=New_vefficiency_lst, NEW_h2o_producedlist=NEW_h2o_producedlist,
517                       NEW_coolingpowerlist=NEW_coolingpowerlist,
518                       NEW_compressorpowerlist=NEW_compressorpowerlist)
519
520     print(sum(NEW_h2_usedlist)+sum(NEW_fuelGTlist))
521     ## Power graph with mass flow rates and current
522
523     fig, axs = plt.subplots(2, 1, figsize=(10, 18), sharex=True)
524
525     # Plot h2_used on the second subplot
526     axs[0].plot(TimeP, NEW_power_effectivelist, label='Net PowerFC', color = 'blue', linestyle = '--')
527     axs[0].plot(TimeP, NEW_PowerFList, label='Gross PowerFC', color = 'blue')
528     axs[0].plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
529     axs[0].plot(TimeP, Pw_P, label='Power required', color='black')
530     axs[0].set_ylabel('Power [W]')
531     axs[0].legend()
532     axs[0].grid(True)
533

```

```

531
532 # Plot altitude on the third subplot
533 axs[1].plot(timeF, Zh, label='Altitude', color='purple')
534 axs[1].set_xlabel('Time (s)')
535 axs[1].set_ylabel('Altitude (m)')
536 axs[1].legend()
537 axs[1].grid(True)
538
539 plt.tight_layout(pad=5.0)
540 plt.show()
541
542
543
544 # Create a figure and a set of subplots
545 fig, ax1 = plt.subplots(figsize=(10, 6))
546
547 # Plot PowerFC, Powerturbogenerator, and h2_used on the first y-axis
548 ax1.plot(TimeP, NEW_power_effectivelist, label='Net PowerFC', color = 'blue', linestyle = '--')
549 ax1.plot(TimeP, NEW_PowerFClist, label='Gross PowerFC', color = 'blue')
550 ax1.plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
551 # ax1.plot(TimeP, NEW_battery_powerlst, label='battery power', color='green')
552 ax1.plot(TimeP, Pw_P, label = 'Power required', color = 'black')
553 ax1.set_xlabel('Time (s)')
554 ax1.set_ylabel('Power (W)')
555 ax1.tick_params('y', colors='r')
556 ax1.legend(loc='upper left')
557
558
559 # Create a second y-axis
560 ax2 = ax1.twinx()
561
562 # Plot altitude on the second y-axis
563 ax2.plot(timeF, Zh, label='Altitude', color='purple')
564 ax2.set_ylabel('Altitude (m)')
565 ax1.tick_params('y', colors='b')
566 ax2.legend(loc='upper right')
567
568 plt.grid(True)
569 plt.show()
570
571
572 # Create a figure and a set of subplots
573 fig, ax1 = plt.subplots(figsize=(10, 6))
574
575 # Plot PowerFC, Powerturbogenerator, and h2_used on the first y-axis
576 ax1.plot(TimeP, New_current_lst, label='cuurent', color = 'blue')
577 ax1.set_xlabel('Time (s)')
578 ax1.set_ylabel('current (A)')
579 ax1.tick_params('y', colors='r')
580 ax1.legend(loc='upper left')
581
582
583 # Create a second y-axis
584 ax2 = ax1.twinx()
585
586 # Plot altitude on the second y-axis
587 ax2.plot(TimeP, New_fcvolt, label='Voltage', color='red')
588 ax2.set_ylabel('Voltage (V)')
589 ax1.tick_params('y', colors='b')
590 ax2.legend(loc='upper right')
591
592 plt.grid(True)
593 plt.show()
594
595
596 ## Power graph with mass flow rates and current
597
598 fig, axs = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
599
600 # Plot h2_used on the second subplot
601 axs[0].plot(TimeP, NEW_h2_usedlist, label='Hydrogen used by FC', color='blue', linestyle='--')

```

```

602 axes[0].plot(TimeP, NEW_o2_usedlist, label='Oxyegn used by FC', color='blue', linestyle='-.')
603 axes[0].plot(TimeP, NEW_fuelGTlist, label='H2 used by GT', color='red', linestyle=':')
604 axes[0].set_ylabel('Mass flow rate (kg/s)')
605 axes[0].legend()
606 axes[0].grid(True)
607
608 axes[1].plot(TimeP, New_current_lst, label='Fuel cell current [A]', color='blue')
609 axes[1].set_ylabel('current (A)')
610 axes[1].legend()
611 axes[1].grid(True)
612
613 # Plot altitude on the third subplot
614 axes[2].plot(timeF, Zh, label='Altitude', color='purple')
615 axes[2].set_xlabel('Time (s)')
616 axes[2].set_ylabel('Altitude (m)')
617 axes[2].legend()
618 axes[2].grid(True)
619
620 plt.tight_layout(pad=5.0)
621 plt.show()
622
623
624 ## Power graph with compressor outlet temperature, tit and and flight e
625
626 fig, axes = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
627
628 axes[0].plot(TimeP, NEW_TtT, label='GT Turbine inlet temperature', color='red')
629 axes[0].plot(TimeP, New_tt3, label='GT Compressor outlet temperature', color='red', linestyle = '--')
630 axes[0].set_ylabel('Temperature [K]')
631 axes[0].legend()
632 axes[0].grid(True)
633
634
635
636 axes[1].plot(TimeP, NEW_compTt1, label='FC Compressor outlet temperature', color='blue')
637 axes[1].set_ylabel('Temperature (K)')
638 axes[1].legend()
639 axes[1].grid(True)
640 # Plot altitude on the third subplot
641 axes[2].plot(timeF, Zh, label='Altitude', color='purple')
642 axes[2].set_xlabel('Time (s)')
643 axes[2].set_ylabel('Altitude (m)')
644 axes[2].legend()
645 axes[2].grid(True)
646
647 plt.tight_layout(pad=5.0)
648 plt.show()
649
650 ## Power graph with efficiency, voltage and and flight e
651
652 fig, axes = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
653
654 # axes[0].plot(TimeP, New_efficiency_lst, label='Fuel cell total efficiency', color='blue')
655 axes[0].plot(TimeP, New_vefficiency_lst, label='Fuel cell voltage efficiency', color='blue', linestyle = '-')
656 axes[0].plot(TimeP, New_gtefficiency_lst, label='Gas turbine efficiency', color='red')
657 axes[0].set_ylabel('Efficiency [-]')
658 axes[0].legend()
659 axes[0].grid(True)
660
661
662 axes[1].plot(TimeP, NEW_power_effectivelist, label='Net PowerFC', color='blue', linestyle = '--')
663 axes[1].plot(TimeP, NEW_PowerFClist, 'b-', label='Gross PowerFC')
664 axes[1].plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='red')
665 axes[1].set_ylabel('Power [W]')
666 axes[1].legend()
667 axes[1].grid(True)
668 # Plot altitude on the third subplot
669 axes[2].plot(timeF, Zh, label='Altitude', color='purple')
670 axes[2].set_xlabel('Time (s)')
671 axes[2].set_ylabel('Altitude (m)')

```

```

672  axs [2].legend()
673  axs [2].grid(True)
674
675  plt.tight_layout(pad=5.0)
676  plt.show()
677
678  # # Create a figure and a set of subplots
679  fig, axs = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
680
681  # Plot PowerFC, Powerturbogenerator, and Power required on the first subplot
682  axs [0].plot(TimeP, NEW_power_effectivelist, label='Net PowerFC', color='blue')
683  axs [0].plot(TimeP, NEW_PowerFClist, 'b--', label='Gross PowerFC')
684  axs [0].plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
685  # axs [0].plot(TimeP, NEW_battery_powerlst, label='battery power', color='green')
686  axs [0].plot(TimeP, Pw_P, label='Power required', color='black')
687  axs [0].set_ylabel('Power (W)')
688  axs [0].legend()
689  axs [0].grid(True)
690
691  # Plot h2_used on the second subplot
692  axs [1].plot(TimeP, NEW_h2_usedlist, label='Hydrogen used by FC', color='blue', linestyle='--')
693  axs [1].plot(TimeP, NEW_o2_usedlist, label='Oxygen used by FC', color='blue', linestyle='-.')
694  axs [1].plot(TimeP, NEW_fuelGTlist, label='H2 used by GT', color='red', linestyle=':')
695  axs [1].set_ylabel('Mass flow rate (kg/s)')
696  axs [1].legend()
697  axs [1].grid(True)
698  # Plot altitude on the third subplot
699  axs [2].plot(timeF, Zh, label='Altitude', color='purple')
700  axs [2].set_xlabel('Time (s)')
701  axs [2].set_ylabel('Altitude (m)')
702  axs [2].legend()
703  axs [2].grid(True)
704
705  plt.tight_layout(pad=5.0)
706  plt.show()
707
708
709  # Create a figure and a set of subplots
710  fig, axs = plt.subplots(3, 1, figsize=(10, 18), sharex=True)
711
712  # Plot PowerFC, Powerturbogenerator, and Power required on the first subplot
713  axs [0].plot(TimeP, NEW_power_effectivelist, label='Net PowerFC', color='blue', linestyle = '--')
714  axs [0].plot(TimeP, NEW_PowerFClist, label='Gross PowerFC', color='blue', linestyle = '-')
715  axs [0].plot(TimeP, NEW_Powerturbogeneratorlist, label='Powerturbogenerator', color='brown')
716  axs [0].plot(TimeP, Pw_P, label='Power required', color='black')
717  # axs [0].plot(TimeP, NEW_battery_powerlst, label='battery power', color='green')
718  axs [0].set_ylabel('Power (W)')
719  axs [0].legend()
720  axs [0].grid(True)
721
722  # Plot h2_used on the second subplot
723  axs [1].plot(TimeP, NEW_compressorpowerlist, label='Compressor Power', color='blue')
724  axs [1].plot(TimeP, NEW_coolingpowerlist, label='Cooling Power', color='red')
725  axs [1].plot(TimeP, NEW_PowerFClist, 'k--', label='Gross Power FC')
726  axs [1].plot(TimeP, NEW_power_effectivelist, label='Net Power FC', color='black')
727  axs [1].set_ylabel('Power (W)')
728  axs [1].legend()
729  axs [1].grid(True)
730  # Plot altitude on the third subplot
731  axs [2].plot(timeF, Zh, label='Altitude', color='purple')
732  axs [2].set_xlabel('Time (s)')
733  axs [2].set_ylabel('Altitude (m)')
734  axs [2].legend()
735  axs [2].grid(True)
736
737  plt.tight_layout(pad=5.0)
738  plt.show()
739
740  # # Plotting the compressor outlet temperature alleen
741  plt.plot(TimeP, New_fcvolt)
742  plt.xlabel('Time (s)')

```

```

743 plt.ylabel('Voltage (V)')
744 plt.title('Fuel cell voltage')
745 plt.grid(True)
746 plt.show()
747
748 ## Plotting the compressor outlet temperature alleen
749 plt.plot(TimeP, NEW_compTt1)
750 plt.xlabel('Time (s)')
751 plt.ylabel('Compressor Outlet Temperature (K)')
752 plt.title('Compressor Outlet Temperature vs Time')
753 plt.grid(True)
754 plt.show()
755
756 ## Plotting the GT compressor outlet temperature alleen
757 plt.plot(TimeP, New_tt3)
758 plt.xlabel('Time (s)')
759 plt.ylabel('GT Compressor Outlet Temperature (K)')
760 plt.title('GT Compressor Outlet Temperature vs Time')
761 plt.grid(True)
762 plt.show()
763
764
765
766
767 ## Plotting the FC efficiency
768 plt.plot(TimeP, New_efficiency_lst, label='FC Efficiency', color='blue')
769 plt.plot(TimeP, New_gtefficiency_lst, label='GT Efficiency', color='red')
770 plt.xlabel('Time (s)')
771 plt.ylabel('Efficiency')
772 plt.title('Efficiency vs Time')
773 plt.legend()
774 plt.grid(True)
775 plt.show()
776
777 plt.plot(TimeP, New_current_lst)
778 plt.xlabel('Time (s)')
779 plt.ylabel('current (A)')
780 plt.title('Current vs Time')
781 plt.grid(True)
782 plt.show()
783
784 # Create a figure and a set of subplots
785 fig, axs = plt.subplots(2, 1, figsize=(5, 7), sharex=True)
786
787 # Plot PowerFC, Powerturbogenerator, and Power required on the first subplot
788 axs[0].plot(TimeP, NEW_compTt1, label='FC Compressor Outlet Temperature', color='blue')
789 axs[0].plot(TimeP, NEW_TtT, label='GT Turbine Inlet Temperature', color='red')
790 axs[0].set_ylabel('Temperature (K)')
791 axs[0].legend()
792 axs[0].grid(True)
793
794 # Plot altitude on the third subplot
795 axs[1].plot(timeF, Zh, label='Altitude', color='purple')
796 axs[1].set_xlabel('Time (s)')
797 axs[1].set_ylabel('Altitude (m)')
798 axs[1].legend()
799 axs[1].grid(True)
800
801 # plt.tight_layout(pad=5.0)
802 plt.show()

```

Listing C.5: Script used to execute the code

## C.6. PROCESSING DATA AND PLOTTING

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from flightdata_reader import read_data
4
5 timeF, Zh, Pw_r, TimeP, Pw_P, flight_condition, v_x, v_z, flight_data, TimeD = read_data('Flight_Data.csv'
)

```

```

6
7
8 # Multiply all the values in Pw_r by 1000
9 Pw_r = [value * 1000 for value in Pw_r]
10 Pw_P = [value * 1000 for value in Pw_P]
11
12 # Read the csv file
13 df = pd.read_csv('outputf1.csv')
14
15 # df2 = pd.read_csv('outputfv2.csv')
16 # df3 = pd.read_csv('outputfv3.csv')
17
18
19 ## Extract data
20
21 # TimeP66 = df2['TimeP']
22 # TimeP55 = df3['TimeP']
23
24 # FCGrossP66 = df2['NEW_PowerFclist']
25 # FCGrossP55 = df3['NEW_PowerFclist']
26
27 # FCnetP66 = df2['NEW_power_effectivelist']
28 # FCnetP55 = df3['NEW_power_effectivelist']
29
30 # GTp66 = df2['NEW_Powerturbogeneratorlist']
31 # GTp55 = df3['NEW_Powerturbogeneratorlist']
32
33 # FCGrossP66 = FCGrossP66 / 1e6
34 # FCGrossP55 = FCGrossP55 / 1e6
35 # FCnetP66 = FCnetP66 / 1e6
36 # FCnetP55 = FCnetP55 / 1e6
37 # GTp66 = GTp66 / 1e6
38 # GTp55 = GTp55 / 1e6
39
40 # plt.plot(TimeP66, FCGrossP66, label='64% FC Gross Power', color='purple')
41 # plt.plot(TimeP66, FCGrossP55, label='55% FC Gross Power', color='royalblue')
42
43 # plt.xlabel('Time [s]')
44 # plt.ylabel('Power [MW]')
45 # plt.title('Power division 55% and 64% fuel cells Over Time')
46 # plt.legend()
47 # plt.grid(True)
48 # plt.show()
49
50 # plt.plot(TimeP66, GTp66, label='TG Power at FC 64%', color='purple')
51 # plt.plot(TimeP66, GTp55, label='TG Power at FC 55%', color='royalblue')
52
53 # plt.xlabel('Time [s]')
54 # plt.ylabel('Power [MW]')
55 # plt.title('Turbogenerator Power at 55% and 64% fuel cells Over Time')
56 # plt.legend()
57 # plt.grid(True)
58 # plt.show()
59
60 ## Plotting the data
61 # plt.figure(figsize=(6, 6)) # Optional: Adjusts the figure size
62
63 # plt.plot(TimeP66, FCGrossP66, label='FC Gross Power P66')
64 # plt.plot(TimeP66, FCGrossP55, label='FC Gross Power P55', linestyle='--')
65 # plt.plot(TimeP66, GTp66, label='GT Power P66')
66 # plt.plot(TimeP66, GTp55, label='GT Power P55', linestyle='--')
67 # plt.plot(TimeP66, FCnetP66, label='FC Net Power P66', linestyle='-.')
68 # plt.plot(TimeP66, FCnetP55, label='FC Net Power P55', linestyle=':')
69
70 # plt.xlabel('Time [s]')
71 # plt.ylabel('Power [W]')
72 # plt.title('Power division 55% and 66% fuel cells Over Time')
73 # plt.legend()
74 # plt.grid(True)
75 # plt.show()
76

```

```

77 ncells = df['ns']
78 efficiency = df['etavs']
79 areas = df['As']
80 fcnet = df['fcnetmaxpower']
81 fcgross = df['FCmaxpower']
82 pcomp = df['compressormax']
83 pcs = df['coolingmax']
84 tgpower = df['TGmaxpower']
85 compmass = df['compressor_mass']
86 csmass = df['coolingsystem_mass']
87 fcnetnew = []
88
89
90
91 for pfcg, pc, pcool in zip(fcgross, pcomp, pcs):
92     fcnetnew.append((pfcg - pc - pcool)/1e6)
93
94 print(fcnetnew)
95 # Create the plot
96 # plt.plot(efficiency, ncells)
97 # plt.xlabel('Voltage efficiency [-]')
98 # plt.ylabel('Number of cells [-]')
99 # plt.title('Efficiency vs number of cells')
100 # plt.grid(True)
101 # plt.show()
102
103 # Convert values to MW
104 fcnet = fcnet / 1e6
105 # fcnetnew_MW = fcnetnew / 1e6
106 pcomp = pcomp / 1e6
107 pcs = pcs / 1e6
108 fcgross = fcgross / 1e6
109 tgpower = tgpower / 1e6
110
111 # # Create the plot
112 # plt.plot(efficiency, compmass, label='Compressor mass')
113 # plt.plot(efficiency, csmass, label='Cooling system mass')
114
115 # plt.xlabel('Voltage efficiency [-]')
116 # plt.ylabel('Mass [kg]')
117 # plt.title('Efficiency vs Mass')
118 # plt.legend()
119 # plt.grid(True)
120 # plt.show()
121
122 # Create the plot
123 plt.plot(efficiency, fcnetnew, label='Fuel cell net power')
124 plt.plot(efficiency, pcomp, label='Compressor power')
125 plt.plot(efficiency, pcs, label='Cooling system power')
126 plt.plot(efficiency, fcgross, label='Fuel cell gross power')
127 plt.plot(efficiency, tgpower, label='Turbogenerator power')
128
129 plt.xlabel('Voltage efficiency [-]')
130 plt.ylabel('Power [MW]')
131 plt.title('Efficiency vs Power')
132 plt.legend()
133 plt.grid(True)
134 plt.show()
135
136 # # Create the plot
137 # plt.plot(efficiency, areas)
138 # plt.xlabel('Voltage efficiency [-]')
139 # plt.ylabel('Cell Area [cm^2]')
140 # plt.title('Efficiency vs Fuel cell area')
141 # plt.grid(True)
142 # plt.show()

```

**Listing C.6:** Script used to plot the output data