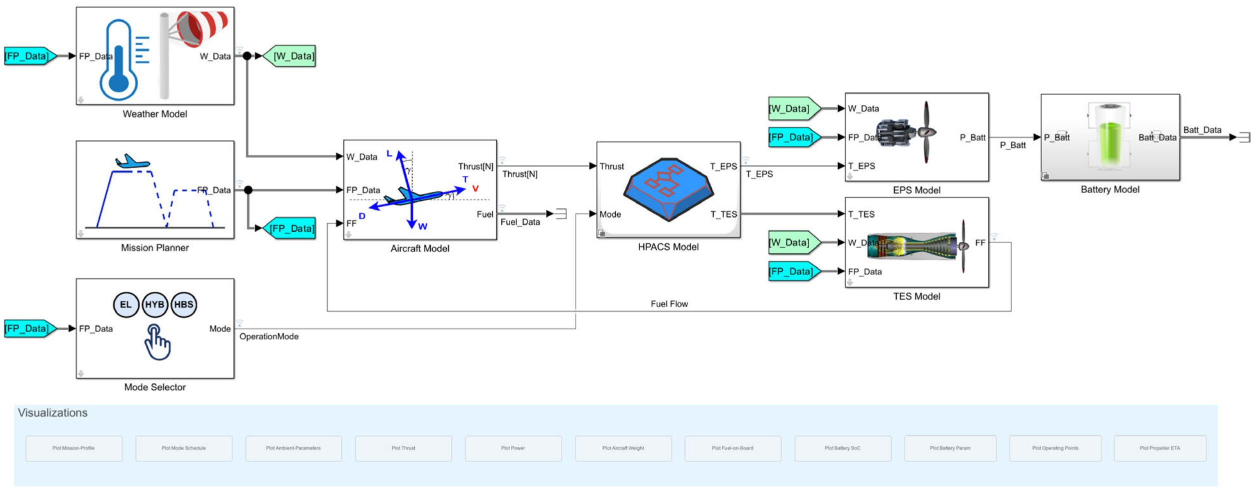


Pre-flight Mission Simulation Model for Estimating State-of-Energy (SoE) in Hybrid Aircraft

Design and Development of Aircraft Performance and Mission Simulation Model, and Analysis of Dynamic Programming Based Optimal Energy Management Strategy

Master’s Thesis in Systems Control and Mechatronics

W A Volanka Stepson
Nilantha Dadallage

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master Degree project report 2024

Pre-flight Mission Simulation Model for Estimating State-of-Energy (SoE) in Hybrid Aircraft

Design and Development of Aircraft Performance and Mission Simulation Model, and Analysis of Dynamic Programming Based Optimal Energy Management Strategy

W A Volanka Stepson

Nilantha Dadallage

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2024

Pre-flight Mission Simulation Model for Estimating State-of-Energy(SoE) in Hybrid Aircraft
Design and Development of Aircraft Performance and Mission Simulation Model, and Analysis of Dynamic Programming Based Optimal Energy Management Strategy

W.A.Volanka Stepson
Nilantha Dadallage

Supervisor: Etienne Lemarchand, Heart Aerospace
Supervisor and Examiner: Anders Grauers, Department of Electrical Engineering, Chalmers University of Technology

Masters Thesis 2024
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Illustration of the pre-flight mission simulation model in MATLAB Simulink, including system models and user interface for generating simulation results.

Typeset in LaTeX
Gothenburg, Sweden 2024

Pre-flight Mission Simulation Model for Estimating State-of-Energy (SoE) in Hybrid Aircraft
Design and Development of Aircraft Performance and Mission Simulation Model, and Analysis of Dynamic Programming Based Optimal Energy Management Strategy

W.A.Volanka Stepson | Nilantha Dadallage
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Accurate estimation of state-of-energy (SoE) is crucial for energy management functions within the aircraft Flight Management System (FMS). In the context of hybrid-electric aircraft, where multiple energy sources are available, estimating SoE becomes a complex challenge. To address this, a pre-flight mission simulation model has been developed and verified for hybrid-electric aircraft. This model serves two primary purposes: first, it aids in developing energy management related FMS functions for hybrid-electric aircraft; second, the simulation model acts as an aircraft performance model, which can be used for developing energy management strategies for hybrid-electric aircraft. The simulation model facilitates detailed mission planning, simulation of different atmospheric conditions, while the simulation model accounts for varying aircraft weight. The methodology for developing and verifying the simulation model is described, and simulation results for various aircraft missions are presented. Furthermore, Dynamic Programming (DP) based optimal energy management strategy is proposed. This algorithm generates the optimal mode-of-operation profile for a given flight mission, ensuring that the specified battery energy reserves remain intact while minimizing the fuel consumption. The performance of this algorithm is analyzed using the simulation model for various missions.

Keywords: modeling, simulation, hybrid-electric aircraft, flight management system (FMS), pre-flight mission simulation, state-of-energy (SoE), energy management strategy, optimization, dynamic programming (DP)

Acknowledgements

We would like to express our sincere gratitude to our industrial supervisor at Heart Aerospace, Etienne Lemarchand, for his invaluable guidance and encouragement throughout the thesis project. His willingness to share his time and expertise has been immensely helpful.

Our heartfelt thanks go to our academic supervisor and examiner at Chalmers, Professor Anders Grauers, for his technical expertise, generous time, and enthusiastic support. His insights and dedication have been instrumental to our work.

Lastly, we extend our deepest appreciation to our families for their unwavering support throughout this journey. Their encouragement has been a constant source of strength.

Volanka Stepson
Nilantha Dadallage
Gothenburg
June 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AGL	Above Ground Level
AMSL	Above Mean Sea Level
CAS	Calibrated Air Speed
CoG	Centre of Gravity
DISA	Delta ISA
DP	Dynamic Programming
EPS	Electrical Propulsion System
EPACS	Electrical Propulsion Actuation and Control System
ECM	Equivalent Circuit Model
FMS	Flight Management system
FoB	Fuel on Board
HPACS	Hybrid Propulsion Automated Control System
IAS	Indicated Airspeed
ISA	International Standard Atmosphere
MSL	Mean Sea Level
OAT	Outside Air Temperature
OCV	Open Circuit Voltage
PA	Pressure Altitude
PBU	Propulsion Battery Unit
PMSM	Permanent Magnet Synchronous Motors
RoC	Rate of Climb
RoD	Rate of Descent
SAF	Sustainable Aviation Fuel
SoC	State of Charge
SoE	State of Energy
TAS	True Air Speed
TES	Turbine Engine System
ToC	Top of Climb
ToD	Top of Descent

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

t	Index for continuous time
k	Index for discrete time step

Sets

W	Set of possible shaft speed values
W_{valid}	Set of valid shaft speed values
$\mathcal{X}$	Set of state values
$\mathcal{U}$	Set of control state values

Parameters and Variables

h	Altitude
v	Aircraft true air speed (TAS)
$\dot{v}$	Aircraft acceleration
γ	Flightpath angle
v_{Wind}	Wind speed
$disa$	Temperature deviation from ISA
qnh	QNH
h_p	Pressure altitude
P_s	Static pressure
ρ	Air density

a	Speed of sound
m_{zf}	Aircraft zero-fuel weight
$m_{\text{fuel,init}}$	Initial FoB
SoC_{init}	Initial PBU SoC
m	Aircraft weight
D_{aero}	Aerodynamic drag
$D_{friction}$	Friction force
μ	Friction coefficient
C_D	Drag coefficient
C_{D_0}	Zero-lift drag coefficient
C_L	Lift coefficient
C_T	Propeller thrust coefficient
C_P	Propeller power coefficient
J	Propeller advance ratio
M_{tip}	Propeller tip mach number
η	efficiency
x^{soc}	PBU SoC state
x^{fob}	Remaining FoB state
N_U	Discretization for control input
N_{soc}	Discretization for PBU SoC state
N_{fob}	Discretization for FoB state
k_p	Control state penalizing factor

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

The air transportation sector continues to be a fundamental element of the global economy, serving both passenger and cargo transportation and it experiences rapid expansion. The latest figures in January 2024 indicate an increase of total passenger traffic by 16.6% year-on-year revenue-passenger kilometers, and marks a remarkable resilience reaching 95.7% of pre-pandemic figures[1]. This demonstrates a strong recovery and suggests a bright future for the sector, underscoring its ability to meet and potentially exceed previous air passenger traffic projections. This data suggests that the air transportation is rapidly expanding in direct correlation with the growing demand. However, it accounts for approximately 2.5% of global CO_2 emissions, and it is predicted to increase with the high demand and the annual growth of the aviation industry[2] [3]. The contribution of emissions from short-to-medium-haul flights is significant due to their high fuel consumption per passenger kilometer[4]. These flights need attention and innovative solutions within the aviation sector to mitigate their substantial carbon footprint[4]. To mitigate the impact of air transportation on the environment, challenging goals have been set by the Advisory Council for Aviation Research and Innovation in Europe (ACARE). As per the ACARE's Flightpath 2050 Europe's Vision for a clean sky program, a 75% reduction of CO2 emissions per passenger kilometer, a 90% reduction in $NO_\times$ emissions, and a 65% reduction in the perceived noise emission are to be achieved by 2050[5].

To meet these emission reduction goals, several futuristic electric aircraft concepts have been proposed by industry leaders, academia, governmental bodies, and start up companies[3]. They have employed different technologies such as electric vertical takeoff and landing (eVTOL) and fixed-wing all-electric and fixed-wing hybrid-aircraft, and are expected to enter service at different time frames. Although all-electric airplanes offer benefits, including zero CO_2 emissions, low noise, and low operational costs, there are many challenges to overcome to produce all-electric airplanes that can meet the needs of commercial aviation. One of the main factors preventing developments of the all-electric commercial airplanes is the limited energy density of today's battery technology. In addition to that, the current electric motor technology also has limitations to overcome. Studies show that motors need to achieve specific power densities over 20kW/kg with efficiencies greater than 96% to realize commercially viable all-electric airplanes[6]. These are technically challenging targets and therefore, all-electric airplanes cannot match the range of fossil fuel-driven commercial airplanes with the current technology.

The hybrid-electric airplanes have gained attention as they can leverage the benefits of both electric and fossil-fuel driven technologies. Hybrid-electric airplanes can achieve a fraction of the CO_2 emission goals while addressing the range limitations inherent to all-electric airplanes. Hybrid-electric airplanes use two or more energy sources with different configurations to power the propulsion systems of the airplanes[7]. The multiple sources of energy in the propulsion systems in the hybrid-electric airplanes results in added complexities, especially in terms of energy management. Optimal management of hybrid-energy for a flight mission ensuring minimum emissions, maintaining required energy reserves without compromising the safety of the flight is a complex problem[8].

1.2 Heart Aerospace and ES-30 Aircraft

Heart Aerospace, a startup company based in Gothenburg, Sweden, is committed to transform the future of air travel with the mission to decarbonize and democratize air travel. Heart Aerospace aims not only to lower the emissions, but to make flying accessible for the many around the world by unlocking cheaper, convenient and effective regional air travel. Since, the hybrid-electric airplanes are cheaper to operate, they will enable increased connectivity by reopening routes that were previously unprofitable with conventional fossil-fuel based aircraft. They aim to cut greenhouse gas emissions by 20% to 95% in short-haul flights that currently contributes to one-third of the aviation industry's emissions.

Heart Aerospace is developing ES-30, a 30 seat regional hybrid-electric aircraft aiming to enter the market around 2030. The ES-30 offers 200km all-electric range and optimized hybrid-electric range up to 400km with full passenger capacity. It has the capability to extend range up to 800km with reduced passenger capacity of 25 providing, airlines with greater operational flexibility. The turnaround time for the ES-30 is expected to be 30 minutes with fast charging capability while the runway length needed for operating ES-30 is 1100m.

The ES-30 consists of two electric propulsion units mounted in-board on the wing and two small turboprop engines mounted out-board on the wing, in a novel independent hybrid propulsion architecture as shown in Figure(1.1). The primary source of energy for ES-30 is the electric energy, and fuel is used for reserve and range extension purposes. The turboprop engines compatible with Sustainable Aviation Fuel (SAF) reduce the carbon footprint of ES-30 in the extended range missions. ES-30 can perform low-noise takeoffs and landings reducing the sound pollution near airports. Additional to these advantages, the low electricity cost compared to jet-fuel and low maintenance cost of electric motors offer lower operating costs for the airlines.

Figure 1.1: Heart Aerospace ES-30 hybrid-electric regional aircraft.

1.3 Purpose, Scope and Contribution

The purpose of this project is to develop the concept of energy management in ES-30 hybrid-electric aircraft. This work involves the concept development of energy management functions for the ES-30 and defining the functional interfaces for them. The energy management function for ES-30 encompasses various functions including pre-flight and in-flight functions. The primary emphasis of this project was on the pre-flight energy management function of ES-30, and the scope of the work was narrowed down to the pre-flight estimation of state-of-energy (SoE) of the ES-30 aircraft for a given mission.

The main contribution from this work lies in the development and verification of the comprehensive simulation model that focuses the pre-flight estimation of the SoE in ES-30 aircraft for given flight mission. While this simulation model can aid in the development of energy management related Flight Management System (FMS) functions, it also can serve as an aircraft performance model for analyzing and evaluating the energy management strategies specific to the ES-30 aircraft. What sets this simulation model apart from existing literature is its multi-disciplinary approach which accounts for the varying aircraft weight and atmospheric conditions. The simulation model is designed to be modular and flexible, allowing for future adaptations based on company requirements. Numerous simulation analyses were conducted and presented, revealing the impact of different weather parameters, aircraft weight on the aircraft state-of-energy (SoE). Further, simulation based analysis results to find optimal cruise altitude and cruise speed for different goals such as minimum time en-route, minimum fuel consumption and minimum battery energy consumption is presented.

Additionally, an optimal energy management strategy based on Dynamic Programming (DP) is proposed for ES-30. The algorithm determines the most effective operation mode profile for a given flight mission, ensuring that the specified reserves remain intact with minimum fuel consumption. The performance of this energy management strategy is analysed using the developed simulation model for various flight missions.

1.4 Thesis Outline

The Section 2 comprises of a detailed literature survey, which summarizes different simulation models and techniques found in the literature related to the interested problem, followed by different the energy management strategies used in hybrid-electric aircraft.
The Section 3 presents the theoretical background arranged in different topics. The readers can enhance their knowledge about different areas including aircraft FMS, simulation approaches, aviation weather conditions, aircraft performance and dynamics, aircraft propulsion systems and system modeling techniques followed by theory related to Dynamic Programming (DP) based optimization.
Section 4 explains the steps followed, and techniques used in modeling the different system models of the simulation model. This includes the modeling architecture for each model, considered input and output signals along with the practical aspect of modeling different system models. Additionally, it presents the verification process and the corresponding results. At last, the problem formulation for DP based energy management strategy is explained.
Section 5 starts with presenting the obtained simulation model, and an explanation of the readily available plots from the simulation model. Then, the various simulations and the obtained simulation results are presented, also the findings from the simulation results are discussed. Finally, the DP based optimal energy management related results are presented and discussed.
Finally, Section 6 summarizes the outcomes of the thesis highlighting the main findings and offers suggestions for future work.

** Due to confidentiality reasons and at the company's request, sensitive information has been replaced with the placeholder "x", and the axes containing sensitive information in the diagrams have been obscured **

1.5 Ethics

Reducing the impact of air transportation on the environment stands as a primary goal for the entire ES-30 project and the company. The successful outcome of this project will contribute to Heart Aerospace's overarching mission: to create the world's greenest, most affordable, and most accessible form of transportation.

This project exposes the team members to sensitive information and intellectual properties (IP) of Heart Aerospace. These sensitive information will be handled with utmost care throughout the project. Our approach throughout the project will demonstrate integrity in all aspects, and the scientific research and information sources used in our project work will be diligently cited, credited, and acknowledged.

2

Literature Survey

A comprehensive literature survey was conducted to explore the state of the art and available simulation tools for estimating the SoE in hybrid aircraft. Additionally the literature survey was expanded to study the energy management strategies employed in hybrid aircraft.

2.1 Simulation Tools for Hybrid Aircraft

With the increased interest in hybrid-electric aircraft in recent years, there is strong interest in research and development of tools and method for different purposes from analysing and quantifying fuel saving potential to design space exploration in hybrid electric aircraft.

A physics-based simulation platform has been developed by Zhao et al. for design, optimization and analysis of hybrid aircraft [9]. This platform uses comprehensive gas turbine performance models based on thermodynamic models. The electrical power system consists of electrical machine, power electronics converter and battery models. The mission model in the platform defines the flight profile trajectory in terms of altitude and Mach number. Further, it consists of an aircraft structural model and a thermal management system models. The platform is developed in Python environment and the platform has been able to analyse trade-offs between sizing and performance characteristics of different subsystems and also many parameters including state of charge of the battery and gas turbine related efficiencies and losses in different sections of the turbine engines.

Cameretti et al. developed a code based model to simulate a flight mission for a turboprop based aircraft with two other gas turbines in hybrid configuration [10]. Numerical investigations have been conducted using a commercial code for the flight simulation initially, followed by MATLAB based optimization for calculating battery pack weight. The turbine engine modelling has been done considering the thermodynamic cycle of turbine engine. The lack of public information for turbine engine components performance maps had been a challenge. Specific fuel consumption map has been used to determine the fuel consumption of the turbine engines. The simulation is based on the mission profile derived from a commercial flight information extracted from Flightradar24 and the considered parameters for the mission profile were altitude and speed versus time. The power profile was calculated through mechanical flight equations implemented in MATLAB considering the altitude and speed profile along with aerodynamic data of ATR42-300 aircraft. This model could compare fuel consumption and emissions for the hybrid configuration.

Modelica based flexible and modular, hybrid-electric aircraft modeling architecture has been developed by Batteh et al. to be used in virtual and physical demonstrator system, National Research Council of Canada (NRC) [11]. This architecture uses multi-physics component models to describe the performance and dynamics of the different systems in hybrid-aircraft. The Aircraft Dynamics Library in Modelon has been extensively utilized for the implementation of the models. 6DOF representation of the aircraft dynamics has been used with a simplified high level turboprop surrogate model as the power train. Analyses on integrated electric aircraft performance has been conducted using Modelon libraries and extended range of mission profiles including takeoffs and landing has been simulated. This model could simulate and the energy consumption as well as the dynamics of the aircraft and and also the aircraft-sub systems.

PLAtform for New Environment-friendly Solutions (PLA.N.E.S), a simulation software for assessing performance, environmental impact and advanced configurations have been developed and validated by Donateo et al [12]. The simulation is based on backward paradigm also know as inverse simulation approach where the flight mission is assumed to be known, and the instantaneous power required and the corresponding fuel consumption is calculated. PLA.N.E.S consists of many power-train configurations to be chosen by the user including conventional, all-electric, different hybrid configurations etc. Further the platform consists of series of blocks for mission generation, simulation, post processing and optimization. PLA.N.E.S has been applied to a medium altitude medium endurance unmanned aerial vehicle as a case study [12].

A simulation tool has been developed to analyse the mission performance of regional hybrid-electric aircraft dynamics by Palaia et al. [13]. A point-mass model has been used to model the aircraft dynamics in the longitudinal plane of the aircraft. The simulation have been conducted for a parallel hybrid-electric power-train architecture where thermal engine and electric motor can independently supply power to the same propeller. Power management strategy have been considered where the power required to fly the aircraft have been spllited between the thermal engine and the electric motor. Average constant efficiency values for propeller efficiency and gear efficiency have been used for computing the required power. The analysis results obtained by the simulation for parallel hybrid aircraft configuration has shown considerable potential in terms of reduction of fuel consumption. Further the simulation results from this tool has given insights into operational envelopes as a function of number of passengers and aircraft range.

Mission, Aircraft and System Simulation (MASS), a modular and parametric tool chain has been developed for analysing the performance of hybrid-electric aircraft for a given reference mission under European Union's Horizon 2020 research and innovation program [14]. The parametric analysis tool chain allows for performance analysis of hybrid-electric aircraft and also it allows sensitivity analysis and optimization studies to support conceptual and multi-disciplinary aircraft design with hybrid-electric propulsion systems. Longitudinal flight path without turning movements are assumed and the aircraft point-mass model considering the longitudinal movement of the aircraft has been implemented within MATLAB environment. Mission model in the tool chain reads a pre-defined table with altitude, speed, flap and landing gear setting as a function of horizontal distance. The aircraft model computes the required thrust from the flight path variables and the corresponding aerodynamic coefficients based on flaps and landing gear setting. A simulation study has been performed for an Airbus A320neo reference aircraft and parameters such as fuel consumption and total energy consumption have been analysed.

Most of the simulation approaches rely on constant efficiency values derived from an average operating point approach for modeling the propeller, motor and turbine engine performance [14] [13]. Further, the mission profiles considered in most of the simulation tools are simplified mission profiles considering altitude and speed of the aircraft [10] as the primary flight path variables. It is crucial to recognize that the ambient conditions play a significant role in flight operations. Factors such as headwind component and ambient temperature can profoundly affect both aircraft and propulsion system performance. However, the ambient parameters have often been overlooked in the existing simulation tools available in the literature.

The simulation model proposed in this work possesses the capability to generate a detailed mission profile that closely align with the ES-30 hybrid-electric aircraft missions. Detailed performance maps provided by the company are used to model various sub-systems to ensure that the model captures the real-world system performance as close as possible. Further, a weather model has been implemented to provide accurate ambient parameters for the simulation, effectively addressing a current research gap.

2.2 Energy Management in Hybrid Aircraft

Hybrid-electric aircraft in contrast to a conventional aircraft has extra degree of freedom in delivering the requested propulsion power as it has multiple sources of energy onboard. Energy management in a hybrid-aircraft is managing these multiple energy sources during a mission, fulfilling operational and safety requirements while the ensuring the energy sources operate within their limits [15]. Many energy management strategies have been explored in the hybrid-electric vehicle context, and these methods have been adopted for energy management in the hybrid-electric aircraft.

The main energy management strategies found in literature can be divided into two, which are rule-based and optimization-based energy management strategies.
The rule-based energy management strategies rely on control schemes governed by predefined rules and operational modes. These rules are determined by the experts considering the performance characteristics and limitations of the different

propulsion systems. These rules are implemented in supervisory controllers in the form of state-machines, look-up tables or similar. The rule-based strategies take instantaneous power requests and other relevant parameters as inputs, providing corresponding outputs based on these input parameters. Notably, rule-based strategies focus solely on instantaneous inputs. Therefore, the rule-based strategies often do not require knowledge about the future trajectory of the aircraft to take control decision at the present moment.

When optimization techniques are used for the energy management strategies, they are referred to as optimal energy management strategies. One such well-known strategy is the Equivalent Consumption Minimization Strategy (ECMS), which relies on an instantaneous cost function. The primary objective of ECMS is to minimize energy consumption by identifying local minima among all feasible instantaneous operating points for both electric and fuel engines. The cost function in ECMS in hybrid-electric context introduces a equivalence factor which converts electrical energy consumption into equivalent fuel consumption to unify the electric energy and fuel consumption to be used in the cost function. As ECMS solves a local instantaneous optimization problem, this method is suitable for online application such as in-flight energy optimization as it is computationally efficient.

Dynamic Programming (DP) is a numerical optimization method based on Bellman's principle of optimality [16]. The DP algorithms are widely used in energy management problems to provide global optimal solution. The application of DP for energy management problem in hybrid electric aircraft has been investigated for parallel hybrid-electric Unmanned Air Vehicle(UAV) [17]. This algorithm optimizes power management and torque-split of the hybrid power-train while considering final state constraints on state variables. Applying DP algorithm for energy management in fuel-cell powered hybrid-electric aircraft has been studied accounting the varying aircraft weight into consideration [18]. An aircraft performance model has been used along with DP algorithm to find the optimal energy management strategy for an aircraft flying an pre-defined trajectory. A method to determine the optimal energy management for hybrid aircraft has been studied considering the battery State-of-charge (SoC) and aircraft weight as the states of the system [19]. Further optimal strategies with constrained and unconstrained final SoC targets have been explored. Comparison results of DP algorithm with other energy management strategies has shown DP performs superior in terms of minimum fuel consumption objective.

While the DP algorithm requires the mission profile to be known a priori, the computational cost increases exponentially with the number of states of the problem, which is often referred to as "the curse of dimensionality". Although these characteristics of DP algorithm often hinders the usage of DP for real-time applications, DP based algorithms can be used for development and performance evaluation of the rule-based strategies as it provides globally optimal solution.

3

Theory

3.1 Flight Management System and Energy Management in Hybrid Aircraft

Flight Management System (FMS) plays a pivotal role in modern aircraft avionics. It seamlessly automates a wide variety of pre-flight and in-flight tasks related to planning, navigation and performance. By handling these responsibilities, the FMS significantly reduces the workload on the flight crew. The FMS receives inputs from various sources, perform computations related to various aspects of an entire flight and delivers accurate information to the flight crew or control commands to the aircraft systems ensuring both safety and operational efficiency of the flight [20]. Some of the typical functions of a FMS in a conventional aircraft are,

1. Flight profile planning and optimization
2. Providing aircraft performance information
3. Determining flight routes for the aircraft to achieve strategic goals
4. Guiding the aircraft along the computed flight profile and monitoring
5. Deriving the optimal speed and altitude based on accumulated performance data

From a system perspective, the FMS comprises several interconnected systems and components, each playing a crucial role. Among these, the following can be considered the main components.

1. Flight Management Computer (FMC):
   The FMC serves as the central brain for the FMS where it handles multitude of critical computations relevant to the flight such as performance computation, fuel and energy management related computations, flightpath adjustment considering the weather, wind and other relevant parameters.

2. Automatic Flight Guidance System (AFGS):
   AFGS is an integral part of the FMS and it is responsible for executing the computed flight path. In the modern aircraft, the FMS can compute four dimensional flight path (longitude, latitude, altitude and time), and do the necessary changes to the flight path via AFGS. It maintains precise control during various flight phases from takeoff to landing. Further, at any time the FMS commands can be overruled by the pilots using Flight Control Unit(FCU)

or flying the aircraft manually.

3. Navigation Systems:
   The FMS can perform navigation related computations for an entire flight mission from departure airport to the destination via waypoints, using airports databases and navigational aids also supplemented by various sensor inputs.

4. Electronic Flight Instrument System (EFIS):
   The EFIS serves as the human-machine interface within the aircraft cockpit. In a simple general aviation aircraft the EFIS can be one or few displays or interfaces while the EFIS is a modern airliner can comprise of a wide variety of displays and instruments. The main components of an EFIS are Primary Flight Displays(PFD), Multi-Function Displays(MFD), Engine Indication and Crew Alerting System (EICAS) and Multi-purpose Control and Display Units(MCDU). With the current advancement of avionics, most of these can feature touch-screens combining display and control functions and serve multi-purpose.

   As shown in Figure(3.1), the different avionics are distributed throughout the cockpit, interfaced together to work hand-in-hand to serve the FMS functions.

Figure 3.1: Illustration of ES-30 cockpit avionics suite.

In the context of hybrid-electric aircraft, certain functions of the FMS take on greater prominence and complexity as the hybrid-electric aircraft uses multiple energy sources. The FMS must intelligently manage these diverse energy sources to optimize the performance and efficiency of the flight. A conceptual interface block diagram indicating the interfacing of different systems and components to form the FMS in a hybrid-electric aircraft is depicted in Figure(3.2).

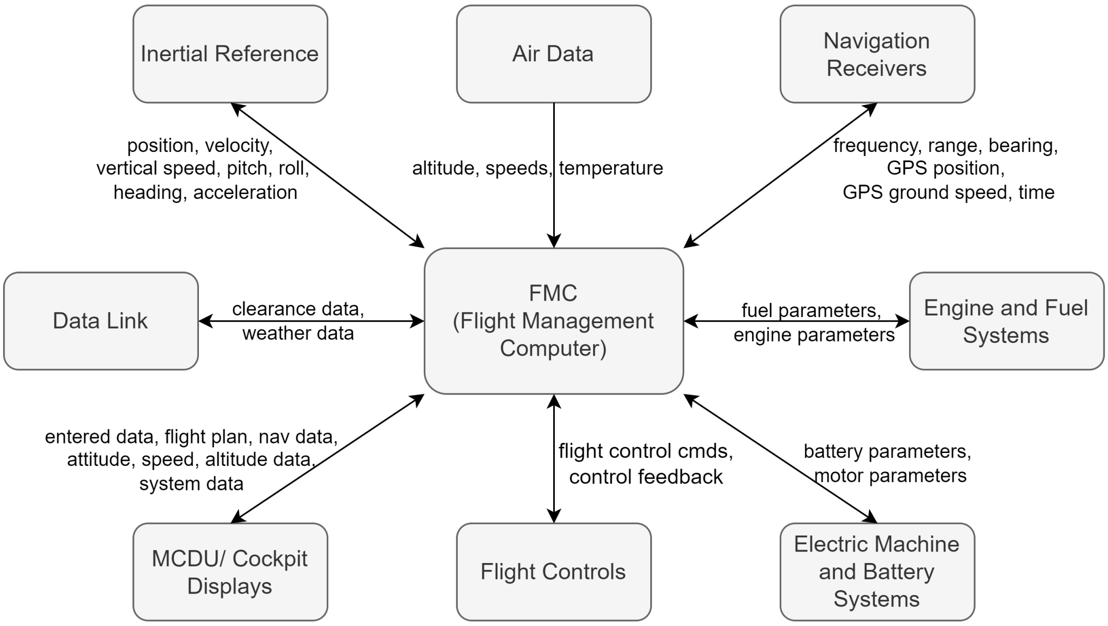


Figure 3.2: Conceptual interface block diagram for FMS in hybrid-electric aircraft.

With the added complexities due to multiple energy sources, the FMS in a hybrid aircraft should serve more functionalities to reduce the workload of the pilots. The following key functions related to energy management becomes more prominent in the FMS in the context of the hybrid-electric aircraft.

1. Estimation of Available Energy Onboard Aircraft:
   Available energy levels and the resulting range and endurance information are crucial for safe operation of a flight. Estimating these parameters with multiple energy sources is a challenging task as these parameters vary with power demand, modes of operation, weather parameters etc. The FMS should consider all these factors and provide accurate estimations for energy levels and the resulting range and endurance information to flight crew both pre-flight and in-flight.

2. Mission Planning and Optimal Flight Strategy:
   The FMS should plan the flight mission, considering both conventional and electric energy sources and it should determine the optimal operation mode schedule for optimal energy utilization of the aircraft considering factors such as available energy levels, weather conditions etc. It should also ensure the aircraft systems operate within their limits throughout the flight without depleting its energy reserves.

3. Target-Progress Performance Calculation:
   The FMS should compute the flight performance related information, including the takeoff speeds, distances, climb rates, cruise speeds, power levels and other relevant performance parameters. There could be changes to the original flight plans due to changed weather conditions, in-flight diversions, air traffic

control(ATC) requests and similar. Nevertheless, the FMS should perform the necessarily corrections and re-computations during the flight to accurately estimate the remaining energy onboard and the resulting range and endurance information. Furthermore, FMS may suggest optimal airspeed, altitude which optimize the overall energy consumption or to achieve specific performance goals such as minimum-time en-route, minimum fuel consumption or minimum battery energy consumption.

The combination of these functions along with many others, constitutes the energy management function of the FMS in hybrid aircraft. Notably, estimating the State of Energy (SOE) for a flight mission both pre-flight and in-flight, holds paramount importance within the energy management function of FMS in hybrid aircraft.

3.2 Aircraft Mission Profile

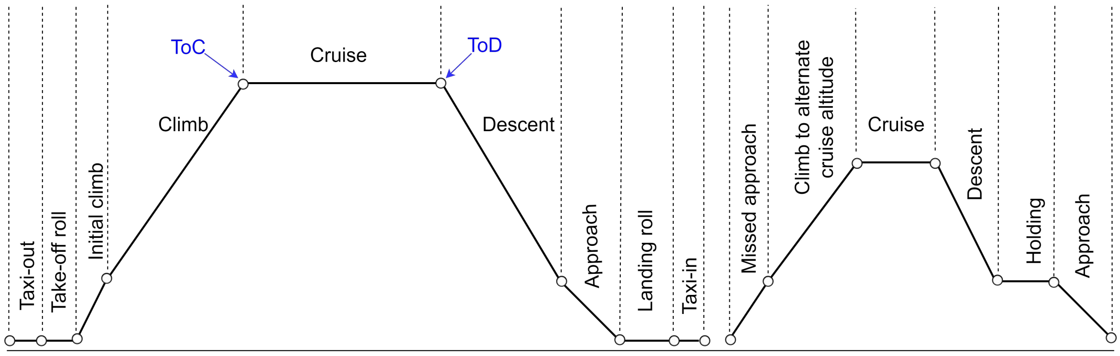


Figure 3.3: Aircraft mission profile

Figure(3.3) depicts the main flight segments for a typical aircraft mission.

Taxi-out: The taxi-out phase refers to the segment where the aircraft moves from the gate or parking area to the runway for takeoff.

Take-off roll: Take-off roll is the phase where the aircraft gradually accelerates to the lift-off and climb-out speeds along the runway. The pilot smoothly advances the throttle to takeoff position and maintains the directional control with rudder pedals while accelerating. Normally the takeoffs are made nearly into the wind for two reasons. First, the aircraft already have some of true airspeed in headwind even before start moving hence the lift-off is quick. Second, the headwind results lower ground speed hence the aircraft uses shorter runway length before the lift-off and also in case of a rejected take-off, it is more easy for the aircraft to come to a stop in headwind.

Initial climb: The initial ascent after take-off until circuit height of 1500ft AGL. Typically after lift-off, the aircraft is accelerated to V_Y, the speed resulting a the best rate-of-climb (RoC). This is the speed at which the aircraft gains the most altitude in the shortest period of time. In case of obstacle clearance after take-off, V_X also known as best angle-of-climb speed is maintained until the obstacle is cleared. V_X is the speed which results the maximum altitude gain in the minimum longitudinal distance. Typically aircraft configuration changes such as landing gear retraction, flap retraction from takeoff flap setting and reducing the aircraft throttle from takeoff to climb setting are performed in this phase to prepare the aircraft for the climb.

Climb: In the climb segment, the aircraft gradually ascents to the cruising altitude. Depending on the ATC, climb is normally performed at an shallow angle of climb with an higher airspeed, which is comfortable for the passengers onboard and also economical for the aircraft. The top-of-climb (ToC) is the point where the climb segment finishes and the cruise segment starts. At this point the aircraft attitude is changed from climb to cruise flight and acceleration to the cruise speed is also typical in flight missions.

Cruise: The level flight at the designated cruising altitude during the main portion of the journey. The cruising is done at an altitude which is most efficient for the aircraft operation. Many factors are considered to decide the cruise altitude such as wind direction, ATC requests, fuel efficiency at altitude and trip distance etc. The last point where the cruise segment finishes and the descent segment starts is called the top-of-descent (ToD). At this point the aircraft typically decelerates to the descent speed and prepares for the descent.

Descent: The gradual descent from cruise altitude to circuit height typically 1500ft AGL. This segment typically requires less power as the aircraft converts its potential energy into kinetic energy during the descent.

Approach: The descent to the destination airport, aligning with the runway for landing. Typically, the aircraft configuration changes such as landing gear deployment, changing flaps to landing setting and decelerating the aircraft to its approach and landing speed occurs within this segment. Due to the extra drag force created by the configuration changes may require higher power to maintain flight.

Landing roll: This is the period the aircraft decelerates to a stop or exit the runway after touchdown on the runway. Typically aircraft use brakes with supplemented by high drag surfaces such as spoilers to slowdown the aircraft. Aircraft with variable propeller pitch can operate the propellers in beta range to slow down the aircraft.

Taxi-in: The movement of the aircraft from the runway to the gate or parking area after landing.

Missed approach: In case of a rejected landing, the aircraft performs a go-around and climbs back up. This is a intense phase of flight where the pilots use maximum thrust or power available. The reason for this is the aircraft approaching in its approach configuration with high drag configuration should change its attitude and gain altitude as well as airspeed as quick as possible.

Climb to alternate cruise altitude: In case of diversion, the climb to an alternate cruising altitude for en-route to the alternate airport.

Holding: Due to reasons such as ATC requests, traffic congestion, poor weather or similar the aircraft delays from proceeding on course and keeps flying in a holding pattern.

The Typical Mission of the ES-30 aircraft is a reference mission defined to evaluate the aircraft and system performance of the ES-30. This mission is focused for a short-haul regional flight mission and comprises of the different flight phases described above. The typical mission is regarded as the baseline mission for energy management related performance analysis.

3.3 ES-30 Propulsion System Architecture

The ES-30 propulsion system architecture comprises of a novel independent hybrid propulsion architecture and the high-level architecture is Illustrated in Figure 3.4.

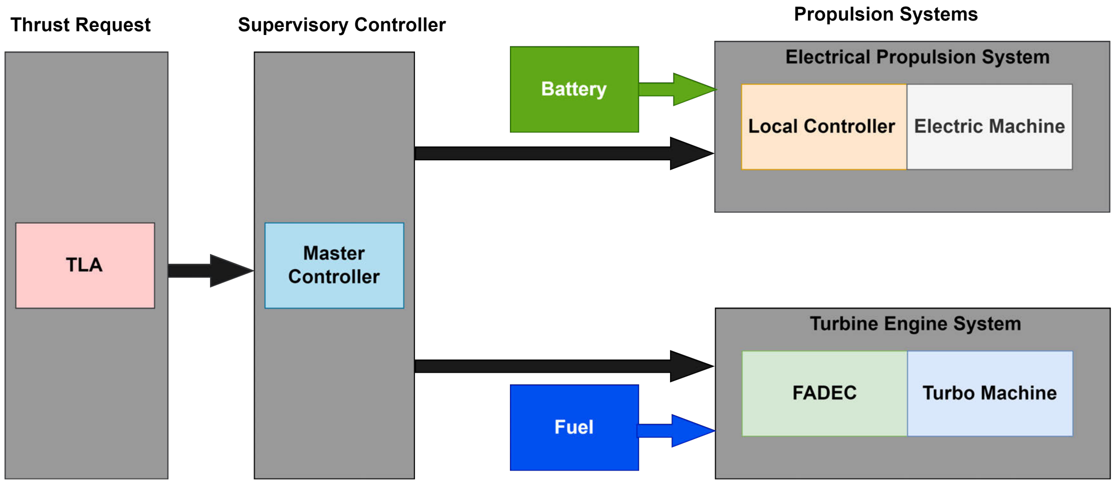


Figure 3.4: ES-30 independent hybrid propulsion architecture.

Throttle Lever Angle (TLA):
TLA refers to command for the propulsion system from the throttle quadrant, Auto Flight Control (AFC) or the FMS. This command corresponds to the aircraft-level requested thrust from the aircraft propulsion system and the TLA is an input for the master controller.

Master Controller:
The master controller serves as the supervisory controller of the ES-30 propulsion system. This system has various functions including supervisory control of the different propulsion units in the aircraft. Based on the TLA request, different mode of operations and SoE of the aircraft, master controller is responsible for commanding the local controllers of the electrical propulsion system and turbine engine propulsion system to ensure the requested thrust or power is consistently available throughout the entire aircraft mission. Additionally, master controller plays an integral role in energy management, working in conjunction with the FMS.

Electrical Propulsion System:
The Electrical Propulsion System of the ES-30 encompasses a comprehensive suite of components necessary for electrical propulsion. These include electric motors, propellers, inverters and power electronics, and Electrical Propulsion Actuation and Control System. The Electrical Propulsion System relies on electric motors to drive the propellers, providing the thrust for the aircraft while the required electrical energy is supplied from the Propulsion Battery. The **Local controller** serves as the local controller for the Electrical Propulsion System, responsible for the simultaneous coordination of motors and propellers to ensure the delivery of requested thrust by the master controller. It sends motor speed commands to the motor controllers and adjusts the propeller pitch through the propeller controllers. They work as closed-loop controllers, continuously monitoring sensor readings related to motor speed, torque and propeller pitch parameters and these parameters are strategically optimized to maximize operational efficiency throughout each phase of flight.

Turbine Engine Propulsion System (TES):
The TES consists of the turbine engines, propeller systems and the Full Authority Digital Engine Control (FADEC). This system is managed by FADEC to ensure optimal performance and efficiency throughout the aircraft operation. **FADEC** is an advanced control system that manages all aspects of turbine engine entirely through electronic means. This system is typically equipped with dual redundancy for enhanced reliability; it can operate with two identical channels to maintain full functionality even if one fails, or it can operate on a single channel backed up by a simplified electronic or hydro-mechanical system for continued operation under alternative modes [21].

3.4 Forward and Inverse Simulation Approaches

The forward simulation approach, also known as dynamic simulation for an aircraft, relies on a mathematical representation of the problem. In this approach, the various aircraft systems are modeled using sets of ordinary differential equations. These equations capture the underlying physics and dynamics of the aircraft. The kinematics of the aircraft, which encompass flight-path parameters like speed, acceleration, and the flight-path angle are the result of the balance of aerodynamic, gravitational, and propulsive forces [12]. A block diagram illustrating the workflow of the dynamic simulation approach for an aircraft is shown in Figure(3.5).

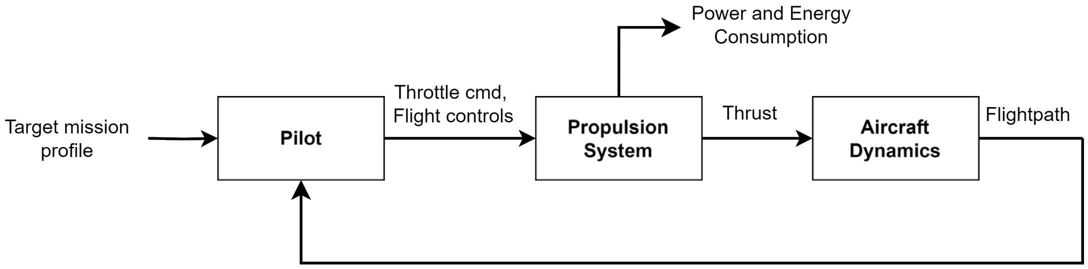


Figure 3.5: Simulation workflow in forward simulation approach for an aircraft.

Essentially, this approach determines the flight profile of the aircraft based on inputs such as pilot commands, the performance capabilities of the propulsion system, and the external forces acting on the aircraft. The forward simulation approach is widely used for analysing the control and stability aspects of the aircraft. While the forward simulation approach provides valuable insights into the dynamic behavior of the aircraft, it does come with certain limitations. High complexity and significant computational costs are drawbacks associated with this method.

The inverse simulation approach also known as the backward simulation is based on the concept of quasi-static process. In the quasi-static process, the system of interest is considered to be driven very slowly such that any transient dynamics are not engaged. For an aircraft, the inverse simulation is based on the mission profile of the aircraft, which encompasses the aircraft's kinematics defined by the flight-path variables. The total simulation time is considered to be a combination of sequence of very short time intervals. Within each time interval, the flight-path parameters of the aircraft are considered to remain constant. The thrust required for the aircraft is computed using the flight-path parameters, aerodynamic and force balance. Subsequently, the power requirements and energy consumption are determined. The simulation workflow for inverse simulation approach is shown in Figure(3.6).

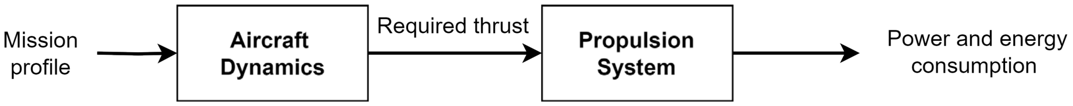
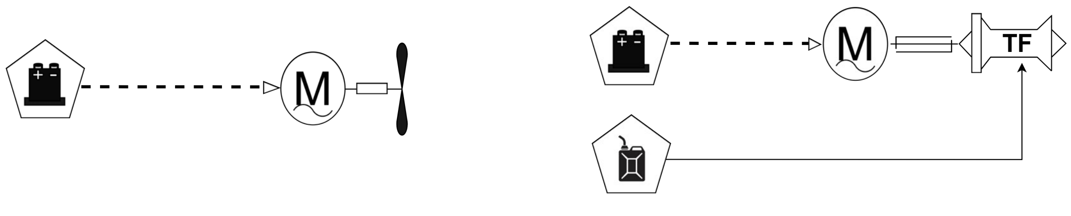


Figure 3.6: Simulation workflow in inverse simulation approach for an aircraft.

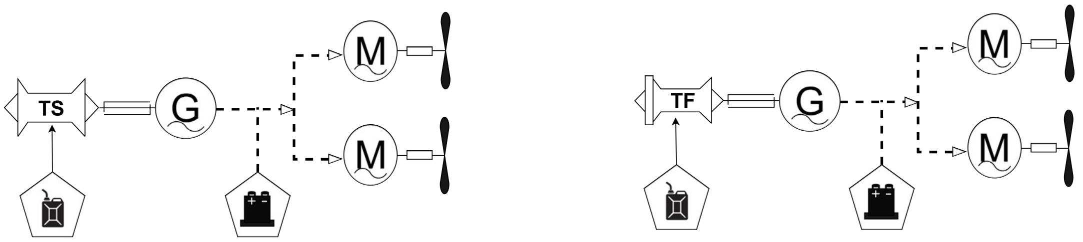
The mission profile has to be predefined for the inverse simulation approach to work effectively. This is quite common and suitable for aircraft mission simulations as the flight missions are known in advance for most of the commercial aircraft missions. These characteristics of the inverse simulation combined with the updated aircraft weight considered in each time-step, makes it suitable for analysis of energy consumption in aircraft missions.

3.5 Hybrid-electric Propulsion Architectures

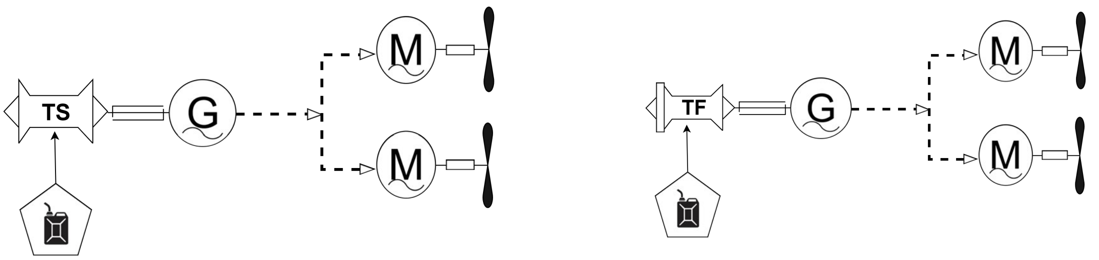
The integration of electric and fuel-based propulsion systems has led to the development of various hybrid-electric propulsion architectures in aviation. These configurations leverage different electric technologies, including batteries, motors, and generators. Six key hybrid-electric propulsion architectures identified in the literature are illustrated in Figure(3.7) [22].



(a) All Electric **(b)** Parallel Hybrid



(c) Series Hybrid **(d)** Series/Parallel Partial Hybrid



(e) Fully Turbo-electric **(f)** Partially Turbo-electric

Figure 3.7: Aircraft Hybrid-electric Propulsion Architectures.

The all-electric propulsion architecture employs batteries as the only source of propulsion power for the aircraft as shown in Figure (3.7a), exclusively relying on them to drive electric motors and generate thrust throughout every phase of flight. Primarily suited for smaller aircraft like general aviation and commuter planes, all-electric systems offer a cleaner, quieter, and more environmentally friendly alternative to conventional propulsion methods. In a parallel hybrid system, a combination of battery-powered electric motors and turbofan(TF) engines work in tandem to propel the aircraft. Both the electric motors and the turbofan engine are mounted on a shared shaft that drives the fan,as shown in Figure (3.7b). This configuration

allows for seamless transition between the two power sources, with either or both providing propulsion as needed during different flight conditions.
Series hybrid propulsion systems distinguish themselves by mechanically separating the turbo-shaft(TS) engine from the propulsive fans. Instead of directly driving the fans, the turbine engine powers an electrical generator, which then supplies electricity to the onboard motors, Figure (3.7c). These electric motors, in turn, drive the fans and/or recharge the batteries. Series hybrid configurations enable distributed propulsion concepts, utilizing multiple small motors and fans for enhanced efficiency and maneuverability. The series/parallel partial hybrid system combines elements of both series and parallel configurations. It features fans that can be directly driven by a turbofan engine, as well as additional fans exclusively powered by electric motors, Figure (3.7d). These motors can draw energy from either a battery or a turbine-driven generator. This hybrid approach maximizes propulsion efficiency by leveraging the benefits of both electric and gas turbine power sources.
Fully turbo-electric systems rely entirely on turbo-shaft driven generators to power electric motors, without any battery storage as shown in Figure (3.7e). In this configuration, the combustion engine directly powers an electrical generator, which then provides electrical power to electric motors responsible for driving the fan. This architecture completely eliminates the need for traditional mechanical transmission systems found in conventional aircraft engines. On the other hand, partial turbo-electric systems combine elements of fully turbo-electric architecture with traditional turbofan engines as shown in Figure (3.7f). In this configuration, some power from the combustion engines is used to directly drive the fans, similar to the parallel hybrid architecture. The remaining power is generated by gas turbine-driven generators and supplied to electric motors to assist in fan propulsion.

3.6 Ambient Parameters

The performance of an aircraft is significantly influenced by atmospheric properties. These properties include ambient pressure, temperature, air density, and the speed of sound. As an aircraft ascends or descends, these primary atmospheric properties change. Additionally, they play a crucial role in calculating other parameters like pressure altitude and density altitude, which are essential for analyzing aircraft performance. Modeling these parameters becomes essential when evaluating how different ambient conditions impact energy consumption in an aircraft.

3.6.1 Standard Atmosphere

The performance of aircraft and propulsion systems depends on several atmospheric properties. These properties are never constant at any particular time or place. Therefore, an idealized steady-state representation of the earth's atmosphere, referred as the International Standard Atmosphere (ISA) have been introduced by the International Civil Aviation Organization (ICAO) [23].

The ISA model divides the atmosphere into several layers based on assumed linear distribution of absolute temperature with altitude. The Mean Sea Level (MSL) is considered to be the zero altitude. The first layer extending from the MSL into 11000 m is the Troposphere. The subsequent layer, extending from 11000 m to 20000 m is the Stratosphere. The Troposphere and Stratosphere are separated by the imaginary boundary known as the Tropopause. Since the operational ceiling of the ES-30 is well below the Tropopause, the conditions of the Troposphere is only considered in this study. According to the ISA model, the following conditions presented in Table(3.1) are assumed to be held in ISA condition.

Table 3.1: ISA atmospheric conditions.

Parameter	**Symbol**	**Value**
ISA temperature at MSL	$(T_0)_{\mathrm{ISA}}$	288.15 K
ISA air density at MSL	$(\rho_0)_{\mathrm{ISA}}$	1.225 $\mathrm{kg\,m^{-3}}$
ISA pressure at MSL	$(P_0)_{\mathrm{ISA}}$	101325 $\mathrm{N\,m^{-2}}$
ISA speed of sound at MSL	$(a_0)_{\mathrm{ISA}}$	340.294 $\mathrm{m\,s^{-1}}$
Temperature lapse-rate in Troposphere	λ	-0.0065 $\mathrm{K\,m^{-1}}$

Other important parameters considered in computing the atmospheric properties are,
Gravitational acceleration g = 9.80665 $\mathrm{m\,s^{-2}}$
Real gas constant for air R = 287.04 $\mathrm{m^2\,K^{-1}}s^2$

As the real atmospheric conditions can deviate from the ISA conditions, the deviation from ISA can be considered to perform more realistic computations for atmospheric properties. The main two parameters to indicate the deviation from ISA conditions are temperature deviation (ΔT_{ISA}), often called as Delta-ISA (DISA) and the pressure deviation (ΔP_{ISA}).

The temperature at MSL T_0, under the atmospheric condition with the temperature DISA, ΔT_{ISA} is given by,

$$T_0 = (T_0)_{\mathrm{ISA}} + \Delta T_{\mathrm{ISA}} \tag{3.1}$$

Similarly, the atmospheric pressure at MSL P_0 with the pressure deviation from ISA ΔP_{ISA} is,

$$P_0 = (P_0)_{\mathrm{ISA}} + \Delta P_{\mathrm{ISA}} \tag{3.2}$$

3.6.2 Temperature at Altitude

The Outside Air Temperature (OAT) is an important parameter for aircraft performance. The thrust force generated by the propulsion system is decreased, with the increasing OAT as the air density reduces with increasing temperature. The true air speed (TAS) of the aircraft must be increased to compensate for the reduction in air density and hence the take-off distance is increased. Another effect of increased OAT is the reduced rate of climb of the aircraft due to the reduction of the excess thrust of the aircraft.

The temperature also has an effect on the indicated altitude in altimeter in the cockpit instruments. As a result of that, if the temperature is higher at a constant indicated altitude, the true altitude that the aircraft is flying is higher. Therefore, altitude temperature correction tables are used to correct the altitude deviations due to temperature [24].

The temperature at an altitude can be performed considering the temperature at MSL, and the temperature gradient in the atmosphere.
T_h, the temperature at an altitude of h is given by,

$$T_h = T_0 + \lambda h \tag{3.3}$$

where, T_0 is the temperature at MSL and λ is the temperature gradient.

3.6.3 Ambient Pressure at Altitude

The ambient pressure also known as static pressure at altitude depends on the variation of pressure and temperature. The static pressure P_h at altitude h is given by,

$$P_h = P_0 + \left[1 + \frac{\lambda}{T_0} h\right]^{-\frac{g}{\lambda R}} \tag{3.4}$$

where, T_0 and P_0 are the temperature and pressure at MSL respectively.,

3.6.4 Air Density

Air density is an important atmospheric property which affects the aircraft performance in many ways, most notably aircraft engine systems, aerodynamic performance of air-frame and propellers. The air density is influenced by atmospheric pressure, temperature and humidity. Considering the dry-air assumption, the air density ρ_h at an altitude h is given by,

$$\rho_h = \rho_0 \left[\frac{T_h}{T_0}\right]^{-\frac{g}{\lambda R} - 1} \tag{3.5}$$

here, ρ_0 is the air density at MSL:

$$\rho_0 = (\rho_0)_{\mathrm{ISA}} \cdot \frac{(T_0)_{\mathrm{ISA}}}{T_0} \tag{3.6}$$

3.6.5 Speed of Sound

The speed of sound vary with ambient temperature. The speed of sound at an altitude below Tropopause can be computed as a function of ambient temperature T_h by,

$$a_h = (a_0)_{\mathrm{ISA}} \cdot \sqrt{\frac{T_h}{(T_0)_{\mathrm{ISA}}}} \tag{3.7}$$

3.6.6 Mach Number

The Mach Number is a ratio between the aircraft TAS and the speed of sound. The Mach Number M for an aircraft flying at V_{TAS} speed and altitude h is given by,

$$M = \frac{V_{\mathrm{TAS}}}{a_h} \tag{3.8}$$

3.6.7 Pressure Altitude

The Pressure Altitude (PA) is the theoretical altitude of an aircraft with respect to the standard datum defined by the ISA conditions [47]. Pressure altitude is the altitude in the ISA model that has the same atmospheric pressure as at the altitude of interest. Pressure altitude is important and widely used as a basis for defining aircraft performance and also important when assigning flight levels to aircraft operating at or above 18000 ft.
Pressure altitude affects the aircraft and engine performance. When the pressure altitude is increased the air density is reduced and the TAS of the aircraft has to be increased to compensate the reduction in air density

As the increased pressure altitude reduces the available thrust and it results longer takeoff distances and reduced climb performance[24].
The pressure altitude (PA) for a given altitude h can be computed considering the inverse of pressure lapse-rate Γ_P and atmospheric pressure at sea level at the interested location P_0 which is often called as QNH in aviation.

$$PA = h + \Gamma_P \left[(P_0)_{\mathrm{ISA}} - P_0 \right] \tag{3.9}$$

The $\Gamma_P = 27\,\mathrm{ft}\,/\,\mathrm{h\,Pa}$ in the lower part of the atmosphere while it can take higher values that that in the higher parts of the atmosphere. As a rule-of-thumb, $\Gamma_P = 30\ \mathrm{ft}\,/\,\mathrm{h\,Pa}$ value is commonly used by the aviation community when computing PA.

3.6.8 Density Altitude

The Density Altitude (DA) is the pressure altitude corrected for non-standard temperature and it represents the combined effect of pressure altitude and the ambient temperature. Density altitude is also used in calculating aircraft performance.
Higher density altitudes results in reduced engine and propeller performance, and it also results in longer take-off distances as well as reduced climb performance [25].
The density altitude can be computed from pressure altitude, ambient temperature at pressure altitude T_{PA} and inverse of temperature lapse-rate Γ_T as

$$DA = PA + \Gamma_T \left[T_h - (T_{PA})_{ISA} \right] \tag{3.10}$$

Here $(T_{PA})_{\mathrm{ISA}}$ is the ISA temperature at the pressure altitude.

3.6.9 True Air Speed and Calibrated Air Speed

The True Air Speed (TAS) is the speed of an aircraft relative to the air it is flying through. The Indicated Air Speed (IAS) is the speed indicated in cockpit indicators for pilots. The Calibrated Air Speed (CAS) is the IAS corrected for sensor and instrumentation errors. The TAS differs to CAS due to the variation of air density at altitudes and atmospheric conditions. The conversion from CAS $\mathrm{m\,s^{-1}}$ to TAS $\mathrm{m\,s^{-1}}$ can be performed as,

$$V_{\mathrm{TAS}} = \left\{ \frac{2}{\mu} \frac{P_h}{\rho_h} \left[\left(1 + \frac{(P_0)_{\mathrm{ISA}}}{P_h} \left(\left(1 + \frac{\mu}{2} \frac{(\rho_0)_{\mathrm{ISA}}}{(P_0)_{\mathrm{ISA}}} V_{\mathrm{CAS}}^2 \right)^{\frac{1}{\mu}} - 1 \right) \right)^{\mu} - 1 \right] \right\}^{\frac{1}{2}} \tag{3.11}$$

where,
P_h and ρ_h are ambient pressure and air density at aircraft altitude respectively.
$\mu = \frac{\mathcal{K}-1}{\mathcal{K}}$ with $\mathcal{K} = 1.4$ (Adiabatic index of air).

3.7 Aerodynamic Forces

An airfoil for a typical aircraft wing and the corresponding parameters are illustrated in Figure(3.8).

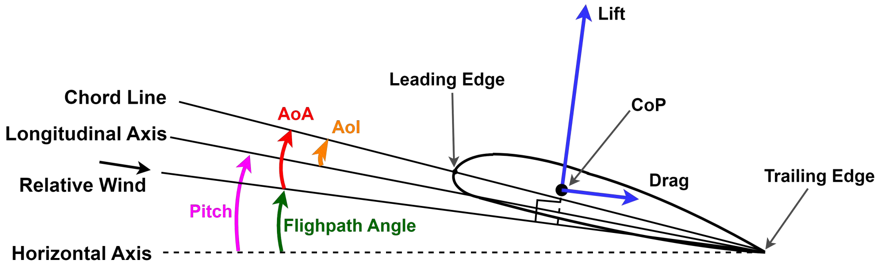


Figure 3.8: Lift and drag forces and related angles for an typical aircraft wing airfoil.

Chord line: Chord line is the imaginary line joining the leading and trailing edges of the airfoil.
AoA: Angle of Attack (AoA) is the angle between the chord line and the relative wind.
AoI: Angle of Incidence (AoI) is the angle between the chord line and the longitudinal axis.
CoP: Center of Pressure (CoP) is considered to be the focal point on which the lift force is acting upon. The position of the CoP vary with the change of AoA.
Pitch: Pitch angle is the angle between the horizontal axis and the longitudinal axis.
Flightpath angle: The angle between horizontal axis and the flightpath vector or the relative wind.

When the aircraft moves forward through the air, the airfoil shape of its wing results low pressure over the wing and higher pressure underneath. This pressure difference creates the lift force according to the Bernoulli's principle. The lift force is always perpendicular to the flightpath vector by definition. The aerodynamic force acting in the opposite direction of the movement of the aircraft is called the induced drag force.

The lift force (L) depends on many factors including aircraft true air speed, air density, reference wing area and the shape of the airfoil and the angle of attack. Further the compressibility and the viscosity of the air also affects the lift force. The various complex dependencies of the shape, AoA and flow conditions on the lift force are lumped together into lift coefficient (C_L) and it is usually determined experimentally.
The lift equation to compute lift force L is,

$$L = \frac{1}{2}\, C_L\, \rho\, v^2\, S_{ref} \tag{3.12}$$

where,
ρ Air density
v Aircraft TAS
S_{ref} Reference wing area

The term $\frac{1}{2}\rho v^2$ in Equation(3.12) is called the dynamic pressure and denoted by q or Q.
The dynamic pressure q can be incorporated into the Equation(3.12) and can be expressed as,

$$L = C_L\, q\, S_{ref} \tag{3.13}$$

The drag force is the net force acting opposing the movement of the aircraft. This is parallel and opposite to the flightpath vector. The drag force consists of two types of drag forces. The first of which is the parasite drag and it can further be splitted to skin friction, interference and form drag.
The form drag is the resistance force acted upon the structure of the aircraft when it moves through air as a result of the roughness and shape of the air-frame and structures. The skin drag occurs as a result of the boundary layer which is stationary relative to the aircraft skin surface. Interference drag occurs as a result of the aerodynamic interference of the different structures of the aircraft for instance the wing struts and extended landing gears. Parasite drag can be simply explained as the total drag independent of the lift force.
The induced drag is the drag occurred as a result of creating the lift force. The behaviour of parasite and induced drag is different with respect to air speed. While induced drag reduces when airspeed is increased, the parasite drag increases. The total drag is the sum of the induced and parasite drag values. Figure(3.9) illustrates the behaviour of induced, parasite and total drag with aircraft TAS which is often called as the aircraft drag curve.

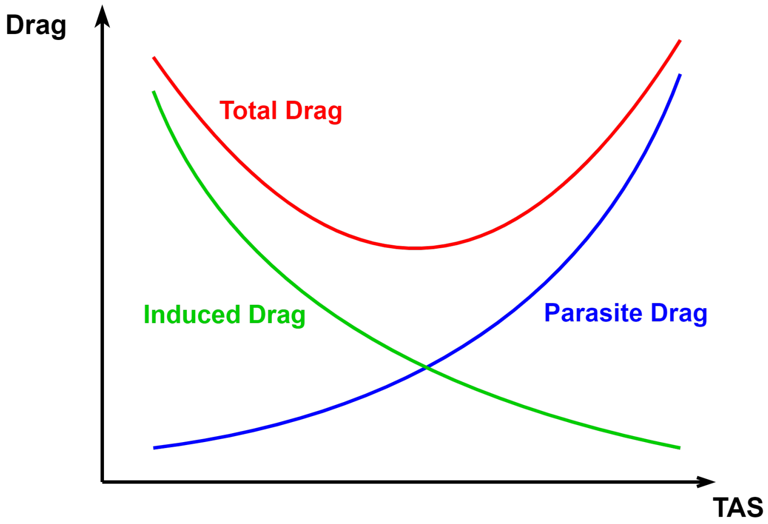


Figure 3.9: Illustration of aircraft drag curve.

The drag coefficient C_D can be expressed as a parabolic function of C_L. Here the C_D is divided into its main components as,

$$C_D = C_{D_0} + KC_L^2 + \Delta C_{D_{Flap}} + \Delta C_{D_{LG}} \tag{3.14}$$

where,
C_{D_0} Zero-lift drag coefficient
K Induced-drag factor
C_L Lift coefficient
$\Delta C_{D_{Flap}}$ Drag coefficient component due to flap extension
$\Delta C_{D_{LG}}$ Drag coefficient component due to landing gear extension

The K value is given by,

$$K = \frac{1}{\pi\, AR\, \phi} \tag{3.15}$$

where, ϕ is the Oswald efficiency factor and the Aspect Ratio AR is given by,

$$AR = \frac{\text{Wing-span}^2}{S_{ref}} \tag{3.16}$$

The aerodynamic drag force D can be determined using the drag coefficient C_D by,

$$D = \frac{1}{2} C_D\, \rho\, v^2\, S_{ref} = C_D\, q\, S_{ref} \tag{3.17}$$

3.8 Aircraft Point-mass Model

The aircraft has 3 principle axes as illustrated in Figure(3.10) and they are,

- Longitudinal Axis: Imaginary axis drawn through the body of the aircraft from tail to nose through the aircraft center-of-gravity (CoG) in the normal direction of flight.

- Lateral Axis: Imaginary line drawn through the aircraft CoG, parallel to the line drawn from left wing-tip to the right wing-tip and perpendicular to the longitudinal axis.

- Vertical Axis: Imaginary line drawn through the aircraft CoG and perpendicular to both longitudinal and lateral axes.

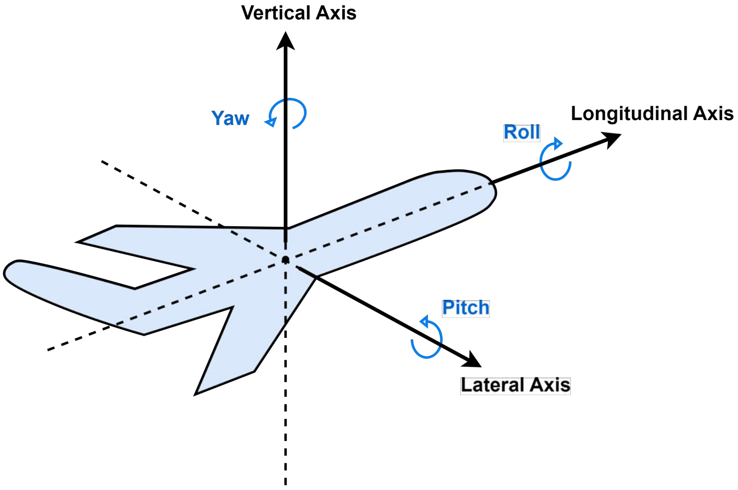


Figure 3.10: Aircraft principle axes and roll, pitch yaw rotations

The aircraft has six Degrees-of-Freedom (DoF) with respect to the three principle axes. They are the transnational motions along longitudinal, lateral and vertical axes and the rotational motions around those three axes which are roll, pitch and yaw respectively.

The motion of the aircraft and the forces acting on the aircraft is important in studies related to energy consumption. There are mainly two approaches in describing the aircraft motion. The Total Energy Model (TEM) approach [26] [27] relates the rate of work done by the forces acting on the aircraft to the changes in kinetic and potential energy.

The other approach is developing aircraft model in in terms of the so called point-mass model. In this approach, all the forces acting on the aircraft are considered to be applied on the aircraft CoG and the force balance equations are used to develop the relation between the flightpath variables and the forces [28]. Different versions of these models can be developed based on the interested parameters and the desired level of fidelity.

Reduced number of DoFs can be considered to develop a simplified point-mass model of the aircraft. Common approach used for aircraft performance models is modeling aircraft as a 3 DoF point-mass model considering the longitudinal movement and the flightpath angle of the aircraft. A free body diagram depicting the forces acting on the aircraft and the relevant other parameters is illustrated in Figure(3.11).

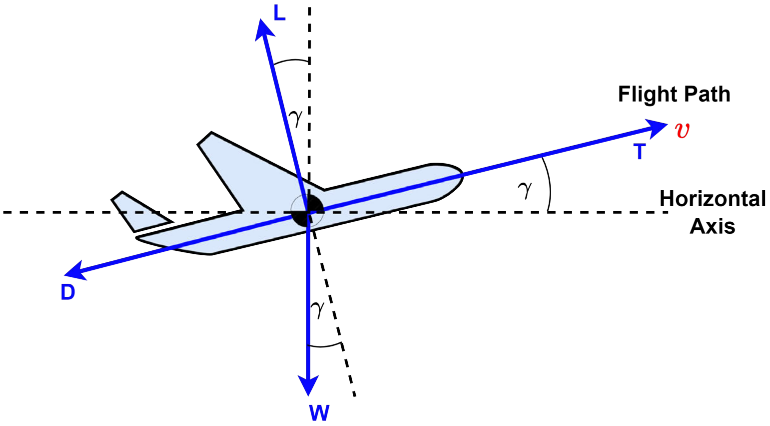


Figure 3.11: Aircraft free-body diagram with flightpath variables and forces.

γ Flightpath angle
$\dot{\gamma}$ Rate of change of flightpath angle
v Aircraft TAS
$\dot{v}$ Rate of change of TAS
L Lift force
W Aircraft weight ($W = m\,g$)
D Aircraft drag
T Aircraft thrust

Considering the free-body diagram of the point-mass model of the aircraft, the following equations can be derived.

For the force balance along the aircraft flightpath:

$$T - D - W\,sin(\gamma) = m\,\dot{v} \tag{3.18}$$

For the force balance perpendicular to the the aircraft flightpath:

$$L - W\,cos(\gamma) = m\,v\,\dot{\gamma} \tag{3.19}$$

The $mv\dot{\gamma}$ component in equation(3.19) is the centrifugal force created when changing the flightpath angle of the aircraft (pitch change around the lateral axis) and this force is perpendicular to the flightpath vector.

3.9 Propellers

The purpose of the propeller is to convert the engine shaft power into linear thrust force which propels the aircraft forward. This should be done as efficiently as possible in the range of the aircraft speeds and ambient conditions. The thrust from the propeller is generated by accelerating a mass of air from a lower velocity to a higher velocity. The propeller blade can be considered as an airfoil moving through the air, and the lift force created by this airfoil contributes to the thrust force generated by the propeller. The propeller blade related important angles and parameters are illustrated in Figure(3.12).

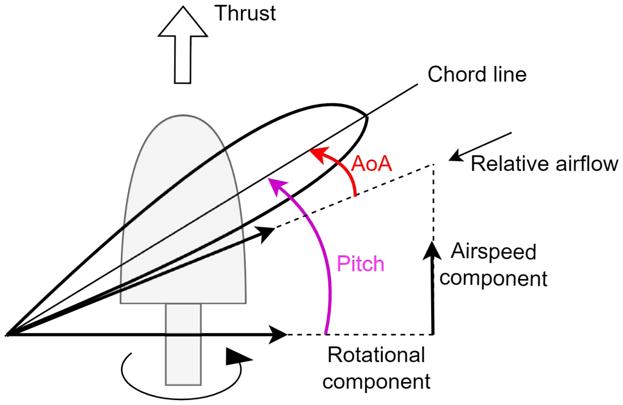


Figure 3.12: Angles and parameters related to the aerodynamics of a propeller blade.

Chord line: The chord line of a propeller blade is the imaginary line joining its leading and trailing edges.
Relative airflow: While the propeller rotates around the propeller hub, the aircraft is also moving forward. Therefore, the relative airflow is the combined effect of both the forward speed of the aircraft and the rotational motion of the propeller.
Rotational component: This speed component is a result of the circular motion of the propeller blade and it is proportional to the propeller angular speed n and propeller diameter D.
Airspeed component: This speed component is equal to the TAS of the aircraft.
Pitch angle is the angle between the chord line and rotational component or the rotational plane of the propeller.
Angle of attack: The AoA is the angle between the chord line and the relative airflow of the propeller.

The main two categories of aircraft propellers are the fixed pitch propellers and the variable-pitch propellers. As the name implies, the fixed-pitch propellers are manufactured to a fixed pitch angle. However, the propeller efficiency highly depends on the aircraft TAS, and the fixed pitch propellers are typically designed for optimal efficiency at cruise setting. However the fixed pitch propellers cannot operate efficiently in the other flight segments such as takeoff, climb and descent.

The variable pitch propellers can change their pitch angle and operate with optimal efficiency at a wide range of aircraft conditions. The propeller pitch is controlled with different mechanisms such as hydro-mechanical governors or electronic control. The pilots can control the propeller pitch in different settings such as Fine, Coarse, Feather or Reverse Pitch depending on the phase or the condition of the flight.

Propellers play an extremely important part of a propeller based aircraft propulsion system. Capturing the characteristics of the propellers are an important task when modeling the propulsion system with propellers. There are mainly three approaches in modeling propellers in the literature. They are,

1. Constant efficiency
2. Constant pitch
3. Propeller performance maps

In constant efficiency based propeller modeling approach, constant values are assigned to propeller propulsive efficiency. Typically, different efficiency values are used during different phases of the mission such as takeoff, climb, cruise, descent and landing.

In the constant pitch based propeller modeling approach, the propellers are considered to be fixed-pitch propellers and the relation of propellers rotational speed to the required power P_{req} and altitude h is mathematically expressed according to Equation(3.20) [29].

$$\omega_{prop} = \omega_{prop,nom} \sqrt[3]{\frac{P_{req}\,\rho_h}{P_{prop,nom}\,\rho_0}} \tag{3.20}$$

where,

$P_{\text{prop,nom}}$ Nominal power
$\omega_{\text{prop,nom}}$ Nominal rotational speed
ρ_0 Air density at MSL
ρ_h Air density at altitude h

The propeller efficiency is calculated with the blade element theory considering the specifications of the propeller.

Propeller modeling using the propeller performance maps are typically used for constant speed propellers. The propellers are characterized by means of propeller maps which contains propeller performance information for a wide range of operation range. The propeller maps are expressed in terms of the following dimensionless parameters.

Advance Ratio (J): Advance ratio is the distance the propeller advances forward per revolution in terms of its diameter.

$$J = \frac{v}{n\,D} \tag{3.21}$$

where n is the propeller rotational speed in revolutions per second, v is aircraft TAS and D is the propeller diameter.

Propeller Tip Mach (M_{Tip}): Propeller tip mach is the ratio between the propeller tip speed and the speed of sound. The propeller tip speed is the resultant of aircraft TAS and the rotational component explained above.

$$M_{Tip} = \frac{\sqrt{v^2 + \left(n \, \frac{D}{2}\right)^2}}{a} \tag{3.22}$$

Thrust Coefficient (C_T): The thrust coefficient relates how much thrust the propeller creates for the given parameters.

$$C_T = \frac{T}{\rho \, n^2 \, D^4} \tag{3.23}$$

where T is the thrust force.

Torque Coefficient (C_Q): The torque coefficient relates how much torque is exerted on the shaft by the propeller.

$$C_Q = \frac{Q}{\rho \, n^2 \, D^5} \tag{3.24}$$

where Q is the shaft torque.

Power Coefficient (C_P): The power coefficient relates to how much power it takes to turn the propeller, which is often called as the absorbed power and this is equal to the shaft power. The power coefficient and the its relation to the torque coefficient is given by,

$$C_P = \frac{P_{Shaft}}{\rho \, n^3 \, D^5} = 2 \, \pi \, C_Q \tag{3.25}$$

where P_{Shaft} is the shaft power or absorbed power.

The propulsive power is different to the shaft power. The propulsive power generated by the rotating propeller is calculated in terms of the propeller thrust and the aircraft TAS as,

$$P_{Propulsive} = T \, v \tag{3.26}$$

Propeller Efficiency (η_{Prop}): The propeller efficiency is the ratio of propulsive power to the absorbed power. The propeller efficiency is given by,

$$\eta_{Prop} = \frac{P_{Propulsive}}{P_{Shaft}} = \frac{T \, v}{P_{Shaft}} = \frac{C_T \, J}{C_Q} \tag{3.27}$$

3.10 Permanent Magnet Synchronous Machine

The electric motors play a pivotal role in hybrid-electric propulsion systems, converting electric energy into mechanical energy through the interaction of magnetic and electric fields. Among the various motor types available, Permanent Magnet Synchronous Motors (PMSM) are widely used in aircraft applications due to their

advantages such as high efficiency, reduced maintenance, quiet operation, and high power density [30]. In the design of PMSMs for aeronautical applications, certain critical specifications must be addressed. The weight of the PMSM is directly linked to the energy consumption of the aircraft, whether in terms of thermal or electric energy. Therefore, minimizing the weight of the motor is crucial for enhancing the overall efficiency and performance of the aircraft [31].
PMSMs are typically classified into two types based on the arrangement of permanent magnets: Surface Permanent Magnet Synchronous Motors (SPMSMs) and Interior Permanent Magnet Synchronous Motors (IPMSMs), as illustrated in Figure (3.13). SPMSMs feature permanent magnets mounted entirely on the rotor's surface. On the other hand, IPMSMs incorporate magnets within the rotor structure. This internal placement of magnets enhances their versatility and performance, making them particularly suitable for aviation applications [32].

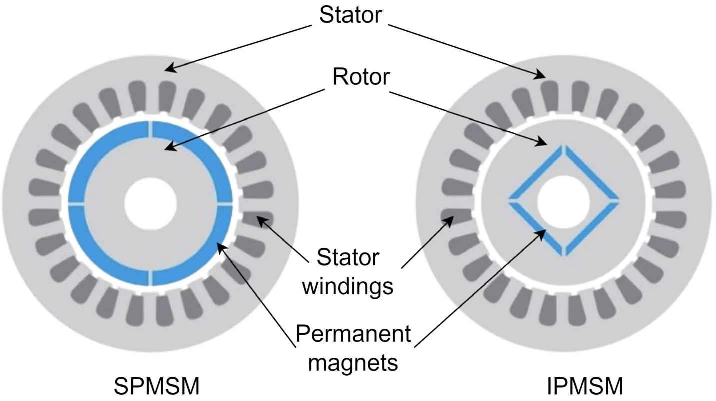


Figure 3.13: IPMSM and SPMSM architecture [32].

The motor generates shaft power, which is transmitted to the propeller shaft through either direct drive or gear drive. This facilitates electric propulsion, resulting in thrust output. The choice between direct coupling and gear-driven configurations depends on the specific requirements and design choices of the aircraft.

The common approaches for modeling motors include first-principle-based modeling and performance map-based modeling. The first-principle approach utilizes differential equations and mathematical relations related to motor dynamics, relying on internal parameters that are often proprietary information. In contrast, performance map-based modeling represents motors at a high level, using data tables and maps derived from empirical tests. Although this approach does not capture internal motor parameters, it provides more realistic results in terms of overall performance and energy-related parameters.

3.11 Turbine Engines

A turboprop is a gas turbine engine which drives a propeller connected to its output shaft. The working principle of turboprop closely follows the general working principle of a gas turbine engine. A turbine engine mainly consists of intake, compressor, combustion chamber, turbines and exhaust. The output shaft is rotated by the hot combustion gas expand through the turbine blades. Contrast to the turbojet and turbofan engines, the turboprop engines does not create significant amount of thrust by the exhaust gases, instead almost all the power generated by the engine is used to drive the connected propeller.

Compared to piston engines, the turboprop engines have greater power-to-weight ratio which allows relatively shorter take-offs and they are also fuel efficient as they can operate at higher altitudes than the piston engines. The turboprop engines can operate efficiently at subsonic speeds below 400 knots. These characteristics of turboprop engines has enable them to be used widespread in regional airline market for example aircraft such as ATR 42/62/72, Bombardier Q400, De Havilland Canada Dash 8-100, 200, 300 and Saab 340 use turboprop engines. A simplified layout of a Pratt and Whitney PT6 turboprop engine depicting the main component is illustrated in Figure(3.14).

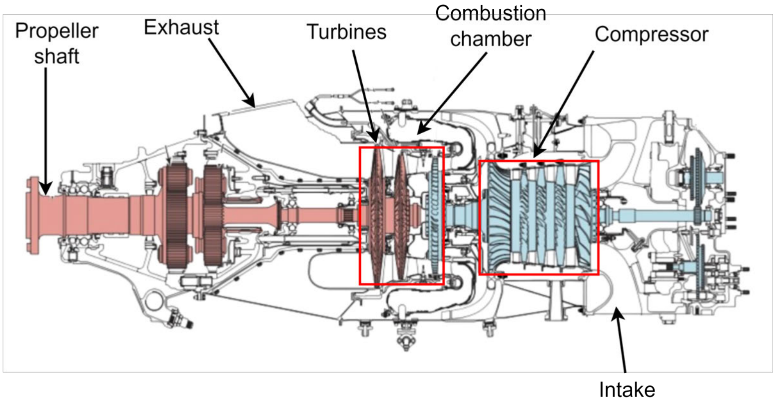


Figure 3.14: Illustration of PT6A Turboprop engine [33].

Turboprop engines can be modeled using various approaches, each differing in terms of accuracy, complexity, and the specific aspects and parameters of interest. One such approach involves thermodynamic modeling. In this method, the different components of the turboprop engine such as the intake, compressors, combustion chamber, turbine, and exhaust are modeled based on their thermodynamic properties and relations. The thermodynamic approach for turbine engines is rooted in the Brayton cycle, which ensures the conservation of energy and momentum across the engine system. Within this cycle, compression and expansion processes are considered isentropic and adiabatic, while combustion occurs at constant pressure [34]. To characterize the operation of compressors and turbines, performance maps are employed. These maps are non-linear empirical models generated from computational fluid dynamics, experimental data, or estimations. Additionally, mechanical losses such as bearing friction, gearbox losses, and thermodynamic losses are factored in to determine the engine output power. While this approach provides valuable insights into parameters like internal temperatures at different locations within the engine and overall performance, its complexity remains a disadvantage. Furthermore, obtaining the necessary input parameters for the model can be challenging, as they often constitute proprietary information held by engine manufacturers.

Another approach in modeling turbine engines is modeling the turbine engine in terms of fuel consumption maps. In this approach the engine is modelled based on the engine efficiency and the fuel consumption to deliver a specific power output. The output power of a turboprop engine corresponds to the engine shaft power and the energy needed to generate this power is delivered from the combusted fuel. Fuel consumption maps take into account the ambient parameters because, the fuel consumption and engine efficiency are significantly affected by the ambient conditions. Typically, these maps consider factors such as pressure altitude, Delta ISA (DISA), and shaft power as the breakpoints or input parameters. Although this approach is simple and provides the accurate information for energy consumption related parameters, this method fails to capture the internal parameters of the engine.

In a turbine engine that produces shaft power, fuel consumption is identified by Brake-Specific Fuel Consumption (BSFC).

$$BSFC = \frac{\dot{m}_{fuel}}{P_{Shaft}} \tag{3.28}$$

where $\dot{m}_{fuel}$ is the fuel flow-rate and P_{Shaft} is the shaft power.

When a considerable thrust is produced by the exhaust of the turboprop engine, the fuel consumption is expressed in terms of Equivalent Brake-Specific Fuel Consumption (ESFC).

$$ESFC = \frac{\dot{m}_{fuel}}{ESP} \tag{3.29}$$

where ESP is the equivalent shaft power which accounts for both the delivered shaft power and the thrust created by the exhaust gas.

These specific fuel consumption maps are based on empirical data and provide very close results to the reality. The specific fuel consumption values are often expressed in kg / kWh or (lb / h) / hp [35].

3.12 Battery

Batteries store energy in the form of chemical energy, which they then convert and release as electricity on demand. A single battery unit is often referred to as a battery cell, and these cells come in various shapes, sizes, and characteristics. The most common types of battery cells include Alkaline, Nickel Metal Hydride (NiMH), and Lithium-ion. Lithium-ion batteries, in particular, are widely used in hybrid electric propulsion systems due to their advantageous characteristics, such as lightweight construction, high specific energy density, long cycle life-span, and low self-discharge rates.

Two crucial parameters of battery cells are rated voltage and rated capacity. The voltage rating represents the nominal voltage level expected from the battery. During operation, the output voltage of a battery cell is not constant and it varies as the cell discharges. At full charge, the voltage is maximum, but it decreases gradually. There exists a relatively constant voltage region within the battery is designed to operate. Operating outside this region can lead to steep charge drops and potential damage. The capacity of a battery cell reflects the energy generated through its chemical processes and it is the integral of the current that can be delivered under certain conditions, measured in Ampere hours (Ah)[36]. Battery cells are arranged in different configurations to achieve the high capacity, voltage and operational requirements of the electrical systems.

Battery cells connected in series increase the voltage while the battery cells connected in parallel increase the A h of the battery. Connecting a number of battery cells in series and parallel forms a battery module. Connecting the battery modules together forms a battery pack. The battery packs are designed to cater the necessary capacity and voltage of particular applications. The battery pack architecture is often denoted by xSyP, where this pack has x number of battery cells in series and y number of battery cells in parallel.

The State-of-Charge (SoC) is another important parameter to indicate the charge level of a battery cell or a battery system. SoC of a battery cell is defined as the ratio of the current releasable $C(t)$ charge of a battery cell to the rated capacity C_{rated}.

$$SoC = \frac{C(t)}{C_{rated}} \tag{3.30}$$

There are different ways to model batteries and these methods differ in the level of complexity and the level of information the models are able to provide. The main methods for modeling batteries, found in the literature are [37],

1. Electro-chemical or physics based models
2. Empirical or data-based models
3. Equivalent circuit models

The chemical reactions within the battery cell is the focus of electro-chemical battery modeling approach and this makes this approach the most accurate. This approach is complex and many parameters are unknown as they are proprietary information of battery cell manufacturers.

The analytical battery modeling approach is based on more abstract representation of the battery cell in terms of a mathematical model. This approach involves less parameters and computational effort as it models the battery cell behaviour without accounting for the chemical and physical processes within the battery cell [38].

Another widely used approach for battery modelling is the Equivalent Circuit Model (ECM) approach. ECM describes the pure electrical behaviour of the battery in terms of ideal electrical components such as voltages, current, resistance, capacitors and inductors. Internal resistance model (Rint Model), 1-RC model, 2-RC model, Randles circuit model and PNGV models are widely used ECM based battery models found in the literature [39].

A basic internal resistance (Rint) ECM battery model is shown in Figure(3.15). This model uses ideal voltage source V_{OC} to simulate the open circuit voltage of the battery and R_i resistor reflects the internal resistance of the battery and the values V_{OC} and R_i are considered to be constant [40].

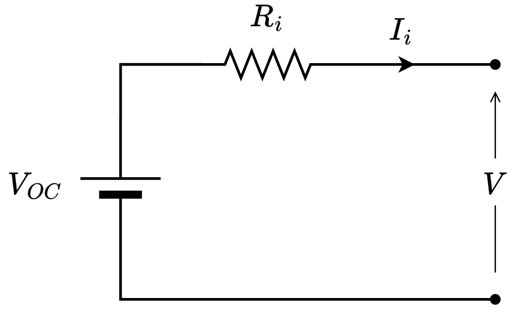


Figure 3.15: Basic ECM for Rint battery model.

Using this model and the Kirchhoff's law, the battery voltage V can be expressed as:

$$V = V_{OC} - I_i R_i \tag{3.31}$$

The power demand on the battery from the load P_{batt} can be related to the battery current I_i by,

$$I_i = \frac{V_{OC} - \sqrt{V_{OC}^2 - 4\,R_i\,P_{batt}}}{2\,R_i} \tag{3.32}$$

The SoC change of the battery when the battery is in use can be expressed as,

$$\dot{\mathrm{SoC}} = \frac{I_{batt}}{C_{rated}} \tag{3.33}$$

3.13 Dynamic Programming

Dynamic Programming (DP) was developed based on Richard Bellman's principle of optimality in the 1950s, and ever since has been used as a powerful technique for designing optimal controllers for systems with constrained states and input variables. Dynamic Programming (DP) is well-suited for addressing constrained, time-varying optimal control problems such as the energy management problem in hybrid-aircraft. The problem formulation to solve the above problem with deterministic DP can be expressed as follows.

Objective function:

$$\min_{u(t)} \mathcal{J}\left(u(t)\right) \tag{3.34}$$

where,

$$\mathcal{J}\left(u(t)\right) = G\left(x(t_f)\right) + \int_0^{t_f} H\left(x(t), u(t), t\right) dt \tag{3.35}$$

System state update model:

$$\dot{x}(t) = F\left(x(t), u(t), t\right) \tag{3.36}$$

Initial states:

$$x(0) = x_0 \tag{3.37}$$

Terminal states constraints:

$$x(t_f) \in \left[x_{f,min}\ ,\ x_{f,max}\right] \tag{3.38}$$

State constraints:

$$x(t) \in \mathcal{X}(t) \subset \mathcal{R}^n \tag{3.39}$$

Control constraints:

$$u(t) \in \mathcal{U}(t) \subset \mathcal{R}^m \tag{3.40}$$

The optimization problem aims to determine an optimal control policy that minimizes the objective function, while ensuring compliance with the time-varying state constraints Equation (3.39) and control input constraints Equation (3.40).

To address the optimization problem using Dynamic Programming (DP), the problem must be discretized. The discretized optimal control problem is formulated as follows.

Consider a discretization with $N + 1$ stages, including the initial stage $k = 0$. At this initial stage, the state is $x(0) = x_0$, and at the terminal stage $k = N$, the state is $x(N) = x_N$. Additionally, let the state and control at stage k be denoted as $x(k) = x_k$ and $u(k) = u_k$, respectively.

The system model in discrete time is,

$$x(k+1) = f\left(x_k, u_k\right) \ \ ;\ k = 0,\, 1,\, 2,\, \ldots,\, N-1 \tag{3.41}$$

Let Π be a control policy such that $\Pi = \{u_0, u_1, \ldots, u_{N-1}\}$.

The total cost of applying Π control policy with the initial state $x(0) = x_0$ is given by,

$$\mathcal{J}_\Pi\left(x_0\right) = g_N\left(x_N\right) + \sum_{k=0}^{N-1} g_k\left(x_k, u_k\right) \tag{3.42}$$

where the first term $g_N\left(x_N\right)$ is the terminal cost over the final state and the second term is the sum of cost-to-go from every stage to the next stage as a function of the current state x_k and the control u_k.

The solution for this discretized optimal problem is the optimal control sequence Π^*. When this control policy is applied to the system described by Equation(3.41), the cost function Equation(3.42) is minimized over the optimal trajectory $x^*(k)$.

$$\mathcal{J}^*_{\Pi}(x_0) = \min_{u \in \Pi} \mathcal{J}_{\pi}(x_0) \tag{3.43}$$

The solution for this formulated problem is determined using the principle of optimality.

The Equation(3.44) is known as the functional equation of DP. It asserts that if one had determined the optimal control, state and the cost from any stage $k+1$ to the terminal stage N, then the corresponding values from any stage k to the terminal stage N can be determined.

$$\mathcal{J}^*_k\left(x_k\right) = \min_{u_k \in \mathcal{U}_k(x_k)} \left\{ g_k\left(x_k, u_k\right) + \mathcal{J}^*_{k+1}\left(x^*_{k+1}\right) \right\} \tag{3.44}$$

This can be further explained by the illustration in Figure(3.16).

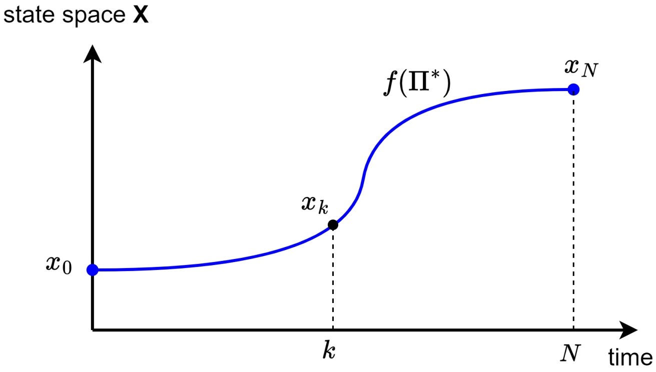


Figure 3.16: Principal of optimality explanation illustration.

Assuming the $\Pi^*(x_0)$ is the optimal trajectory for going from x_0 to x_N, and a point x_k on this trajectory is reached, according to the principal of optimality, the optimal solution to the optimization problem for going from x_k to x_N is $\Pi^*(x_k)$.

The DP optimization process involves two main steps: backward recursion and forward recursion. In the backward recursion the goal is to find the optimal cost-to-go for each state. Starting from the final stage, the problem is solved to find the optimal cost to-go for all points in the state-space grid moving back one stage at a time until the initial stage. The backward recursion of the DP algorithm involves two steps,

1. End cost calculation step:
   The terminal cost calculation is done considering the final stage at $k = N$,

$$\mathcal{J}_N(x_N) = g_N(x_N) \tag{3.45}$$

2. Intermediate cost calculation step:
   For the remaining stages the calculation is performed by,

$$\mathcal{J}_k(x_k) = \min_{u_k \in \mathcal{U}_k(x_k)} \{g_k(x_k, u_k) + \mathcal{J}_{k+1}(f_k(x_k, u_k))\} \tag{3.46}$$

Once the optimal cost-to-go values are obtained from the backward recursion, the forward recursion is executed. In the forward recursion, the optimal state trajectory from the initial stage to the final stage is determined considering the previously computed cost-to-go values. The result from the DP algorithm is the optimal control policy which results the optimal cost. The pseudo code for a simple implementation of the deterministic DP algorithm is presented in Algorithm(1).

```
Data: $X_{\text{grid}}$, $U_{\text{grid}}$, $N$, $f()$
Result: $J$, $U$
for $k = N - 1$ to 1 do
    for all $i_x$ in $X_{grid}$ do
        reset $J_k$;
        for all $i_u$ in $U_{grid}$ do
            $[x_{k+1}, g_k] = f(X_{\text{grid}}(i_x), U_{\text{grid}}(i_u), k, \dots)$;
            $J_k = g_k + J(x_{k+1}, k + 1)$;
            if $J_k < J_{k,previous}$ then
                $J_{\text{opt}} = J_k$;
                $u_{\text{opt}} = U_{\text{grid}}(i_u)$;
            end
        end
        $J(i_x, k) = J_{\text{opt}}$;
        $U(i_x, k) = u_{\text{opt}}$;
    end
end
```

Algorithm 1: Pseudo code for a simple implementation of deterministic DP algorithm [36].

Dynamic Programming (DP) offers several advantages, one of which is its ability to guarantee the globally optimal solution by considering all feasible state trajectories and control policies while adhering to the specified constraints. However, when dealing with non-linear systems, discretizing the models can lead to interpolations between grid points that may not precisely align with the grid points in the next stage, which leads to some error. Additionally, another disadvantage of DP is the curse of dimensionality: as the number of states increases, the computational complexity grows exponentially. Nevertheless, DP is a powerful and very useful technique as it can be used as a optimal performance benchmark.

4

Methods

The advantages and disadvantages of both forward and inverse simulation approaches were thoroughly examined, alongside the expected information as the outputs from the intended simulation model. Key expected results from the model primarily revolve around energy consumption related parameters, while the system dynamics and transients are not the primary focus in this simulation. Taking all these factors into account, the inverse simulation approach was selected for the intended simulation model.

"Pre-flight Prediction of SOF" workflow proposed in the ED-309 document, titled "Guidance on VTOL Energy Level Information Provided to the Crew" by EUROCAE [41], was used as the basis for conceptualizing the workflow of the simulation model. While the original workflow in ED-309 is aimed for eVTOL aircraft, the workflow of the simulation model in this work was adapted to suit an aircraft in independent hybrid-electric configuration.

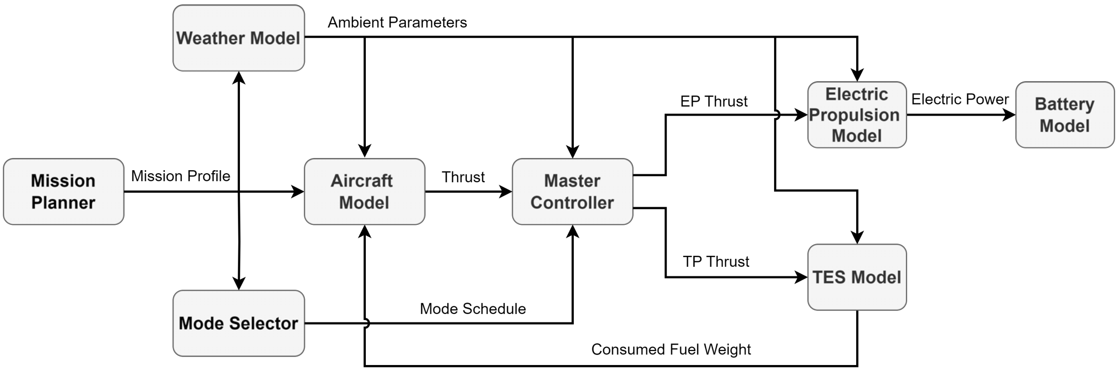


Figure 4.1: Flow diagram for the simulation model for estimating the SoE of hybrid aircraft.

The flow diagram of the proposed simulation model is illustrated in Figure(4.1). The simulation model consists of following main models:

1. Mission Planner
2. Weather Model
3. Mode Selector
4. Aircraft Model

5. Master Controller Model
6. Electric Propulsion Model
7. TES Model
8. Battery Model

The whole simulation is based on the reference mission profile generated by the Mission Planner model. The Weather Model computes the ambient parameters of the aircraft for the duration of the aircraft mission. The Mode Selector model generates and provide the operation mode schedule for the aircraft mission. Based on the ambient parameters and mission profile, the Aircraft Model computes the aircraft-level required thrust. This is the thrust required for the aircraft to be in a state described by the flightpath parameters in the given ambient conditions at each time step. The Master Controller splits the aircraft level thrust required to Electric Propulsion System Model and Turbine Engine System Model based on the operation mode. The Electric Propulsion Model converts the required thrust into required power and the Battery Model determines the battery states when delivering the required power to the electric propulsion system. Similarly, the TES Model determines the fuel consumption to deliver the required power and then the fuel states and aircraft weight changes are computed in the Aircraft Model.

All the models listed above are separately modelled, tested, and integrated together to obtain the complete simulation model.

4.1 Modeling

Each model in the simulation model follows the architecture illustrated in Figure(4.2).

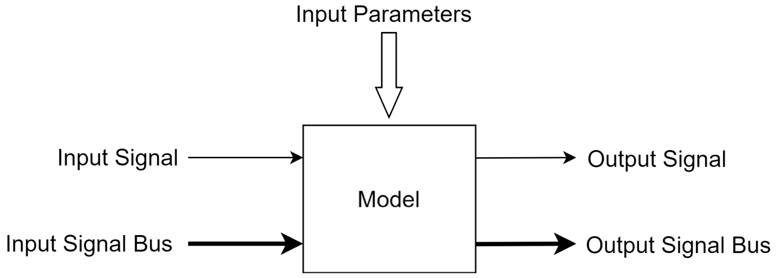


Figure 4.2: Model architecture with input signals, output signals and input parameters.

Input Parameters: Input Parameters are user specified parameters for the model. These are usually setup one-time before running the simulation either by manual entry via simple Graphical User Interface (GUI) or loaded from a predefined file.

Input Signals: Input Signals are the signals taken into the model to perform the computations during the simulation. Individual signal inputs are indicated with narrow arrows while signals buses containing more than one signals are indicated by thick arrows.

Output Signals: Output Signals are the results given out by the model, after performing the computations during the simulation. The narrow arrows indicate individual signals while signal busses are indicated with thick arrows.

The data flow within each model is organized into three distinct layers: Input, Application, and Output layers. This arrangement, as illustrated in Figure(4.3), serves two essential purposes. Firstly, it ensures a clear delineation of the task for each layer. Secondly, it facilitates straightforward troubleshooting, allowing easy identification and resolution of any issue.

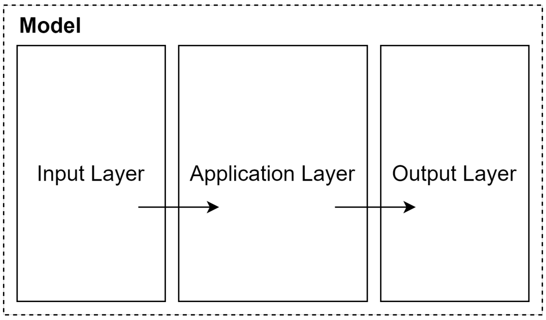


Figure 4.3: Input, Application and Output Layers with arrows pointing the direction of data flow.

Input Layer: The input layer handles receiving all the inputs into the model. Further, selecting the required signals from signal-buses, unit conversions needed within the model are performed in the input layer.

Application Layer: The application layer is the main body within the models. All the computations are performed within this layer.

Output Layer: The output layer handles all the output signals within the models. Unit conversions, data-type conversions as well as signal-bus creation is performed in the output layer.

The simulation models were developed and integrated in MATLAB Simulink environment also leveraging the Simscape and Stateflow.
The high-level function and the methodology used in developing each model, along with the underlying assumptions, are explained in the following sections.

4.1.1 Mission Planner

The Mission Planner serves the crucial function of generating the reference mission profile for the simulation. The mission planner model has the ability to generate a mission profile with all the flight phases in ES-30 typical mission including go-around and flying to alternate and hold. The model is developed based on the following key assumptions:

- The mission profile only considers the longitudinal flight (forward flight with a flightpath angle).
- Turns, manoeuvres, and any roll or yaw rotations are not factored into the mission profile.
- Only steady-state conditions are considered, with transient dynamics of the aircraft being disregarded. This is essentially the state changes are treated as momentary occurrences. For example, the TAS and the FPA momentarily shifts from the climb values to the cruise values at the top of the climb (ToC).

There are no input signals for the mission planner model. The output signals from the mission planner is the FP_{Data} signal-bus as shown in Figure(4.4), which contain signals describing the mission profile of the aircraft (flightpath parameters) for the reference mission.

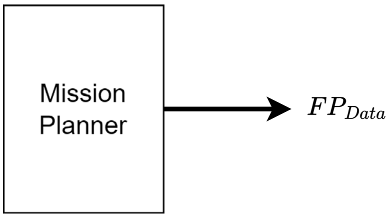


Figure 4.4: Output signal of the Mission Planner.

The input parameters for the Mission Planner are the constraints for different segments of the mission and they are summarised in Table(4.1).

Table 4.1: Input parameters of the mission planner.

Mission Segment	**Input Parameters**
Trip	Trip distance Departure airport elevation Destination airport elevation
Taxi	Taxi-out time Taxi-in time Taxi speed
Take-off and Initial Climb	Take-off distance Take-off speed Initial-climb flightpath angle
Climb	Climb speed Climb flightpath angle
Cruise	Cruise altitude Cruise speed
Descend	Descent speed Descent flightpath angle
Approach and Land	Approach flightpath angle Landing distance Landing speed
Missed-approach or Diversion	Distance to alternate airport Alternate airport elevation Cruise altitude to alternate Hold time Hold speed

Addition to the above parameters, the headwind value is internally retrieved from the Weather Model and used in the computations of the Mission Planner.

The input layer of the Mission Planner reads the input parameters and perform the necessary unit conversions. Further, any input parameter value corresponding to speed in CAS are converted to TAS values using Equation(3.11).

The application layer of the Mission Planner consists of mainly two components.

1. Mission Segments Calculator
2. Mission Profile Generator

4.1.1.1 Mission Segment Calculator

The Mission Segment Calculator model within the Mission Planner serves to compute essential parameters such as speed, acceleration, distance, and time for each mission segment. This model is implemented as a Matlab function within the Simulink environment. It receives input parameters passed from the input layer of the Mission Planner. For each mission segment, the necessary parameter values are calculated using straightforward equations of motion. Generic computations for the Takeoff, Climb and Cruise segment are explained below. The wind component is assumed to be acting along horizontal axis and it can take positive or negative values corresponding to headwind and tailwind conditions respectively.

Takeoff:

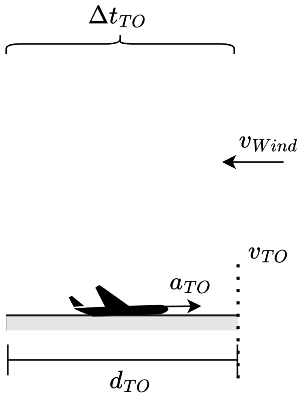


Figure 4.5: Kinematic diagram of aircraft for takeoff segment.

Considering the Figure(4.5),
Inputs: Take-off speed (v_{TO}), Headwind speed (v_{Wind}), Take-off distance (d_{TO}).
Outputs: Aircraft acceleration (a_{TO}), Time for take-off roll (Δt_{TO})
Computations:
Considering the simple equation of motion $v^2 = u^2 + 2as$, take-off acceleration is:

$$a_{TO} = \frac{(v_{TO} - v_{Wind})^2}{2d_{TO}} \tag{4.1}$$

Considering the equation of motion $v = u + at$, time for take-off roll is:

$$\Delta t_{TO} = \frac{v_{TO} - v_{Wind}}{a_{TO}} \tag{4.2}$$

Climb:

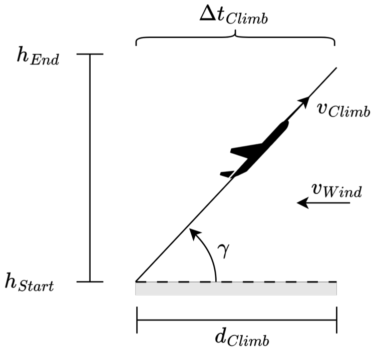


Figure 4.6: Kinematic diagram of aircraft for climb segment.

Considering the Figure(4.6),
Inputs: Initial altitude (h_{Start}), Final altitude (h_{End}), Flightpath angle (γ), Climb speed (v_{Climb}), Headwind speed (v_{Wind})
Outputs: Climb time (Δt_{Climb}), Horizontal distance travelled (d_{Climb})
Computations:
Considering the altitude gain, the climb time can be computed as:

$$\Delta t_{Climb} = \frac{(h_{End} - h_{Start})}{v_{Climb}\sin(\gamma)} \tag{4.3}$$

The horizontal distance travelled in climb is given by:

$$d_{Climb} = (v_{Climb}\cos(\gamma) - v_{Wind}) \cdot \Delta t_{Climb} \tag{4.4}$$

Cruise:

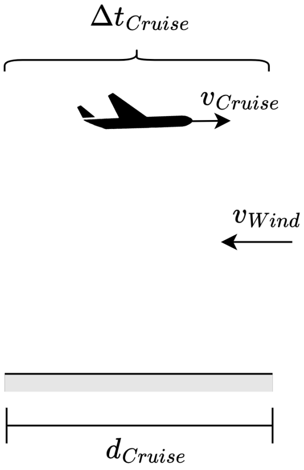


Figure 4.7: Kinematic diagram of aircraft for cruise segment.

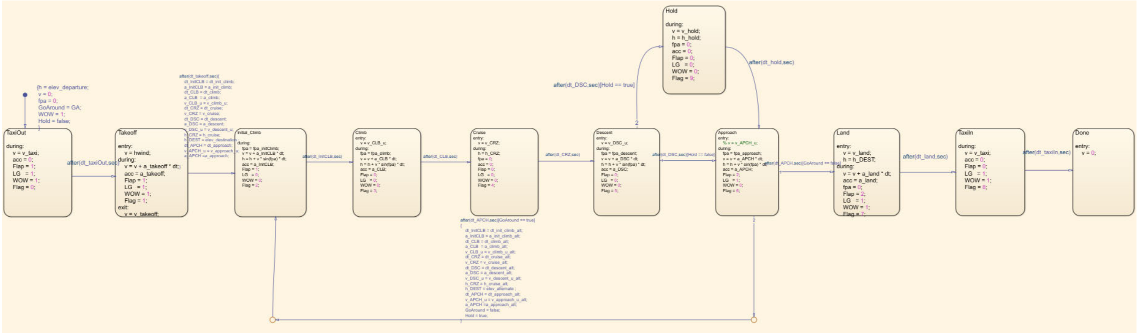
Considering the Figure(4.7),
Inputs: Cruise speed (TAS) (v_{Crusie}), Headwind speed (v_{Wind}), Distance (d_{Cruise})
Outputs: Cruise time (Δt_{Cruise})
Computations:
Considering the simple equation of motions,

$$\Delta t_{Cruise} = \frac{d_{Cruise}}{(v_{Cruise} - v_{Wind})} \tag{4.5}$$

By performing similar computations, the time duration and other relevant parameters for all mission segments are determined. Additionally, during the climb and descent phases, variations in ambient parameters lead to changes in TAS, even when the CAS remains constant. To account for this effect, the computations treat the TAS change as a constant acceleration throughout the corresponding mission phase. The output values from the Mission Segments Calculator model is fed to the Mission Profile Generator.

4.1.1.2 Mission Profile Generator

The Mission Profile Generator model is responsible for generating flightpath variable signals for the simulation. These signals are based on input parameters produced by the Mission Segment Calculator. The model is structured as a finite state machine, with each state corresponding to a main mission segments. Transition conditions involve comparing the simulation time with the duration of each mission segment. State transition occurs based on the simulation time, and each state is active only during its intended time period according to the mission segment. Within each state, equations of motion evolve in discrete time steps to update signals for velocity (v), altitude (h), acceleration ($\dot{v}$) and FPA (γ) signals. The signals for flap configuration ($flap$), weigh-on-wheel state (wow), landing gear configuration (lg), time t and a flag to identify mission segment ($flag$) are generated based on the corresponding states. This model is implemented as state-machine in Simulink using Stateflow as shown in Figure(4.8).



```
{h = elev_departure;
 v = 0;
 fpa = 0;
 GoAround = GA;
 WOW = 1;
 Hold = false;
}

TaxiOut
during:
  v = v_taxi;
  acc = 0;
  Flap = 1;
  LG  = 1;
  WOW = 1;
  Flag = 0;

Takeoff
entry:
  v = hwind;
during:
  v = v + a_takeoff * dt;
  acc = a_takeoff;
  Flap = 1;
  LG  = 1;
  WOW = 1;
  Flag = 1;
exit:
  v = v_takeoff;

after(dt_takeoff,sec){
  dt_InitCLB = dt_init_climb;
  a_InitCLB = a_init_climb;
  dt_CLB = dt_climb;
  a_CLB  = a_climb;
  v_CLB_u = v_climb_u;
  dt_CRZ = dt_cruise;
  v_CRZ = v_cruise;
  dt_DSC = dt_descent;
  a_DSC = a_descent;
  v_DSC_u = v_descent_u;
  h_CRZ = h_cruise;
  h_DEST = elev_destination;
  dt_APCH = dt_approach;
  v_APCH_u = v_approach_u;
  a_APCH =a_approach;
}

Initial_Climb
during:
  fpa = fpa_initClimb;
  v = v + a_InitCLB * dt;
  h = h + v * sin(fpa) * dt;
  acc = a_InitCLB;
  Flap = 1;
  LG  = 0;
  WOW = 0;
  Flag = 2;

Climb
entry:
  v = v_CLB_u;
during:
  fpa = fpa_climb;
  v = v + a_CLB * dt;
  h = h + v * sin(fpa) * dt;
  acc = a_CLB;
  Flap = 0;
  LG  = 0;
  WOW = 0;
  Flag = 3;

Cruise
entry:
  v = v_CRZ;
during:
  h = h_CRZ;
  fpa = 0;
  acc = 0;
  Flap = 0;
  LG  = 0;
  WOW = 0;
  Flag = 4;

Descent
entry:
  v = v_DSC_u;
during:
  fpa = fpa_descent;
  v = v + a_DSC * dt;
  h = h + v * sin(fpa) * dt;
  acc = a_DSC;
  Flap = 0;
  LG  = 0;
  WOW = 0;
  Flag = 5;

Hold
during:
  v = v_hold;
  h = h_hold;
  fpa = 0;
  acc = 0;
  Flap = 0;
  LG  = 0;
  WOW = 0;
  Flag = 9;

Approach
entry:
  % v = v_APCH_u;
during:
  fpa = fpa_approach;
  v = v + a_APCH * dt;
  h = h + v * sin(fpa) * dt;
  acc = a_APCH;
  Flap = 2;
  LG  = 1;
  WOW = 0;
  Flag = 6;

Land
entry:
  v = v_land;
  h = h_DEST;
during:
  v = v + a_land * dt;
  acc = a_land;
  fpa = 0;
  Flap = 2;
  LG  = 1;
  WOW = 1;
  Flag = 7;

TaxiIn
during:
  v = v_taxi;
  acc = 0;
  Flap = 0;
  LG  = 1;
  WOW = 1;
  Flag = 8;

Done
entry:
  v = 0;

after(dt_APCH,sec)[GoAround == true]
{
  dt_InitCLB = dt_init_climb_alt;
  a_InitCLB = a_init_climb_alt;
  dt_CLB = dt_climb_alt;
  a_CLB  = a_climb_alt;
  v_CLB_u = v_climb_u_alt;
  dt_CRZ = dt_cruise_alt;
  v_CRZ = v_cruise_alt;
  dt_DSC = dt_descent_alt;
  a_DSC = a_descent_alt;
  v_DSC_u = v_descent_u_alt;
  h_CRZ = h_cruise_alt;
  h_DEST = elev_alternate ;
  dt_APCH = dt_approach_alt;
  v_APCH_u = v_approach_u_alt;
  a_APCH =a_approach_alt;
  GoAround = false;
  Hold = true;
}
```

Figure 4.8: Stateflow implementation of Mission Profile Generator model within Mission Planner Model.

The Mission Segment Calculator and Mission Profile Generator models are integrated together within the application layer of the Mission Planner model as shown in Figure(4.9).

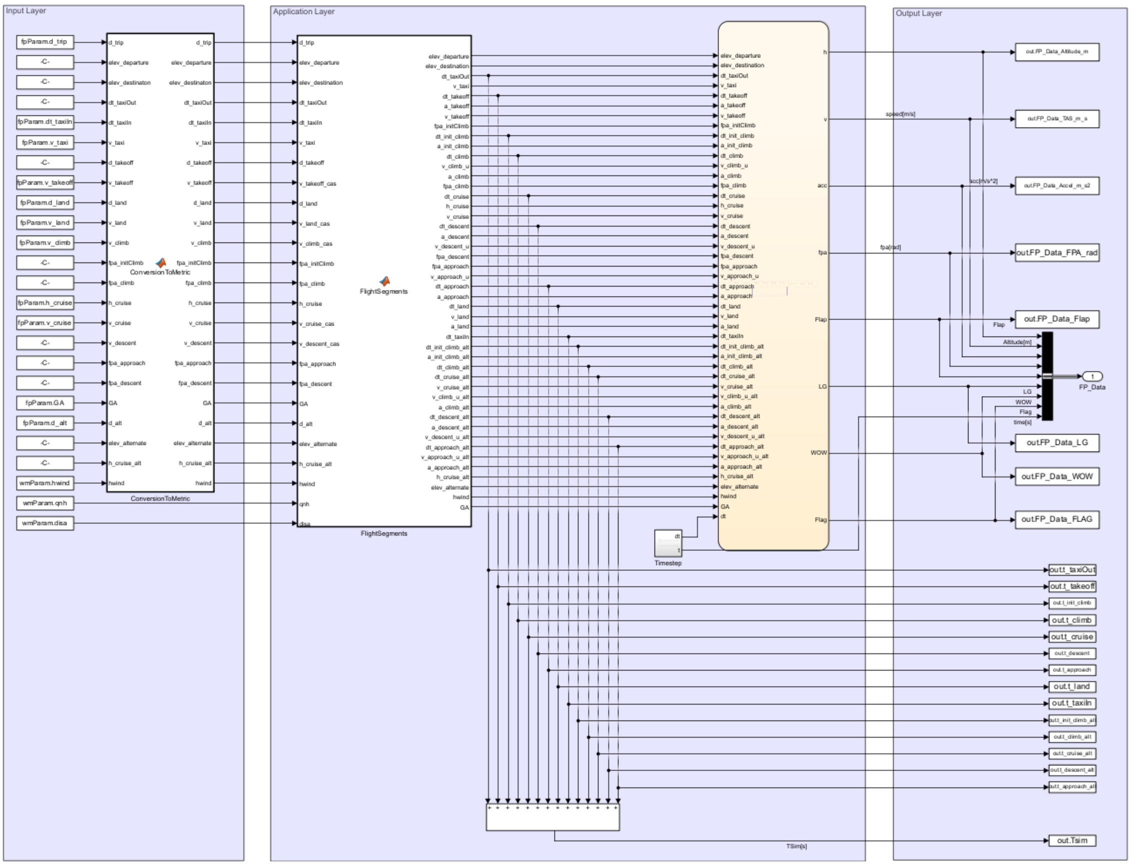


Figure 4.9: Simulink implementation of the Mission Planner.

The output layer of the Mission Planner creates the Flightpath Signal Bus FP_{Data} including the flightpath signals listed in Table(4.2).

Table 4.2: Signals in the FP_{Data} signal-bus output from Mission Planner.

Signal Description	**Symbol**
Altitude	h
Speed	v
Acceleration	$\dot{v}$
Flightpath angle	γ
Flap configuration	$flap$
Landing gear configuration	lg
Weight-on-wheels state	wow
Flag	$flag$
Time	t

4.1.2 Weather Model

The Weather Model allows users to specify or modify weather conditions for the simulation. It generates signals with the ambient parameters of the aircraft based on user-specified parameters and the mission profile. The input parameters for the weather model are listed in Table(4.3) and these parameters can be easily modified by the user using the model mask.

Table 4.3: Input Parameters for the Weather Model

Input Parameters	**Variable**	**Unit**
Headwind	v_{Wind}	kts
DISA	$disa$	^{o}C
QNH	qnh	hPa

The input and output signals for the Weather Model is shown in Figure(4.10).

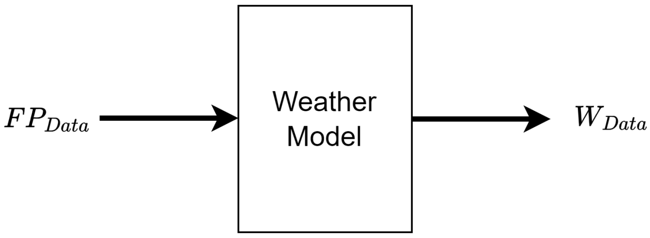


Figure 4.10: Input and output signals of the Weather Model.

The input signal for the weather model is the FP_{Data} signal-bus from the Mission Planner and the model outputs W_{Data} signal-bus which contains the ambient parameter signals. The input layer of the weather model selects the altitude signal (h) from the FP_{Data} signal-bus and perform the required unit conversions. The output layer creates the output signal-bus including the ambient parameters listed in the Table(4.4).

Table 4.4: Ambient parameter signals from the Weather Model.

Signal Description	**Symbol**
Pressure altitude	h_p
Static pressure	P_s
Outside air temperature	OAT
Air density	ρ
Speed of sound	a

The application layer of the weather model consists of the following models to determine the output ambient parameter signals in the Table(4.4).

1. Pressure Altitude Model

2. Outside Air Temperature Model
3. Static Pressure Model
4. Air Density Model
5. Speed of Sound Model

4.1.2.1 Pressure Altitude Model

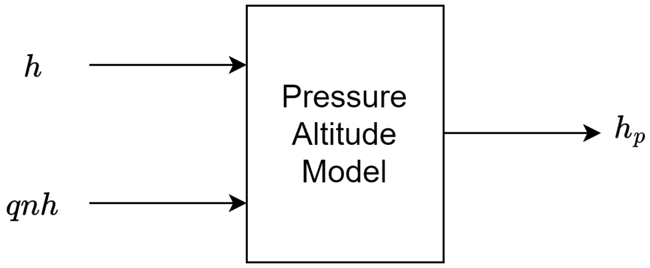


Figure 4.11: Inputs and output of Pressure Altitude Model.

The Pressure Altitude Model takes the h signal and the user specified qnh value as the inputs as shown in Figure(4.11). This model is implemented as a MATLAB function in Simulink and it computes the pressure altitude (h_p) using the Equation(3.9).

4.1.2.2 Outside Air Temperature Model

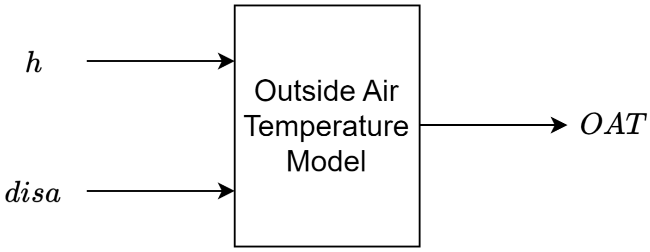


Figure 4.12: Inputs and output of Outside Air Temperature Model.

The Outside Air Temperature Model takes the h signal and the user specified $disa$ value as the inputs as shown in Figure(4.11). This model is implemented as a MATLAB function in Simulink and it computes the OAT using the Equation(3.1) and Equation(3.3).

4.1.2.3 Static Pressure Model

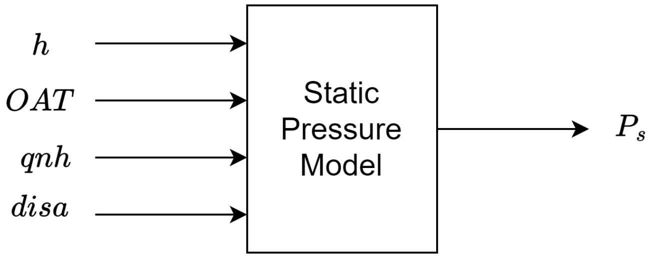


Figure 4.13: Inputs and output of Static Pressure Model.

The Static Pressure Model takes h signal, user specified qnh and $disa$ values along with the OAT signal, the output from the Outside Air Temperature model as the inputs as shown in Figure(4.13). This model is implemented in Simulink as a MATLAB function using the Equation(3.2) and Equation(3.4) to compute the output signal for static pressure (P_s).

4.1.2.4 Air Density Model

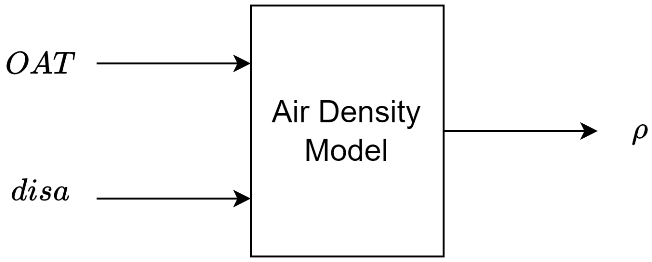


Figure 4.14: Inputs and output of Air Density Model.

The Air Density Model takes user specified $disa$ value and the OAT signal from Outside Air Temperature as the inputs as shown in Figure(4.14). The MATLAB function based implementation of the model computes the air density ρ using the Equation(3.5) and Equation(3.6).

4.1.2.5 Speed of Sound Model

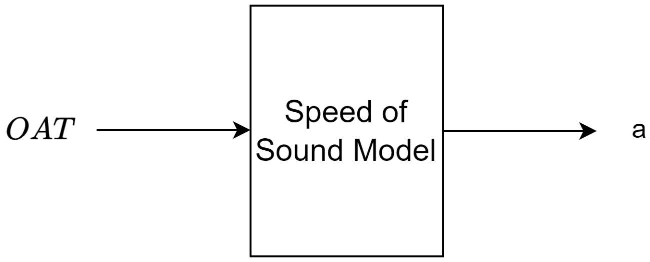


Figure 4.15: Inputs and output of Speed of Sound Model.

The speed of sound model takes OAT, the output signal from the Outside Air Temperature model as the input and the speed of sound (a) is computed using Equation(3.7) as shown in Figure(4.15). The model is implemented as a MATLAB function in Simulink.

The Weather Model was created by integrating the five models described above, as illustrated in Figure(4.16).

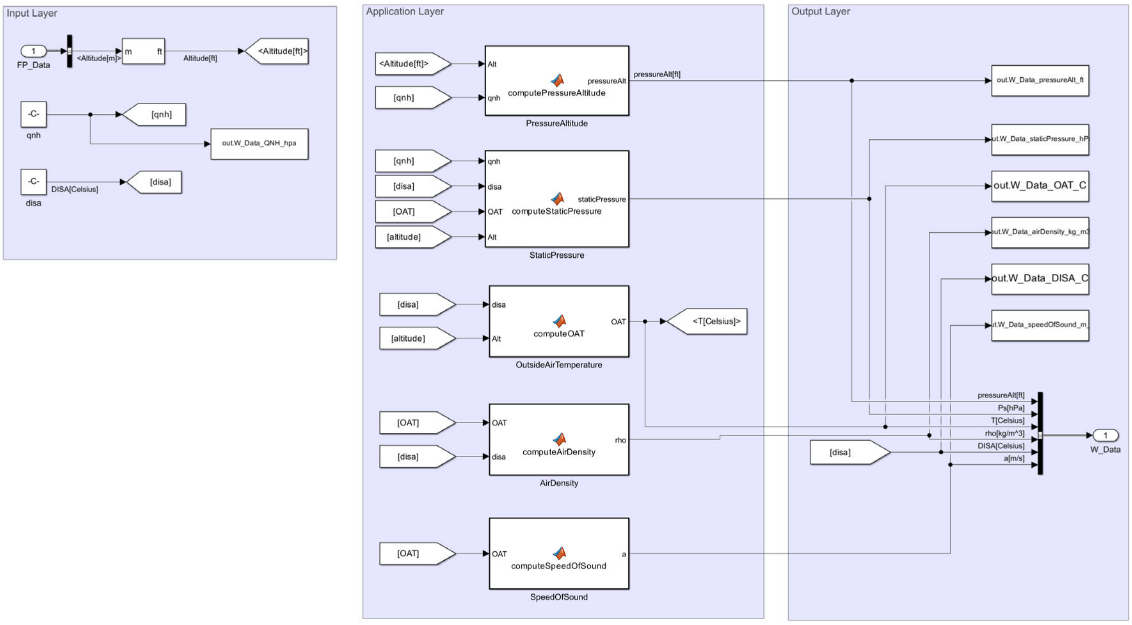


Figure 4.16: Simulink implementation of the Weather Model.

The output layer of the Weather Model creates the W_{Data} signal-bus, including the ambient parameter signals listed in Table(4.4).

4.1.3 Mode Selector

The Mode Selector model allows users to choose different modes of operation for various mission segments within the mission profile for the simulation. The input and output signals of the Mode Selector are depicted in Figure(4.17).

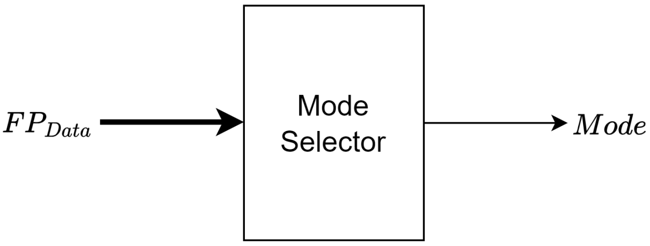


Figure 4.17: Input and output signal of Mode Selector.

The Mode Selector model receives the FP_{Data} signal-bus from Mission Planner as its input. The operation modes for different segments of the mission profile selected by the user is retrieved from the model as the input parameters. This mode selection is facilitated through a simple Graphical User Interface (GUI) through the mask of the Mode Selector. The $Mode$ signal is the output from this model, where this signal represents the operation mode for each time step of the simulation.

The process of assigning user-selected operation modes to different segments of the mission profile, with respect to the simulation time is primarily executed through the mask using callback functions. Within the application layer of this model, a function generates the output signal based on the user-specified modes of operation. During simulation execution, the output signal $Mode$ at each time step is assigned a value corresponding to the previously executed mode assignment.
The Simulink implementation of the Mode Selector is shown in Figure(4.18).

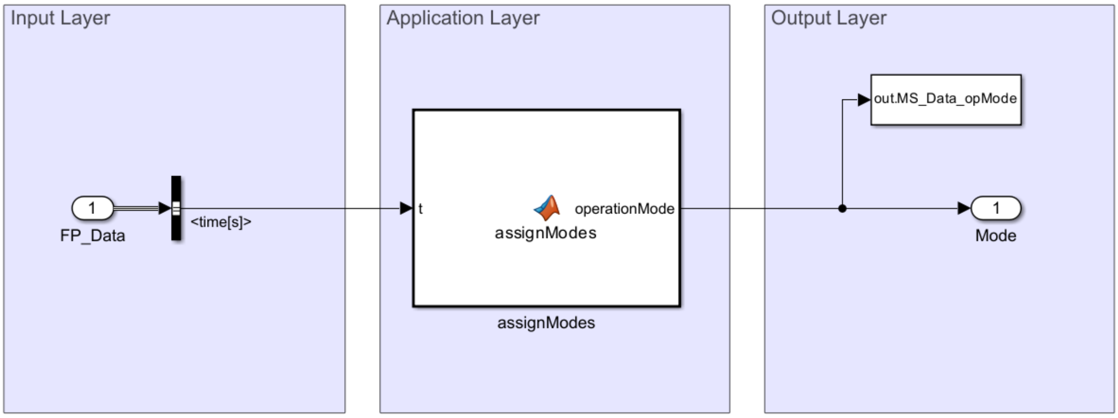


Figure 4.18: Simulink implementation of the Mode Selector.

4.1.4 Aircraft Model

The Aircraft Model within the simulation determines the aircraft-level required thrust force by solving flight dynamics and force balance equations. It considers the mission profile and ambient conditions to compute the required thrust.

To make the simulation setup more convenient for the user, specification of initial parameters for the simulation corresponding to initial weight and energy levels is facilitated through the mask of the Aircraft Model. The Aircraft Model accepts the following input parameters listed in Table(4.5).

Table 4.5: Input Parameters for the Aircraft Model

Input Parameters	**Unit**
Aircraft zero-fuel weight (m_{zf})	kg
Initial fuel onboard ($m_{\text{fuel,init}}$)	kg
Initial Battery SoC (SoC_{init})	%

The input and output signals of the Aircraft Model are shown in Figure(4.19).

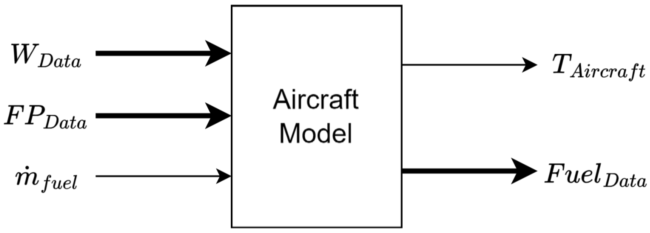


Figure 4.19: Input and output signals of Aircraft Model.

The Aircraft Model receives the W_{Data}, FP_{Data} and $\dot{m}_{fuel}$ as the inputs, where $\dot{m}_{fuel}$ is an output signal from the TES Model. From the FP_{Data} signal-bus, the flight-path parameter signals v, $\dot{v}$, γ, $flap$, lg and wow signals are extracted. Similarly, ρ signal is extracted from the W_{Data} signal-bus. Along with these extracted signals, the $\dot{m}_{fuel}$ is sent to the application layer of the Aircraft Model.

The output from the aircraft model is the aircraft-level required thrust $T_{Aircraft}$ signal and the $Fuel_{Data}$ signal-bus containing the fuel consumption related signals.

The application layer of the aircraft model mainly consists of the following sub-models.

1. Aircraft Point-mass Model
2. Aircraft Lift-drag Polar Model
3. Aircraft Mass Model
4. Remaining Fuel Model

4.1.4.1 Aircraft Point-mass Model

The function of the Aircraft Point-mass Model is to solve the force balance equations to compute the aircraft-level thrust required at each time step in the simulation. The input and the output signals of the model are shown in Figure(4.20).

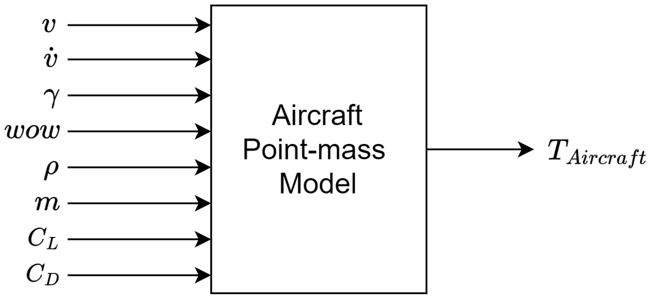


Figure 4.20: Input and output signals of Aircraft Point-mass Model.

The signals v, $\dot{v}$, γ, wow, and ρ are input signals from the input layer of the Aircraft Model. The signal m is output signal from the Aircraft Mass Model and both C_L and C_D signals are output signals from the Aircraft Lift-drag Polar Model.

A point-mass model with 3DoF dynamics are considered in the Aircraft Point-mass Model. Further, the forces acting on the aircraft along the vertical-plane are only considered.

The following equation from Equation(3.18) considering the force balance of the aircraft gives $T_{Aircraft}$, the required thrust at any time step.

$$T_{Aircraft} = m\,\dot{v} + D + m\,g\,sin(\gamma) \tag{4.6}$$

where, g is the gravitational acceleration.

D is the total force acting on the aircraft against its forward motion. The D has two components to it and they are the aerodynamic drag force D_{aero} and the friction force $D_{friction}$ acting on the aircraft wheels only during takeoff and landing roll.

$$D = D_{aero} + D_{friction} \tag{4.7}$$

D_{aero} is computed within the point-mass model using Equation(3.17) as follows.

$$D_{aero} = \frac{1}{2}\,C_D\,\rho\,v^2\,S_{ref} \tag{4.8}$$

The S_{ref} value is preset within the model aligning with the ES-30 specification.

When the aircraft is on ground, the friction force acting on the aircraft wheels are parameterized using the following equation.

$$D_{friction} = \mu\,N \tag{4.9}$$

where, μ is the friction coefficient of the wheels and N is the normal force acting on the wheels vertically up from the ground. Considering the vertical force balance, N is given by,

$$N = m\,g - L \tag{4.10}$$

where, L is the lift force acting on the aircraft which is calculated considering the force balance of the aircraft as per the Equation(3.19).

These set of equations are integrated as a custom Simscape model using Simscape Language, leveraging the acausal modeling capability. The Simscape based model of the Aircraft Point-mass Model is shown in Figure(4.21).

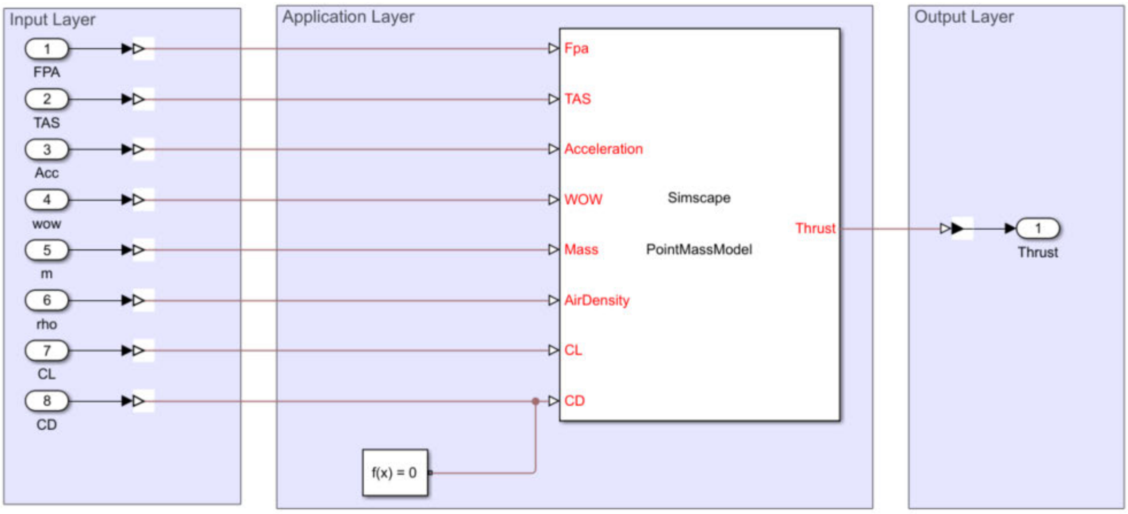


Figure 4.21: Simscape based implementation of the Aircraft Point-mass Model.

4.1.4.2 Aircraft Lift-drag Polar Model

The function of the Lift-drag Polar Model is to compute the lift-coefficient (C_L) and the drag-coefficient (C_D) of the aircraft for the simulation.

The input and output signals of the Lift-drag Polar Model is shown in Figure(4.22).

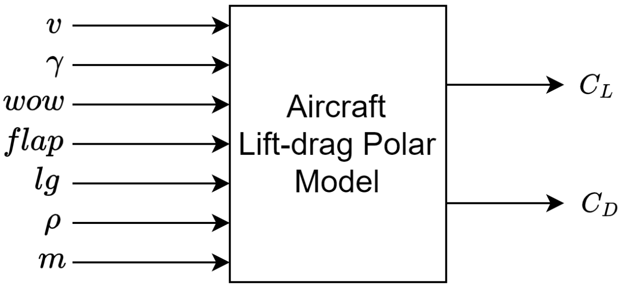


Figure 4.22: Input and output signals of Aircraft Lift-drag Polar Model.

The signals v, γ, wow, $flap$, lg, ρ are from the input layer of the Aircraft Model and the signal m is the output signal of Aircraft Mass Model.

Computing C_L :

Following derivation was performed to determining the momentarily C_L value considering the forces acting on the aircraft at any time step.
From the Equation(3.19),

$$L = m\, g\, cos(\gamma) + m\, v\dot{\gamma} \tag{4.11}$$

Substituting L from Equation(3.12),

$$m\, g\, cos(\gamma) + m\, v\, \dot{\gamma} = \frac{1}{2} C_L\, \rho\, v^2\, S_{ref} \tag{4.12}$$

The $mv\dot{\gamma}$ is the centrifugal force due to the rate of change of flightpath angle. Fundamentally from the quasi-static simulation approach, we are not interested in the transients of the changes of γ. Moreover, the γ values and also the changes in γ values are small throughout the aircraft missions relevant to this simulation model. Therefore, $\dot{\gamma}$ was neglected and the following simplified formula was derived.

$$m\, g\, cos(\gamma) = \frac{1}{2}\, C_L\, \rho\, v^2\, S_{ref} \tag{4.13}$$

The C_L value is given by,

$$C_L = \frac{2\, m\, g\, cos(\gamma)}{\rho\, v^2\, S_{ref}} \tag{4.14}$$

Computing C_D :

Considering the Equation(3.14) the drag coefficient C_D is given by,

$$C_D = C_{D_0} + K\, C_L^2 + \Delta C_{D_{Flap}} + \Delta C_{D_{LG}} \tag{4.15}$$

Here the zero-lift drag coefficient C_{D_0} is a constant and the value is preset within the model.

K and $\Delta C_{D_{Flap}}$ are values depend on the flap configuration. Therefore the values for K is mapped inside the model as a function of flap configuration. Based on the value of signal $flap$ at each time step, K and $\Delta C_{D_{Flap}}$ are assigned the corresponding pre-defined values.

The $\Delta C_{D_{LG}}$ corresponds to the extra drag component due to extension of the landing gears, hence it depends on the signal lg. $\Delta C_{D_{LG}}$ is assigned the corresponding value based on the signal lg within the model.

Based on the above described values and the computed C_L value, the drag coefficient C_D is determined in the Aircraft Lift-drag Polar Model. This model is implemented as a MATLAB function in the Simulink environment within the Aircraft Model as shown in Figure(4.23).

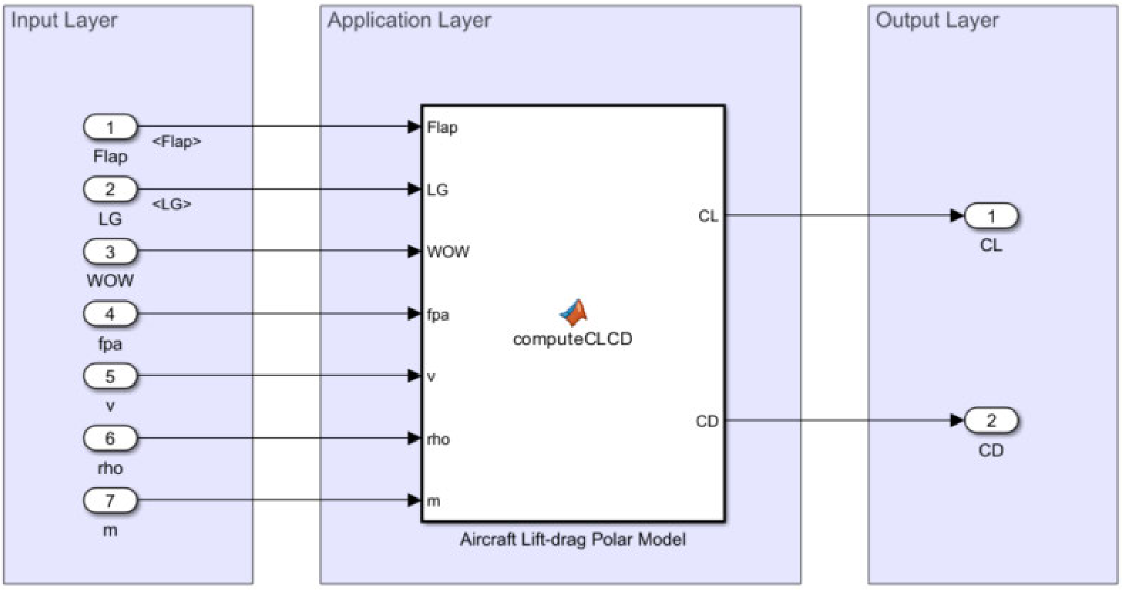


Figure 4.23: Simulink implementation of the Aircraft Lift-drag Polar Model.

4.1.4.3 Aircraft Mass Model

The function of the Aircraft Mass Model is to update the aircraft mass as the mission progresses, accounting for the mass reduction due to fuel burn.

The input and output signals of the Aircraft Mass Model is shown in Figure(4.24).

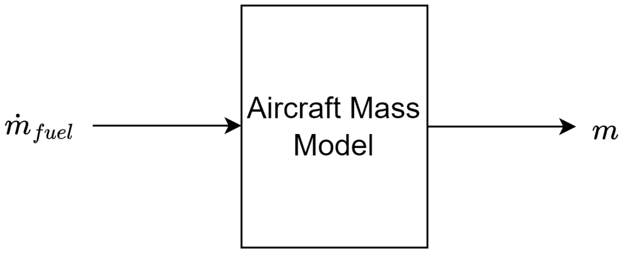


Figure 4.24: Input and output signals of the Aircraft Mass Model.

The model takes $\dot{m}_{fuel}$ signal from the TES Model, which expresses the mass of fuel consumption per unit time. It also considers the initial aircraft weight computed cinsidering the user specified initial parameters m_{zf} and $m_{\text{fuel,init}}$.

$$M_{\text{init}} = m_{\text{zf}} + m_{\text{fuel,init}} \tag{4.16}$$

The model computes the aircraft mass m at any time step by subtracting the cumulative value of consumed fuel mass from the initial aircraft mass. The Aircraft Mass Model is implemented as a Simulink model as shown in the Figure(4.25).

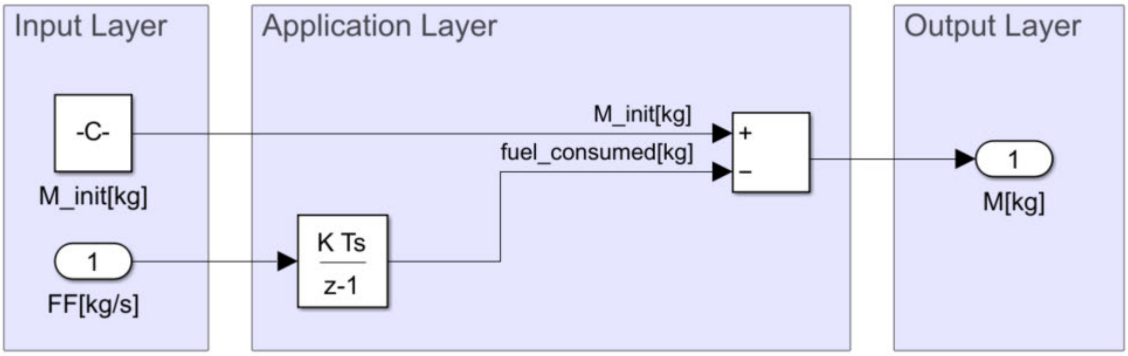


Figure 4.25: Simulink implementation of the Aircraft Mass Model to determine the changing aircraft mass due to fuel burn.

4.1.4.4 Remaining Fuel Model

The Remaining Fuel Model is responsible for calculating and updating the remaining fuel in the aircraft as a result of fuel consumption during the mission. The input and output signals of the Remaining Fuel Model is shown in the Figure(4.26).

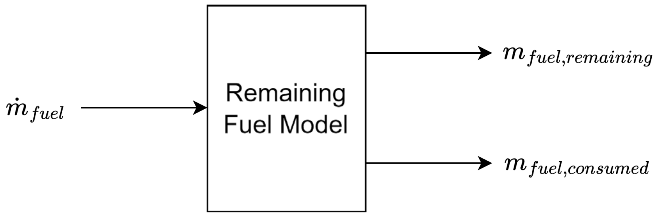


Figure 4.26: Simulink model of the remaining fuel model to update the state of remaining amount of fuel.

The model takes the $\dot{m}_{fuel}$ signal as an input and also considers the user specified initial FoB $m_{\text{fuel,init}}$ value. The cumulative amount of consumed fuel mass is subtracted from the initial FoB to determine the remaining amount of fuel at each time step. Saturation block is used to enforce the fuel-tank limits to the model. Simulink implementation of the model is shown in Figure(4.27).

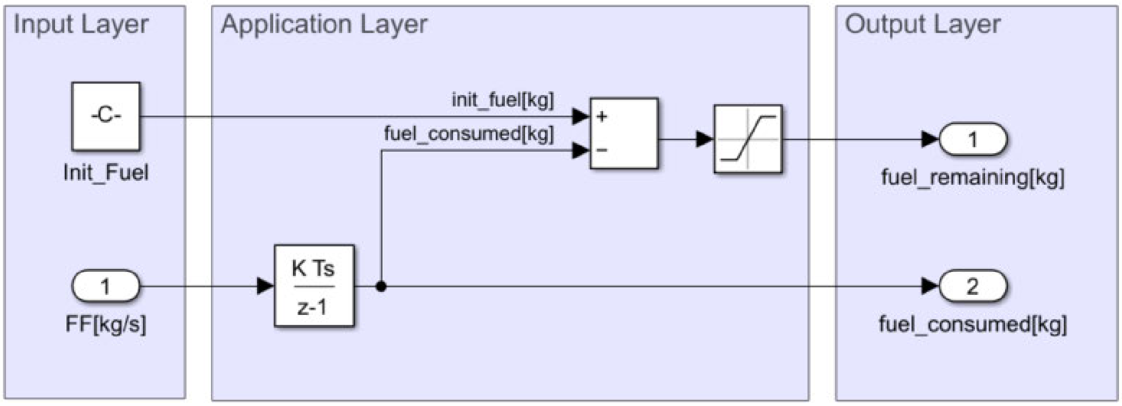


Figure 4.27: Simulink model of the remaining fuel model to update the state of remaining amount of fuel.

The above described four models are integrated to obtain the Simulink model for the Aircraft Model as shown in Figure(4.28). The output layer creates and outputs the $T_{Aircraft}$, along with fuel consumption related signal-bus $Fuel_{Data}$.

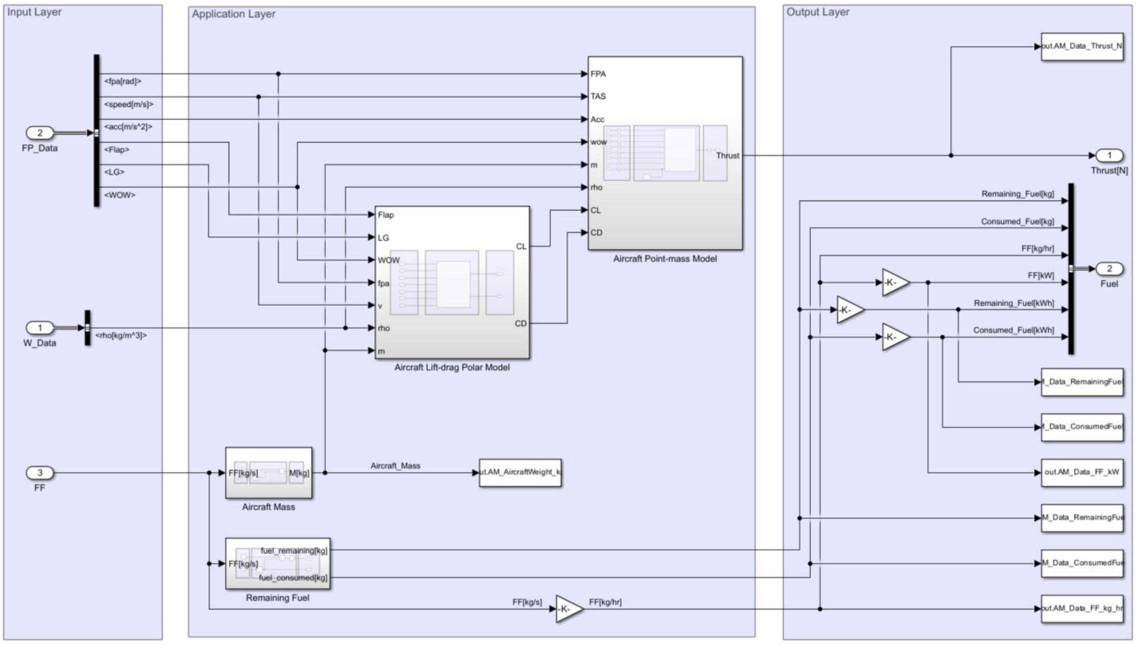


Figure 4.28: Simulink implementation of the Aircraft Model.

4.1.5 Master Controller Model

The function of the Master Controller Model is to divide the aircraft level thrust required $T_{Aircraft}$ between the Electric Propulsion System Model and TES Model based on the operation mode schedule. The Master Controller Model considers the following parameters along with the mode of operation to split the aircraft-level thrust. The input parameters for the Master Controller Model are:

$T_{TES,MCP}$: Thrust when TES at maximum continuous power
$T_{EPS,MCP}$: Thrust when EPS at maximum continuous power
$T_{TES,IDLE}$: Thrust when TES at IDLE power

The input and output signals of the Master Controller Model are shown in Figure(4.29).

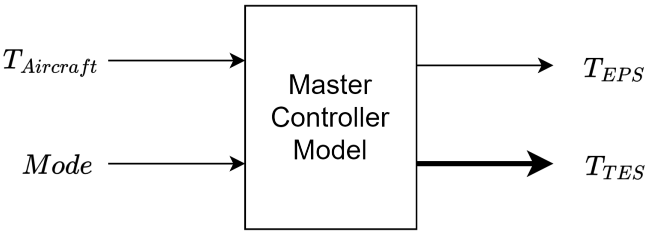


Figure 4.29: Input and output signals of Master Controller Model.

The inputs for the Master Controller Model are $T_{Aircraft}$ and $Mode$ signals which are outputs from Aircraft Model and Mode Selector respectively. The application layer of the model is based on a rule-based thrust-split algorithm, which decides the split of the $T_{Aircraft}$ between the electric and turbine propulsion systems based on the operation mode and the system limitations specified as the input parameters. The rule-based algorithm for the Master Controller Model is presented in Algorithm(2).

The Algorithm(2) is implemented in the application layer of the Master Controller Model using Enabled Subsystem Blocks, each corresponding to a different operation mode. The Simulink implementation of the model is shown in Figure(4.30).

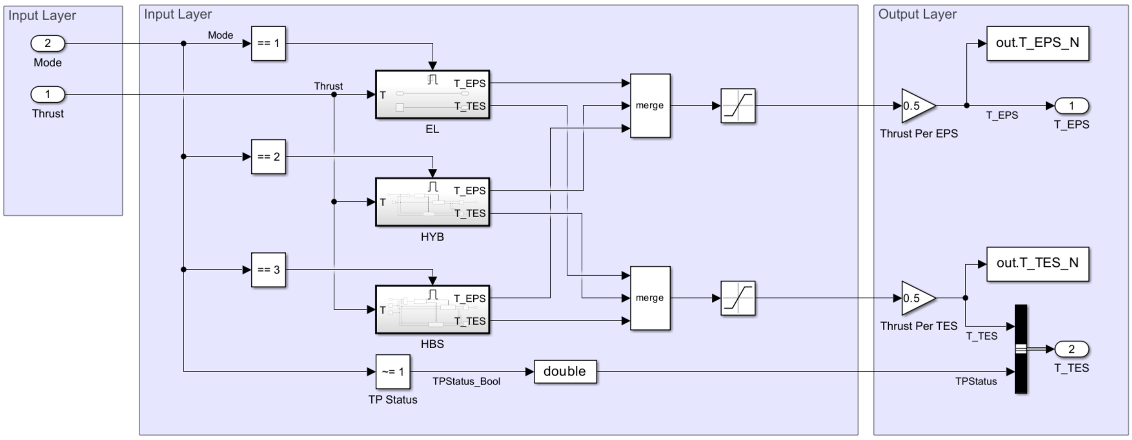


Figure 4.30: Simulink model of Master Controller

```
Data: $T_{Aircraft}$, $Mode$, $T_{TES,MCP}$, $T_{TES,IDLE}$, $T_{EPS,MCP}$
Result: $T_{EPS}$, $T_{TES}$, $TP_{Status}$
if $Mode == EL$ then
    $T_{EPS} \leftarrow T_{Aircraft}$;
    $T_{TES} \leftarrow 0$ ;
    $TP_{Status} \leftarrow False$;
else
    if $Mode == HYB$ then
        $TP_{Status} \leftarrow True$;
        if $T_{Aircraft} \leq T_{TES,MCP}$ then
            $T_{TES} \leftarrow T_{Aircraft}$;
            $T_{EPS} \leftarrow 0$ ;
        else
            $T_{TES} \leftarrow T_{TES,MCP}$ ;
            $T_{EPS} \leftarrow T_{Aircraft} - T_{TES,MCP}$ ;
        end
    end
    if $Mode == x$ then
        $TP_{Status} \leftarrow True$;
        if $T_{Aircraft} \leq T_{EPS,MCP} + T_{TES,IDLE}$ then
            $T_{TES} \leftarrow T_{TES,IDLE}$;
            $T_{EPS} \leftarrow T_{Aircraft} - T_{TES,IDLE}$ ;
        else
            $T_{EPS} \leftarrow T_{EPS,MCP}$ ;
            $T_{TES} \leftarrow T_{Aircraft} - T_{EPS,MCP}$ ;
        end
    end
end
```

Algorithm 2: Rule-based algorithm for the Master Controller Model.

The output layer of the Master Controller Model creates the output signal-bus T_{TES} including the TP_{Status} and T_{TES} signals. The output signals of the Master Controller Model are T_{TES} signal-bus and the signal T_{EPS}.

4.1.6 Electric Propulsion System Model

The Electric Propulsion System model represents the Electric propulsion system of the hybrid aircraft. The function of this model is to compute the electrical power required from the battery system, when delivering the requested thrust.

The input and output signals of the Electric Propulsion System model are depicted in Figure(4.31).

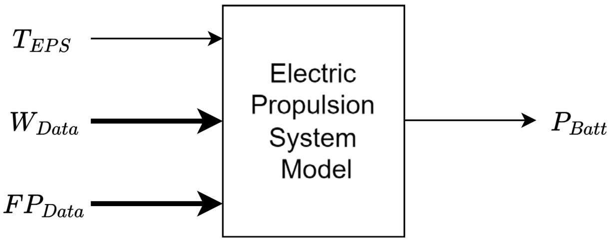


Figure 4.31: Input and output signals of Electric Propulsion System model

T_{EPS} signal is an output signal from the Master Controller Model and the FP_{Data} is the signal-bus output from Weather Model. The input layer receives these signals and the v signal is extracted from the FP_{Data} signal-bus. Similarly, ρ and a are extracted from the W_{Data} signal-bus . These signal along with the T_{EPS} signal are sent to the application layer.

The application layer of the Electric Propulsion System Model consists of the following three main sub-models.

1. Propeller Model
2. PMSM Model
3. Inverter-loss Model

The methodology followed for modelling these main three sub-models of the Electric Propulsion System Model are explained below.

4.1.6.1 Electric Propulsion System Propeller Model

The propellers considered in this simulation setup are constant speed propellers. To achieve a requested thrust under the given ambient parameters from propellers, there are 2 DoF to be controlled. They are the propeller pitch and the propeller rotational speed.

The approach of modeling propellers using propeller performance maps was chosen with the intent of capturing the propeller characteristics as close as possible to the real system.

Instead of using a dynamic controller in the Electric Propulsion System Model, the approach to implement a map based propeller model as depicted in Figure(4.32) chosen. This map based propeller is based on the results from an externally run propeller optimization algorithm. The propeller model takes T_{EPS}, v, a and ρ as the inputs and outputs the optimal propeller rotation speed ω, propeller torque τ and the propeller efficiency η based on a lookup table.

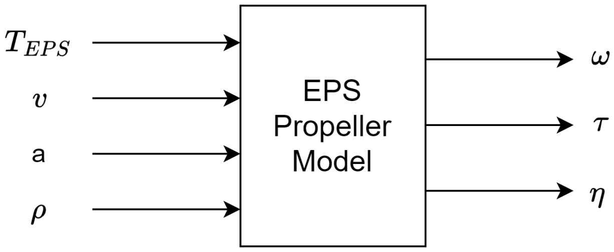
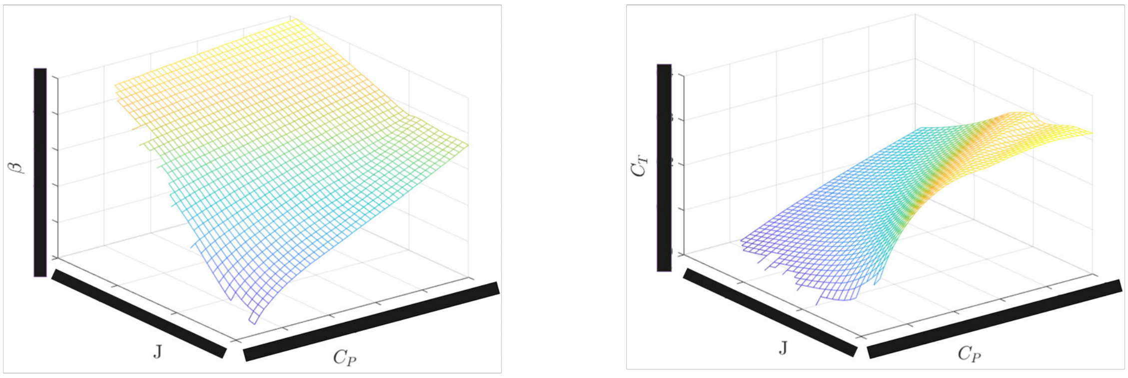


Figure 4.32: Input and output signals of Electric Propulsion System propeller model

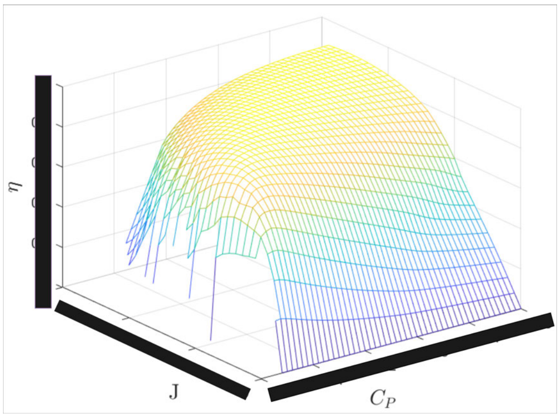
The propeller optimization algorithm used to generate the lookup tables for the propeller model was based on the local controller control law suggested in the previous study [42]. New specifications of the ES-30 EPS system were incorporated when generating the lookup tables, for example the propeller diameter values. The propeller performance maps of a non-proprietary propeller from Hartzell Propeller Inc. provided by Heart Aerospace were used. The provided propeller maps were raw data tables of propeller pitch β, propeller thrust coefficient C_T and propeller efficiency η_{prop} against propeller advance ratio J, propeller power coefficient C_P and propeller tip Mach M_{tip}.
The first step was to import these set of raw data into MATLAB environment and examine. The independent variable ranges were as follows.

C_P : 48 discretised points in $[0.03, 0.5]$
J : 30 discretised points in $[0.03, 2.9]$
M_{tip} : $[0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$

β, C_T and η_{prop} contains values in $48 \times 30 \times 6$ sized grids. The propeller maps for β, C_T and η_{prop} when $M_{tip} = 0.5$ is visualized in Figure(4.33a), Figure(4.33b) and Figure(4.33c) respectively.



(a) Propeller performance map for β **(b)** Propeller performance map for C_T



(c) Propeller performance map for η_{prop}

Figure 4.33: Propeller performance maps for β, C_T and η_{prop} when $M_{tip} = X$.

Closely following the steps suggested in [42], the workflow with following steps was formulated to find the optimal propeller speed, pitch, and torque values corresponding to a thrust request for given flightpath variables and ambient conditions.

Step 1: Discretize the motor shaft speed range by a fixed step-size to obtain the set W, which is the possible set of speeds for the propeller to operate.

$$W = \{\omega \mid \omega \in [\omega_{motor,min} \, , \, \omega_{motor,max}]\} \tag{4.17}$$

Step 2: Sweep through the values in set W to determine the set of propeller speeds which results a $M_{tip} \in [M_{tip,min}, M_{tip,max}]$ based on the v, a, and propeller diameter D values, considering the Equation(3.22). This set of propeller speeds is the valid set of propeller speed to operate W_{valid}.

$$W_{valid} = \{\omega \mid \omega \in W \wedge \omega \rightarrow M_{tip} \in [M_{tip,min} \, , \, M_{tip,max}]\} \tag{4.18}$$

Step 3: Sweep through the propeller speed values in set W_{valid} and compute the corresponding J, M_{tip} and C_T values using Equation(3.21), Equation(3.22) and Equation(3.23) respectively.

Step 4: Using the the propeller performance map for C_T, interpolate the C_T map for the each M_{tip} value corresponding to $\forall \omega \in W_{valid}$. Using these interpolated C_T maps, evaluate the C_P values for each $\omega \in W_{valid}$ which correspond to the J and C_T values already computed in Step 3.

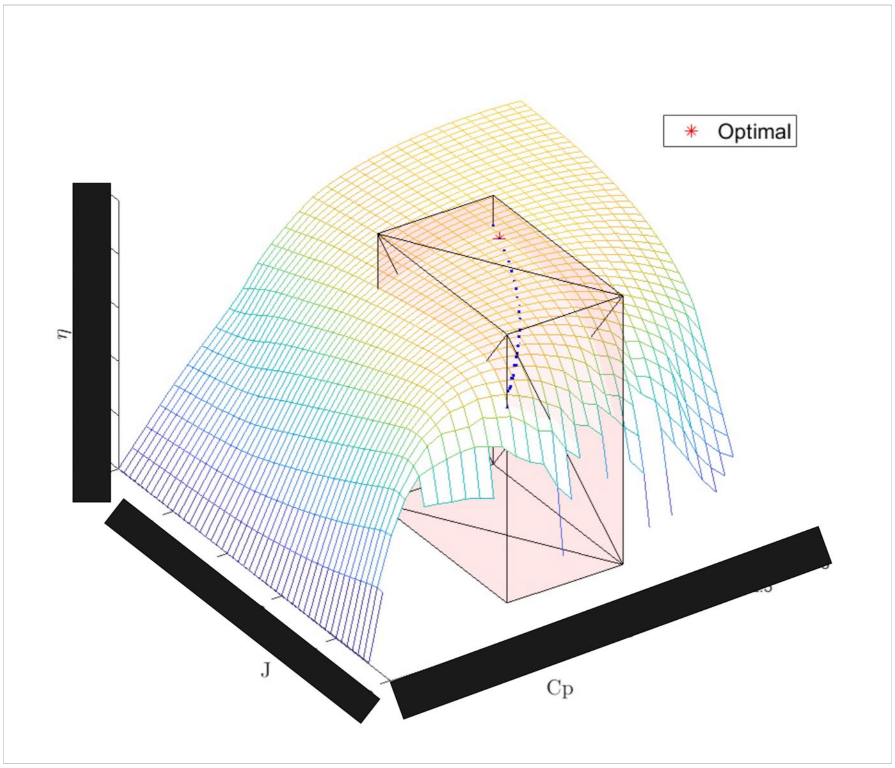
Step 5: The obtained set of points defined by $\{< C_P,\ J,\ M_{tip} > \ | \, \omega \in W_{valid}\}$ are the valid operating points for the propeller to operate for the given thrust request, TAS and ambient conditions.

Step 6: For each operating points found in Step 5, the propeller efficiency η_{prop} value is interpolated using propeller performance map for η_{prop}. Then the operating point resulting the maximum efficiency η^*_{prop} is chosen as the optimal operating point and this optimal operating point is defined by $< C^*_P,\ J^*,\ M^*_{tip} >$.

Step 7: Then the optimal propeller pitch angle corresponding to the optimal operating point $< C^*_P,\ J^*,\ M^*_{tip} >$ is interpolated from the propeller performance map for β.

Step 8: The torque τ is computed using Equation(3.24) and Equation(3.25). The shaft power can be computed either in terms of Equation(3.25) or using the ω and τ values.

The working principal of this optimization workflow can be visualized as follows for better understanding: Given a motor speed range $[0, 1500]$ rpm with the discretization with the step size of 50, along with the following parameters $T_{EPS} = 16000\,\mathrm{N}$, $v = 90\,\mathrm{m\,s^{-1}}$ and $\rho = 1.225\ \mathrm{kg\,m^{-3}}$, the possible operating points, and the optimal operating point resulting in the maximum propeller efficiency are visualized in Figure(4.34).



[redacted]

[redacted]

[redacted]

Figure 4.34: Visualization of valid operating points and the optimal operating point resulting the maximum efficiency in propeller performance map for η_{prop}.

While each operating points has its own interpolated η_{prop} map, all the operating points are plotted on the same η_{prop} map corresponding to the optimal operating point for visualization purposes. The shaded volume is the valid region for the propeller to operate for the given v, ρ, a and T_{EPS}.

By employing the algorithm described above, three lookup tables were generated for τ, ω and η with brake-points encompassing the conditions of the ES-30 mission. Further, factors such as TAS ranges, and operation ceiling for the ES-30 were considered to specify the ranges for the brake-points. However, for simplification, these tables only cater to positive thrust requests, while any negative thrust requests are neglected. The Simulink implementation of the propeller model, incorporating these lookup tables is shown in Figure(4.35).

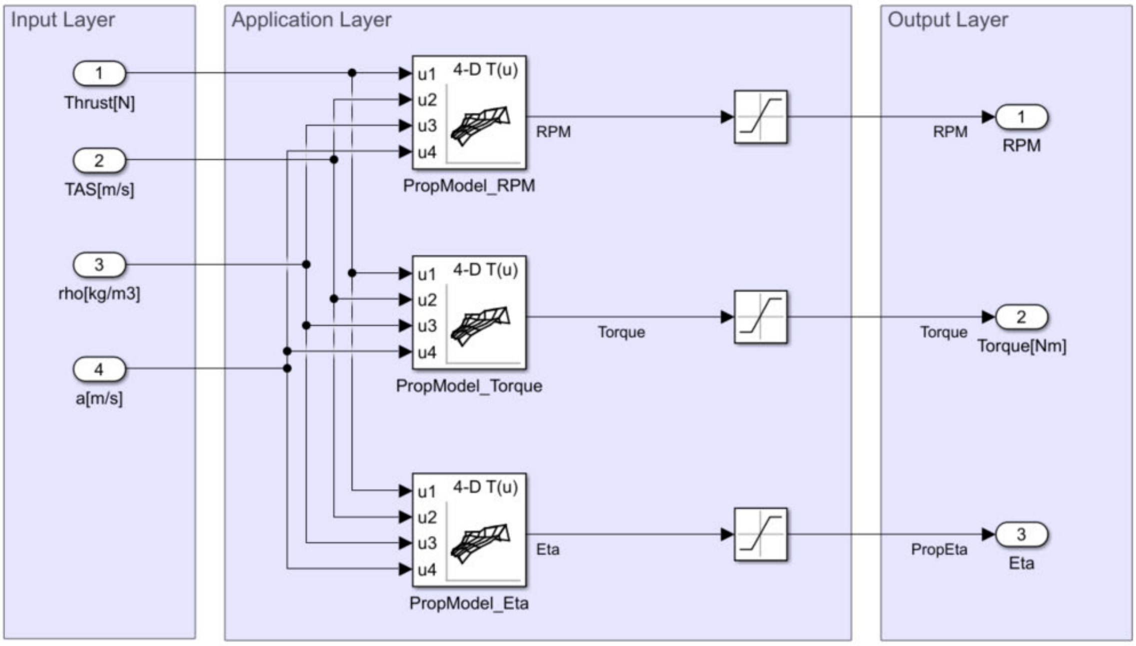
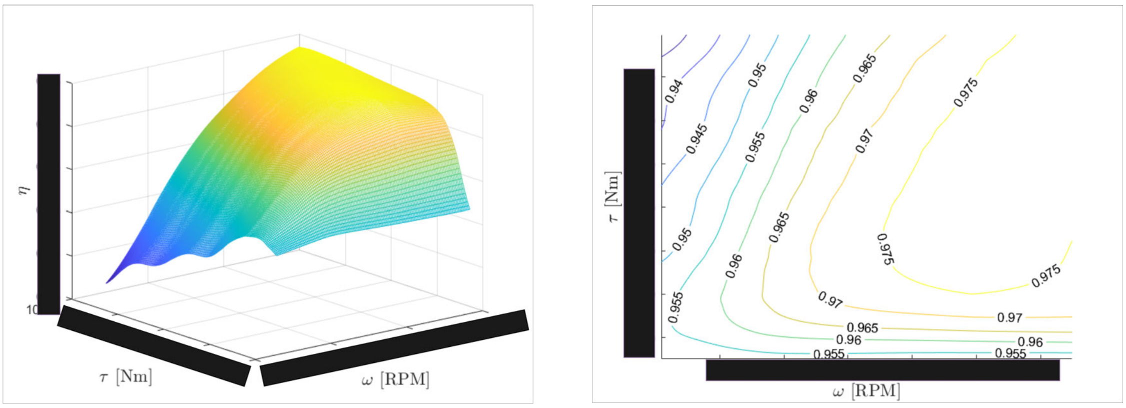


Figure 4.35: Simulink implementation of EPS propeller model.

4.1.6.2 PMSM Model

The PMSM model within the electrical propulsion system Model of the simulation model represents the electric motor used in the electrical propulsion system of hybrid aircraft. As discussed in the theory section, there are many approaches for modeling motors. Given the primary focus of this simulation model is the energy consumption of the electrical propulsion system, a data-driven modeling approach using motor efficiency map was adopted for modeling the PMSM model. Specifically, a motor efficiency map provided by Heart Aerospace for a PMSM was utilized for the PMSM model. The map contain efficiency η values against shaft torque τ and shaft rotational speed ω. The digitization of the map was followed by pre-processing steps to eliminate noise from the digitized efficiency map. The efficiency map obtained after these steps is shown in Figure(4.36).


[redacted]
[redacted]
[redacted]
[redacted]
[redacted]

(a) PMSM efficiency mesh plot **(b)** PMSM efficiency contour plot

Figure 4.36: PMSM efficiency maps

Modeling of PMSM was based on the assumption that all the losses occur when converting the electrical energy to the mechanical energy by PMSM is represented by the efficiency map. Further, any other losses of the PMSM including thermal losses were neglected.

The input and output signals of the PMSM Model are illustrated in Figure(4.37).

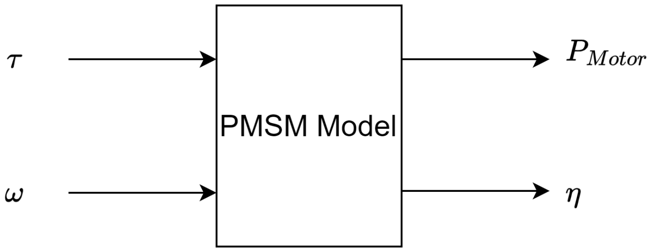


Figure 4.37: Input and output signals of PMSM Model.

The input signals τ and ω are the torque and shaft speed signal outputs from Propeller Model. The output signals P_{Motor} and η are the signals for power and motor efficiency respectively.

The PMSM efficiency is the ratio between output shaft power P_{Shaft} and the power drawn by the PMSM P_{Motor},

$$\eta = \frac{P_{Shaft}}{P_{Motor}} \tag{4.19}$$

The efficiency map of the PMSM was integrated into a lookup table and implemented within the PMSM Model in Simulink. Given input values of τ and ω, the efficiency value η is evaluated from the lookup table based on the PMSM efficiency map. Subsequently, using Equation(4.19), the P_{Motor} value is determined. The Simulink implementation of the PSMS Model is shown in Figure(4.38).

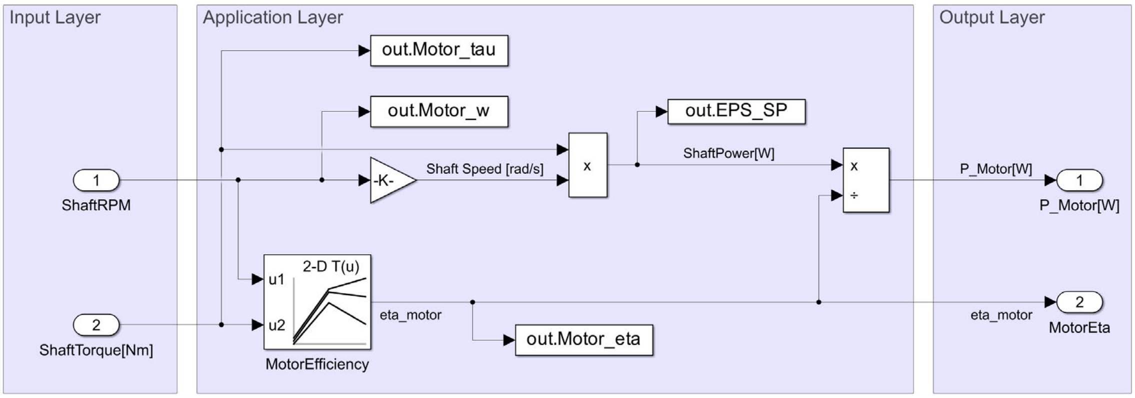


Figure 4.38: Simulink implementation of the PMSM Model

4.1.6.3 Inverter-loss Model

The EPS in ES-30 aircraft consists of components other than the PSMS such as inverter and motor drivers related power electronics which are also important for the operation of the EPS. The losses of these units are represented by the Inverter-Loss Model in the EPS Model.

The input and output signals for the Inverter-Loss Model is shown in Figure(4.39).

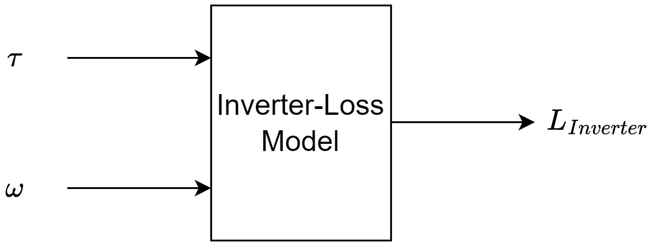


Figure 4.39: Input and output signals of Inverter-Loss Model

The input signals τ and ω are the output signals from the Propeller Model. The Inverter-Loss Model calculates and provide the power loss values for the given input signal values.

The inverter loss considered in this model is assumed to be a function of ω and τ with a fixed loss component. The relationship between inverter-loss $L_{Inverter}$ to ω and τ is assumed to be:

$$L_{Inverter} = L_0 + k_\omega\,\omega + k_\tau\,\tau \tag{4.20}$$

where, $L_0 = 1000\,\mathrm{W}$, $k_\omega = 6.7\ \mathrm{W\,/\,rpm}$ and $k_\tau = 1\,\mathrm{W\,/\,N\,m}$.

Using the Equation(4.20), the Inverter-Loss Model was implemented in Simulink with a MATLAB function as shown in Figure(4.40).

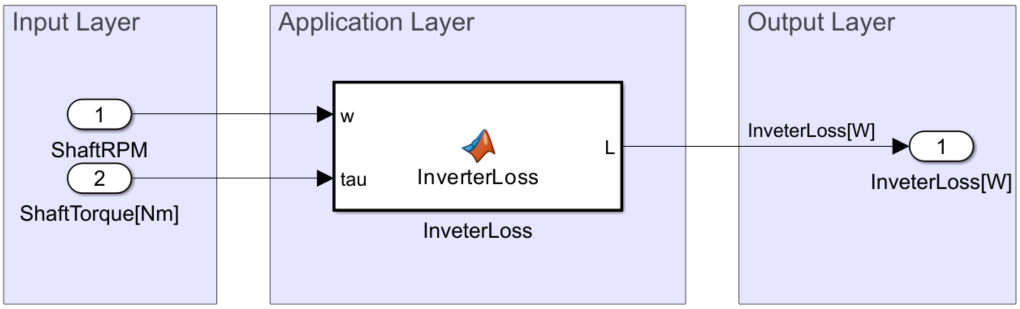


Figure 4.40: Input and output signals of Inverter-Loss Model

The total power demand by the EPS Model denoted as P_{Batt} is obtained by summing the power drawn by the motor P_{Motor} and the inverter loss $L_{Inverter}$.

$$P_{Batt} = P_{Motor} + L_{Inverter} \tag{4.21}$$

The EPS Model was obtained by integrating the EPS Propeller Model, PMSM Model and Inverter-Loss Model as depicted in the Figure(4.41).

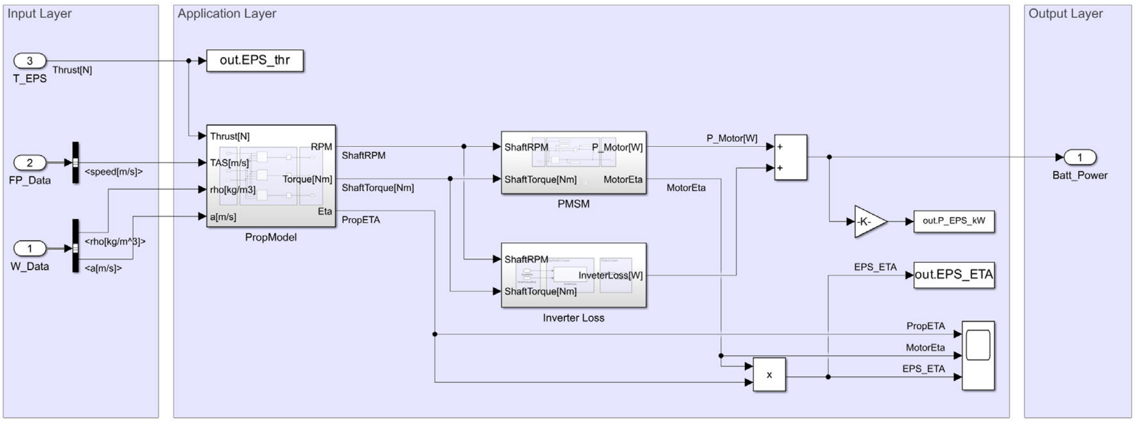


Figure 4.41: Simulink implementation of the EPS Model

4.1.7 TES Model

The TES Model within the simulation model represents the turboprop engine systems in hybrid aircraft. The function of this model is to determine the amount of fuel consumed when delivering thrust requested by the master controller. The input and output signals of the TES Model are shown in Figure(4.42).

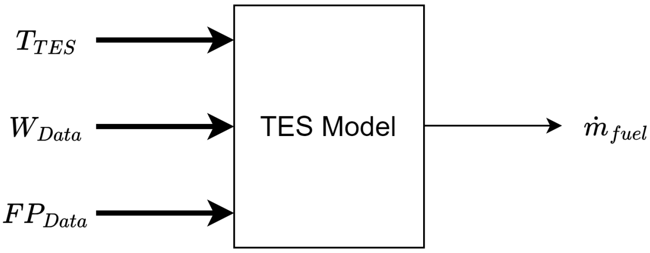


Figure 4.42: Input and output signals of TES Model

The inputs to the TES Model are T_{TES}, W_{Data} and FP_{Data} signal-buses, the outputs from the Master Controller Model, Weather Model and Mission Planner Model respectively.

The input layer of the TES Model retrieve the signals ρ, h_p and a from the W_{Data} signal-bus, v from FP_{Data} signal-bus and T_{TES} and TP_{status} signals from the T_{TES} signal-bus and these extracted signals are sent to the application layer.

The application layer of the TES Model consists of two main sub-models and they are,

1. TES Propeller Model
2. Power Specific Fuel Consumption(PSFC) Model

The methodology followed for modelling these two sub-models are explained below.

4.1.7.1 TES Propeller Model

The approach followed in modeling the TES propeller model is similar to the approach in modeling the EPS propeller model. The propeller optimization algorithm explained in 4.1.6.1 along with the diameter size of the proposed TES propeller was used to generate the data tables. These data tables were utilized in the lookup table based TES propeller model. The inputs and outputs for the TES Propeller Model are shown in Figure(4.43).

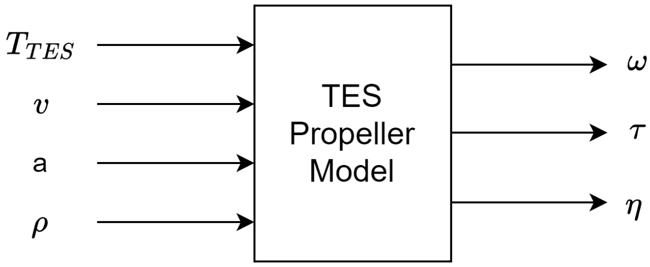


Figure 4.43: Inputs and outputs of the TES Propeller Model

The TES Propeller Model takes T_{TES}, v, a and ρ as inputs. The propeller model evaluates and outputs the propeller speed ω and shaft torque τ along with the

propeller efficiency η to deliver the requested T_{TES} considering the TAS and ambient parameters from predefined lookup tables.
The Simulink implementation of the TES Propeller Model is shown in Figure(4.44).

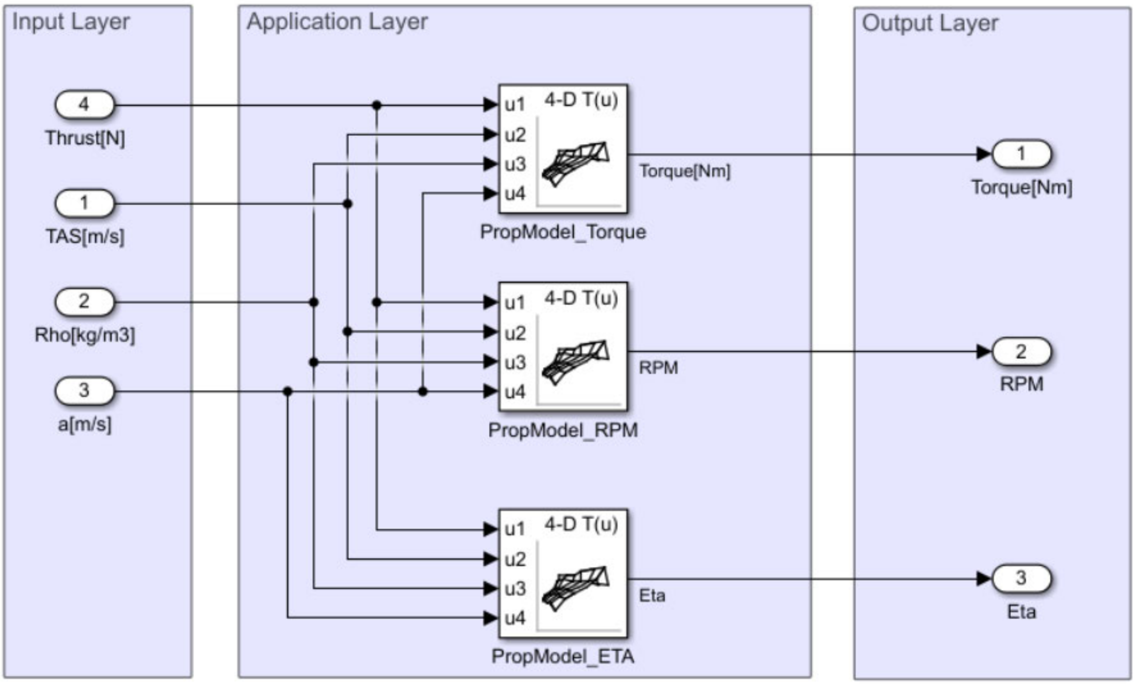


Figure 4.44: Simulink implementation of the TES Propeller Model

4.1.7.2 PSFC Model

The function of the PSFC Model within the TES Model is to determine the fuel consumption of TES to deliver the power demand specified by the ω and τ request from the TES Propeller Model.

The inputs and output of the PSFC Model is illustrated in Figure(4.45).

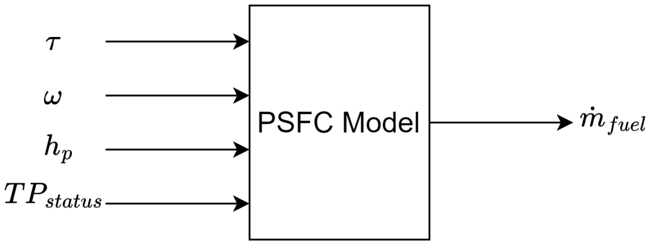


Figure 4.45: Inputs and output of PSFC Model

τ and ω are output signals from the TES Propeller Model. h_p and TP_{status} signals are the pressure altitude and turboprop status signals respectively. The output from the PSFC Model is the fuel flow rate $\dot{m}_{fuel}$ signal.

A specific fuel consumption map of a turbine engine provided by the Heart Aerospace was the basis of the PSFC Model. The fuel consumption map contains specific fuel consumption in kg / kWh as a function of pressure altitude h_p and shaft power P_{Shaft}. The Figure(4.46) shows the specific fuel consumption map.

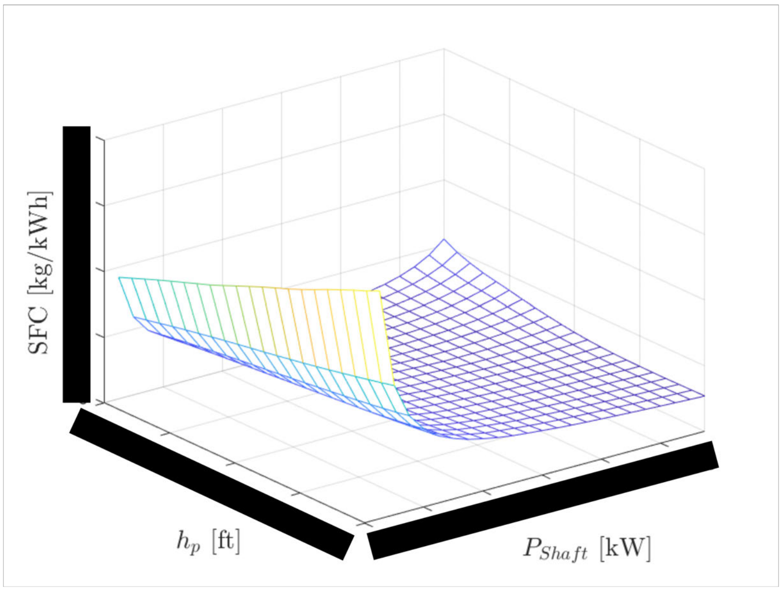


Figure 4.46: PSFC map in kg / kWh against pressure altitude h_p and shaft power P_{Shaft}.

As evident from the Figure(4.46), the map exhibits instability for low power values, resulting in unstable outcomes when implemented in its original format. To address this issue, the PSFC map was converted to units of kg / s by multiplying each grid-point values with its corresponding power value. The resulting PSFC map in kg / s after this conversion is shown in Figure(4.47).

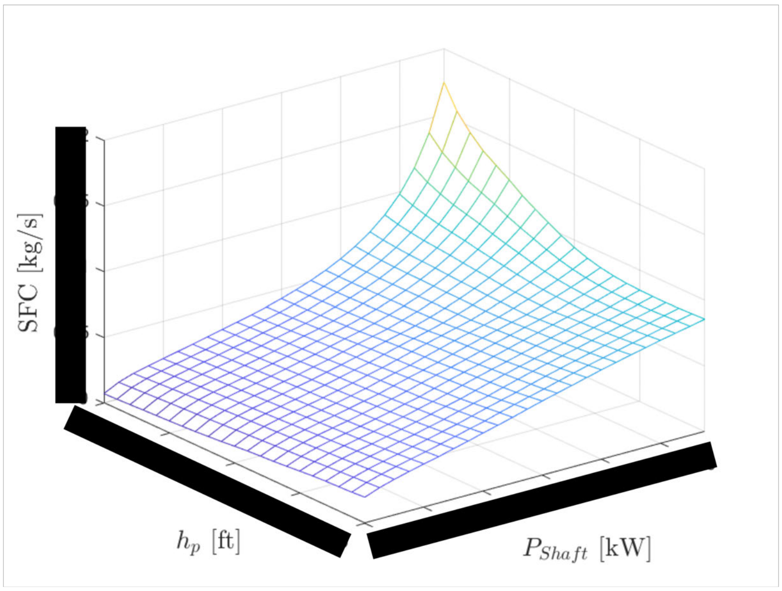


Figure 4.47: SFC in kg/s against pressure altitude h_p and shaft power P_{Shaft}.

An issue with the new specific fuel consumption map in kg / s is that there is non-zero value corresponding to zero power. This results in a behaviour that the TES consumes fuel even if the requested power from the TES is zero. To mitigate this issue, the turboprop status signal TP_{status} is being used as a reference to indicate the TES status where,

$$TP_{status} = 0 \;\rightarrow\; \text{TES is OFF}$$
$$TP_{status} = 1 \;\rightarrow\; \text{TES is ON}$$

This new specific fuel consumption map is implemented in the PSFC Model as a lookup table. For the given breakpoints of h_p and P_{Shaft}, the lookup table determines the fuel flow-rate $\dot{m}_{fuel}$. This value is multiplied by the TP_{status} signal value to get the $\dot{m}_{fuel}$ output signal. The Simulink implementation for the PSFC map is shown in Figure(4.48).

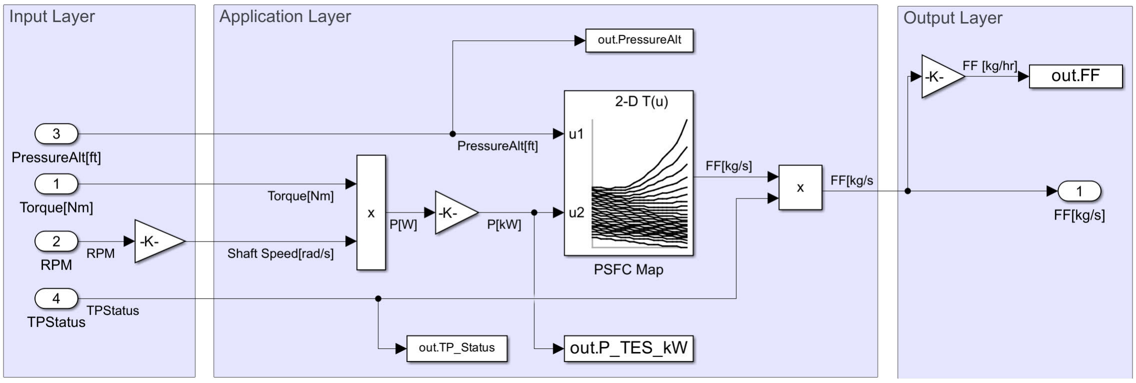


Figure 4.48: Simulink implementation of the PSFC Model.

The TES Propeller Model and the PSFC Model were integrated together to create the TES Model shown in Figure(4.49). The output layer of the TES Model outputs the fuel flow-rate $\dot{m}_{fuel}$ in kg / s as the output signal.

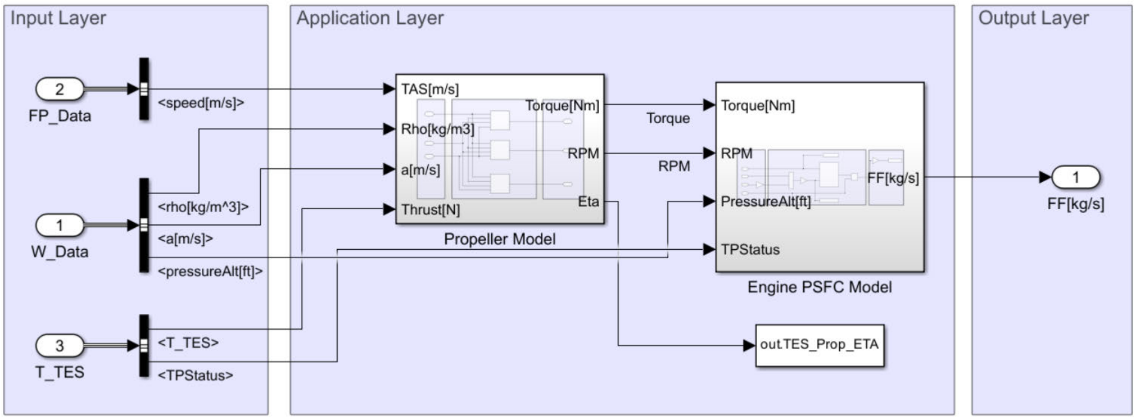


Figure 4.49: Simulink implementation of the TES Model.

4.1.8 Battery Model

The Battery Model in the simulation model represent the battery system which delivers the electrical poser for the Electric Propulsion System in hybrid aircraft.

The Battery Model takes the input parameters listed in Table(4.6) to configure the the Battery Model settings.

Table 4.6: Input Parameters for the PBU Model

Input Parameters	**Unit**
Number of parallel cells	Nos
Number of series cells	Nos
Cell capacity	Ah
Max Cutoff voltage	V
Min Cutoff voltage	V

Additional to the parameters in the Table(4.6), Battery Model also uses the Initial Battery SoC ($\mathrm{SOC_{init}}$) parameter, specified by the user in the Aircraft Model mask. The input and output signals of the Battery Model are shown in Figure(4.50).

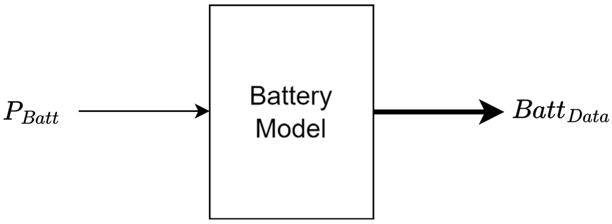


Figure 4.50: Input and output of Battery Model.

The signal P_{Batt} is the output signal of EPS Model which contains the value of electrical power drawn by the EPS from the PBU. $Batt_{Data}$ is a signal-bus containing PBU energy consumption related parameters which are later being used to visualize simulation results.

The input layer of the PBU Model receives the P_{Batt} signal and the power demand for an single PBU unit is determined. The application layer of the PBU model utilizes a battery model provided by Heart Aerospace. This model was adapted to fit the inverse simulation approach. This battery model was an Equivalent Circuit Model(ECM) implementation based on [43] and the ECM consists of 3 RC pairs and 1 IR pair along with an additional resistor to account for the internal resistance of the battery cell. To capture the non-linear behaviour of Open Circuit Voltage (OCV) with varying cell SoC, the model consists of a lookup table based on empirical data. The above described non-linear relationship between OCV and SoC for a typical battery cell is illustrated in Figure(4.51). The specific battery cell values are then scaled based on the number of parallel and series cells, as well as the cell capacity specified by the user within the Battery Model.

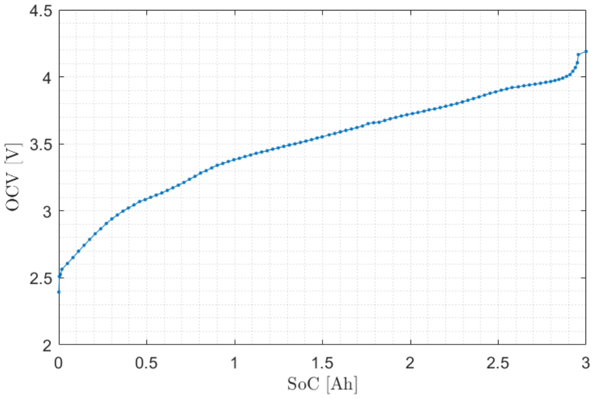


Figure 4.51: Variation of Open Circuit Voltage (OCV) of battery cell with varying SoC.

The Simulink implementation of the PBU Model is shown in Figure(4.52).

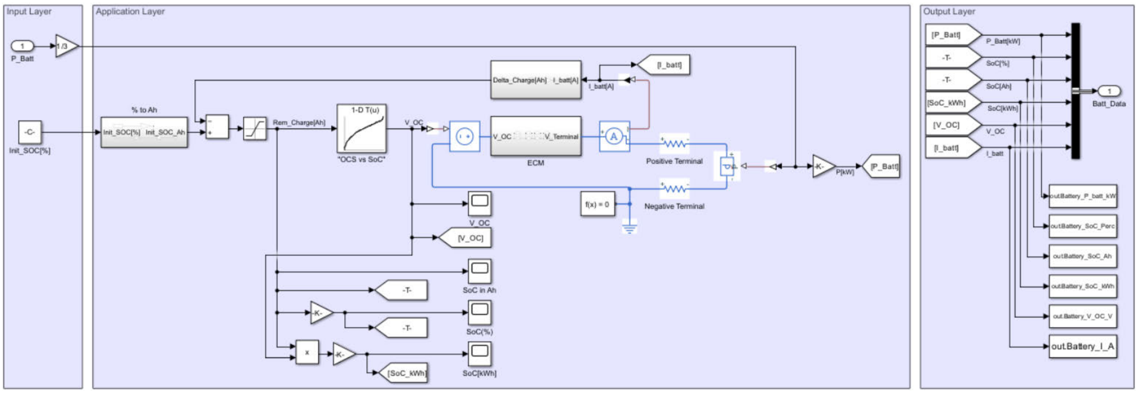


Figure 4.52: Simulink implementation of the PBU Model.

The output layer of the PBU Model creates the Batt_{Data} signal-bus including the necessary battery parameters and energy consumption related signals.

4.2 Verification

Model verification plays a crucial role in ensuring the accuracy and the reliability of the simulation results. As the simulation model comprises various interconnected models with distinct functions, different approaches were employed to verification each model. Further, the verification process involved subject experts from the different departments. The typical mission data provided by the company served as the baseline data for the verification process. The verification process for the different models in the simulation model are described below.

4.2.1 Verification of Mission Planner

Given that the primary function of the mission planner is to generate flightpath signals for user-specified missions, the verification process involved two main steps: a parameter consistency check and a comparison against the baseline mission profile.

The first step of the parameter consistency check involved generating different mission profiles and verifying that the user-specified parameter values were correctly reflected in the generated missions. This process of verification that all mission profile-related parameter values directly specified by the user (summarized in Table (4.1)) were accurately utilized and reflected in the mission profile. Various missions were created and tests were conducted varying each of these parameters to ensure that the mission profile correctly captures and utilizes them in the generated mission profiles. Additionally, the parameters calculated from the user-specified inputs in the mission profile were cross-checked involving both manual calculations and MATLAB script-based method and confirmed that the model consistently generates the expected results.

The comparison against the baseline mission profile was conducted using the mission profile related data from the baseline mission. The baseline mission for hybrid aircraft was created within the mission planner model and then the simulation output was compared against the baseline mission data. Specifically, the altitude, aircraft speed, and flight time were considered for this comparison. Figure (4.53) shows the altitude and CAS from the mission planner plotted alongside the baseline mission data. The results demonstrate that the mission profile generated by the mission planner in the simulation model closely align with the baseline mission.

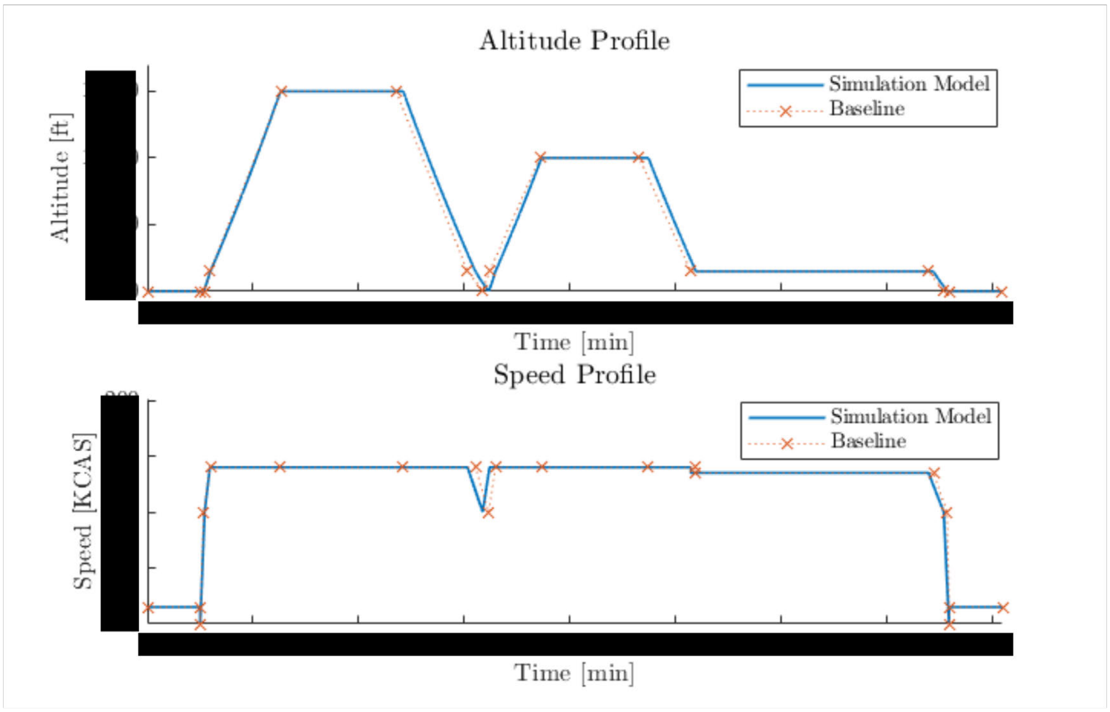


Figure 4.53: Comparison of mission profile from mission planner model with baseline data for hybrid aircraft mission.

4.2.2 Verification of Weather Model

The weather model within the simulation model facilitates atmospheric conditions for both ISA conditions and deviations from ISA conditions. The weather model was verification considering both scenarios, across an altitude range from sea level up the operational ceiling of the typical regional aircraft.

For ISA conditions, the ambient parameter output values from the weather model including OAT, Static Pressure, Air Density, and Speed of Sound were compared with the corresponding values in the ISA table [23]. Figure(4.54) depicts the verification results for these parameters. Notably, the parameter values from the weather model closely align with the ISA table values.

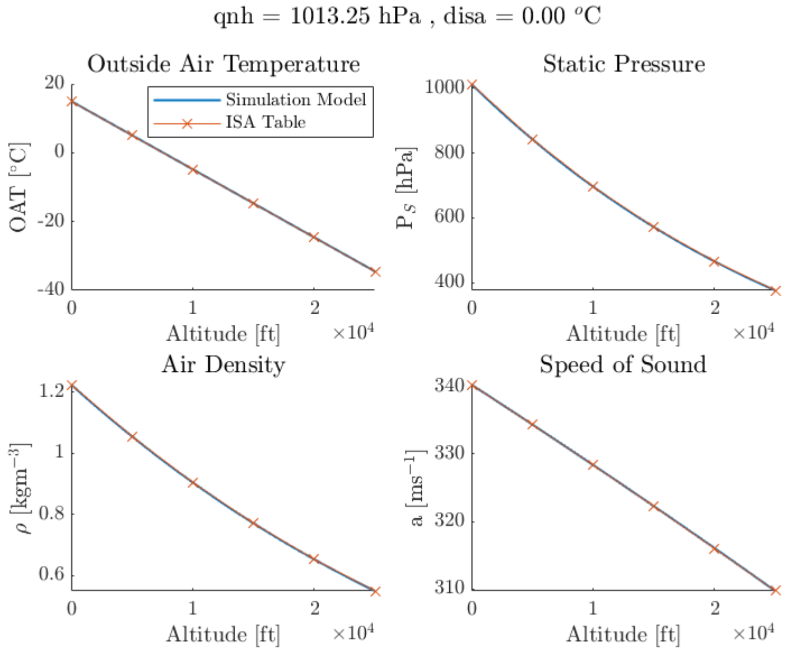


Figure 4.54: Comparison of ambient parameter values from weather model for ISA condition with corresponding values from ISA table.

For temperature deviation from ISA conditions, the ambient parameter values from the weather model were generated varying the temperature DISA value $-50\,^{o}$C to $+35\,^{o}$C, while keeping the QNH value at 1013.25 hPa. The generated results from weather model were compared with the corresponding values obtained from the AeroToolbox [44], considering the altitude range of a typical regional aircraft. The verification result for DISA=$+10\,^{o}$C depicted in Figure(4.55).

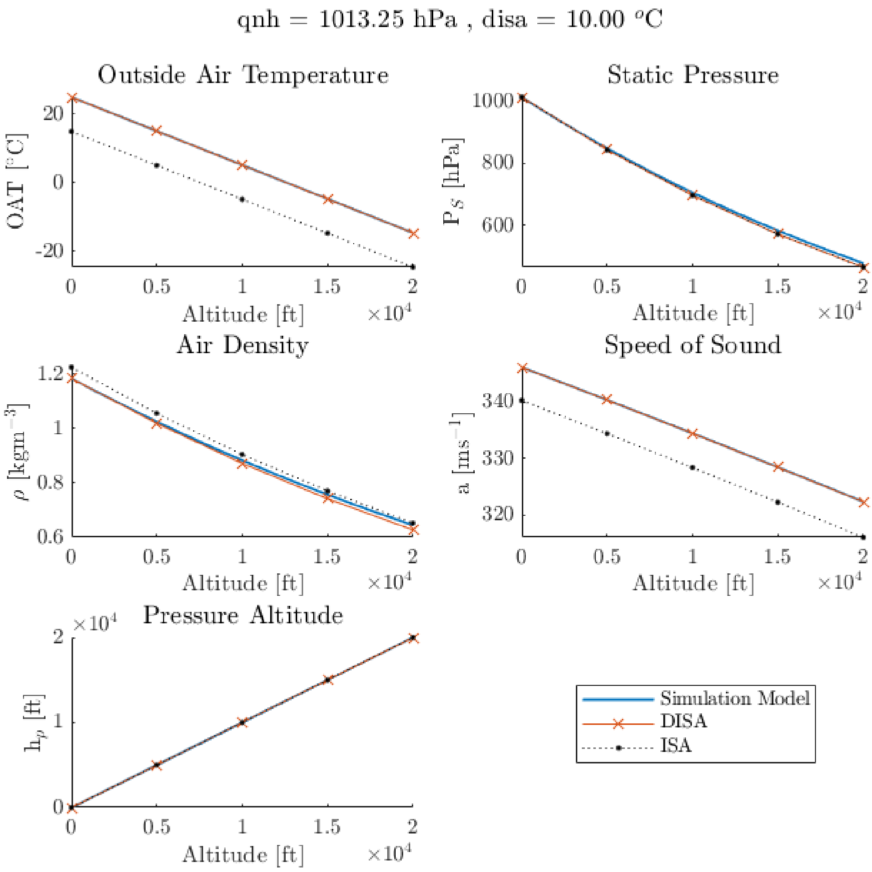


Figure 4.55: Comparison of ambient parameter values from weather model for temperature DISA = +10 ^{0}C condition with corresponding values from baseline data.

For pressure deviation from ISA conditions, the ambient parameter values from the weather model were generated for different the QNH values while keeping the temperature at ISA value. The ambient parameters including the Pressure Altitude values were compared with the obtained values from AeroToolBox [44] and E6B flight calculator [45]. The verification result for QNH =1000 hPa is depicted in Figure(4.56).

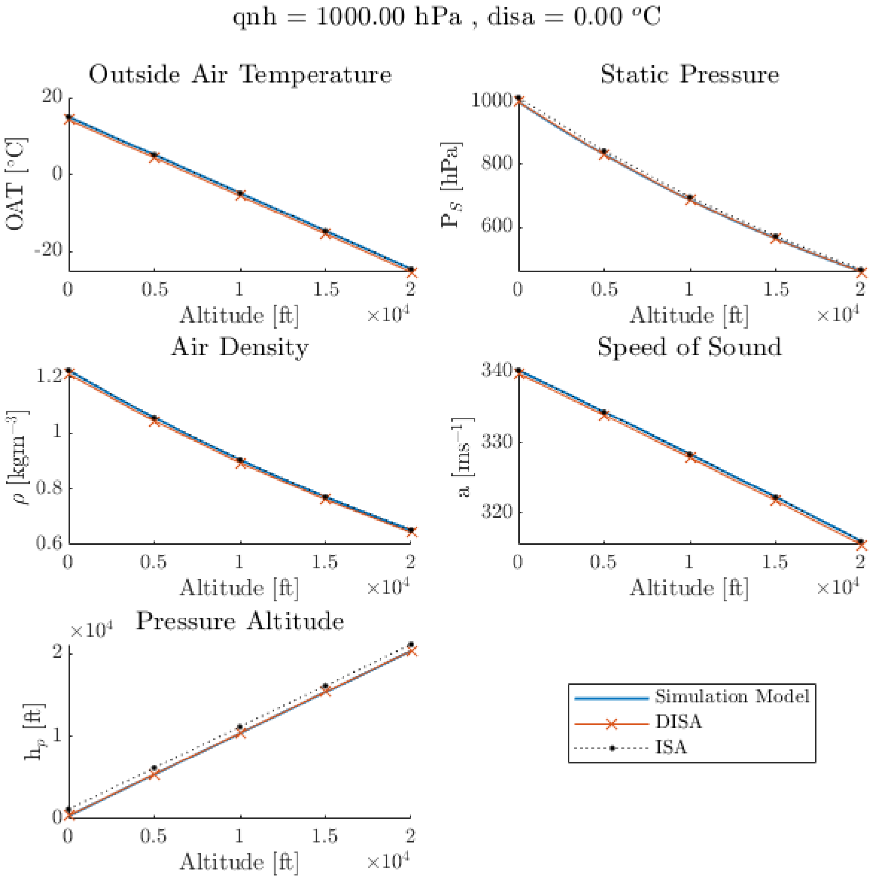


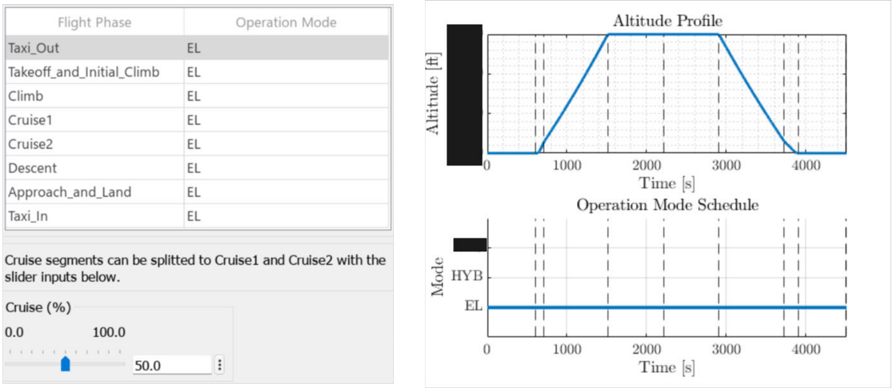
Figure 4.56: Comparison of ambient parameter values from weather model for pressure deviation from ISA condition (QNH = 1000 hPa) with corresponding values from baseline data.

The verification results both ISA and DISA conditions demonstrate that the weather model captures the ambient parameters adhering closely to their respective baseline references for both scenarios.

4.2.3 Verification of Mode Selector

The verification of the mode selector aimed to ensure that user-specified operation modes were correctly assigned to the corresponding segments of a flight profile. Different modes (EL, HYB, and HBS) were assigned to various segments within the mission profile through the mask of the mode selector, and the outcomes from the mode selector was examined.

Scenario 1: Each segment of the mission is assigned EL mode and the resulted operation mode schedule is EL throughout the mission as shown in Figure(4.57).



Flight Phase	Operation Mode
Taxi_Out	EL
Takeoff_and_Initial_Climb	EL
Climb	EL
Cruise1	EL
Cruise2	EL
Descent	EL
Approach_and_Land	EL
Taxi_In	EL

Cruise segments can be splitted to Cruise1 and Cruise2 with the slider inputs below.

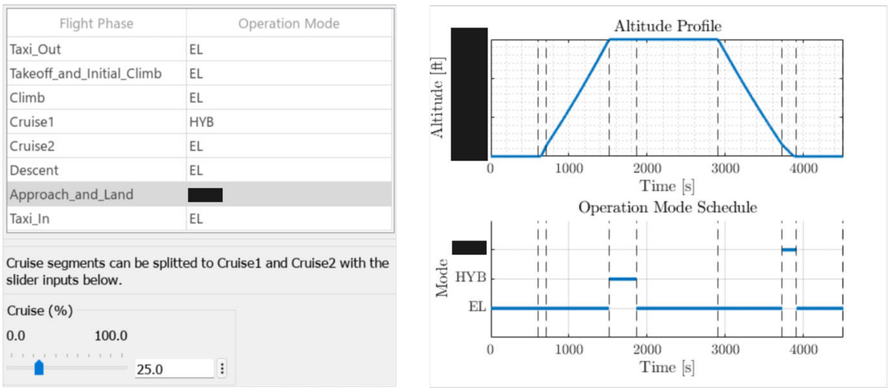
Cruise (%)

0.0 100.0

50.0

Figure 4.57: Mode selector verification results for Scenario 1.

Scenario 2: In this scenario, first 50% of cruise (Cruise1) is assigned HYB mode and the remaining cruise segment (Cruise2) is assigned EL mode while the Approach and Land is assigned HBS mode. The mode assignment in the mode selector mask and the resulted operation mode schedule is shown in Figure(4.58).



Flight Phase	Operation Mode
Taxi_Out	EL
Takeoff_and_Initial_Climb	EL
Climb	EL
Cruise1	HYB
Cruise2	EL
Descent	EL
Approach_and_Land	[redacted]
Taxi_In	EL

Cruise segments can be splitted to Cruise1 and Cruise2 with the slider inputs below.

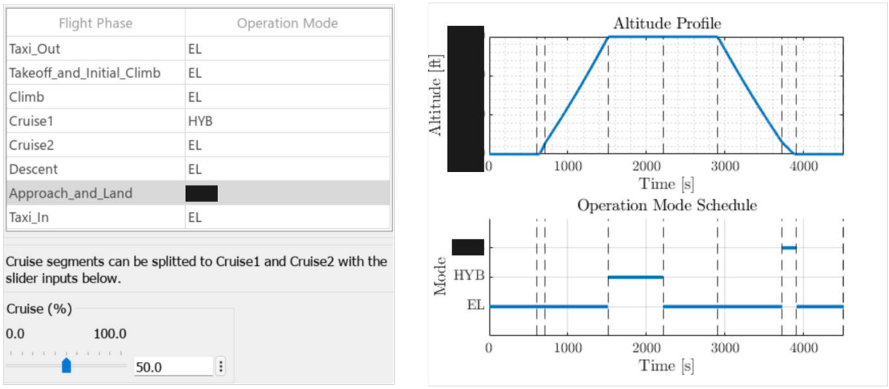
Cruise (%)

0.0 100.0

25.0

Figure 4.58: Mode selector verification results for Scenario 2.

Scenario 3: In this scenario, the modes are assigned similar to the Scenario 2, but the cruise segment is splitted to 25:75 ratio between Cruise1 and Cruise2. The output result for this mode assignment is shown in Figure(4.59).



Flight Phase	Operation Mode
Taxi_Out	EL
Takeoff_and_Initial_Climb	EL
Climb	EL
Cruise1	HYB
Cruise2	EL
Descent	EL
Approach_and_Land	[redacted]
Taxi_In	EL

Cruise segments can be splitted to Cruise1 and Cruise2 with the slider inputs below.

Cruise (%)

0.0 100.0

50.0

Figure 4.59: Mode selector verification results for Scenario 3.

4.2.4 Verification of Aircraft Model

The verification of the aircraft model involved two-step approach to ensure its accuracy and reliability. Initially, the aircraft model is simulated with a mission profile, and the thrust output from the model for different segments of the mission profile was compared with manually calculated values. This step verified the proper handling of equations within the model and its ability to correctly calculate the required thrust.

In the second step, comparison between the results generated by the aircraft model and baseline data was performed. Specifically, the lift-to-drag ratio (C_L/C_D) for varying aircraft speed given by the aircraft model was compared against corresponding baseline data. The comparison, illustrated in Figure (4.60), demonstrates that the results from the model closely aligns with the baseline.

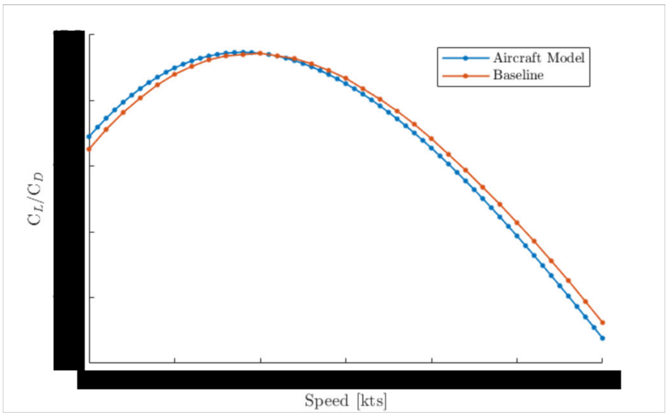


Figure 4.60: Comparison of lift-to-drag ratio (C_L/C_D) for varying aircraft speed from aircraft model and baseline data.

4.2.5 Verification of Electrical Propulsion System Model

The verification process for the Electrical Propulsion System model involved testing of EPS propeller model and PMSM model within the Electrical Propulsion System model. The propeller model utilizes lookup tables to determine the torque, speed and the efficiency for a given thrust request based on flightpath parameters and atmospheric conditions. Various missions were simulated varying the parameters and the resulting propeller efficiency values were compared with the propeller efficiency values in the baseline data corresponding to different phases of flight. The comparison of propeller model efficiency and the efficiency values from baseline mission data are plotted in the Figure(4.61).

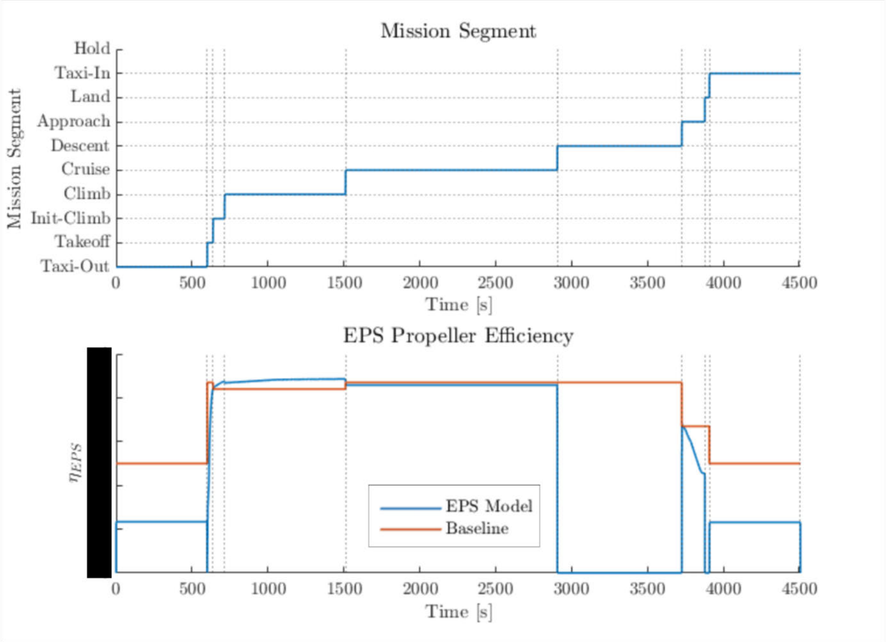


Figure 4.61: Comparison of propeller efficiency of Electrical Propulsion System model and and baseline data.

The EPS propeller model determines the efficiency of the propeller considering flightpath parameters and ambient parameters while the baseline data considers constant propeller efficiencies for different phases of the mission. In descent segment, the propeller efficiency from the model is zero as a result of neglecting negative thrust request and no positive work is done by the propeller in this segment.

The PMSM model was tested for different combinations of shaft speed and torque values to ensure that lookup table based implementation for the PMSM model properly handles and output the expected outputs. Notably the simple motor efficiency map used in the model does not cover the torque value range requested by the propeller model and this is handled by the lookup table by extrapolating the available data. The model was thoroughly tested for these scenarios to ensure that the extrapolation from the lookup tables provide valid and the expected outputs.

4.2.6 Verification of TES

The TES model verification involved testing both TES propeller model and the PSFC model within the TES model. The TES propeller model verification step followed the similar approach in verification the EPS propeller data and also utilized the same baseline data. The PSFC model was tested for different combinations of pressure altitude and shaft power values and different modes of operation, to verify that the model handles the inputs properly and output valid and reasonable results, specifically for scenarios with extrapolation of SFC map.

4.3 Dynamic Programming Based Optimal Energy Management Strategy

The energy management problem discussed in this work is to determine the optimal operation mode schedule for completing a given mission without depleting the energy reserves of the PBU, while minimizing fuel consumption. This problem can be characterized as a non-linear, time-variant, constrained multi-stage decision-making problem. Dynamic Programming (DP) is employed to solve this problem and determine the optimal energy management strategy for given mission.

The optimization problem is formulated such that, the turbine engine status is the decision variable, while the PBU SoC and remaining FoB states are the dependent variables for the optimization problem. The objective function considers both the fuel consumption and the state changes of the turboprop engine. As the optimization variable is the turboprop engine status, the optimization algorithm optimizes turboprop engine state to minimize the objective function value, ensuring that the state and control constraints are satisfied.

In this work, an open-source DP solver developed by ETH Zurich, implemented in MATLAB [46] was utilized to solve the optimization problem. The steps involved in formulating the energy management problem and adapting it to suit the DP solver are explained below.

States:
For the optimization problem, two system states are considered and they are the PBU SoC and the remaining fuel-on-board (FoB). These two states are dependent variables in this problem.

$$x_k^{soc} = \text{SoC} \tag{4.22}$$

$$x_k^{fob} = \text{FoB} \tag{4.23}$$

Control Input:
The control input in this problem is the turbine engine ON/OFF status.

$$u_k = \begin{cases} 1 \text{ ; Turbine engine ON} \\ 0 \text{ ; Turbine engine OFF} \end{cases} \tag{4.24}$$

Constraints:
The system constraints encompass both control and state constraints. The state

and control trajectories must remain within the admissible sets defined by these constraints.

$$u_k \in \{0, 1\} \tag{4.25}$$

$$x_k^{soc} \in [\text{SoC}_{\text{min}}\ ,\ \text{SoC}_{\text{max}}] \tag{4.26}$$

$$x_k^{fob} \in [\text{FoB}_{\text{min}}\ ,\ \text{FoB}_{\text{max}}] \tag{4.27}$$

Initial States:
The initial states of the problem are the initial SoC and FoB levels onboard the aircraft.

$$x_0^{soc} = \text{SoC}_{\text{init}} \tag{4.28}$$

$$x_0^{fob} = \text{FoB}_{\text{init}} \tag{4.29}$$

Terminal Constraints:
The optimization goal is to ensure that the PBU SoC level reaches or remains above the target final SoC value while ensuring minimum fuel consumption. To achieve this, soft constraints on both the SoC and FoB are introduced.

$$x_N^{soc} \in [\text{SoC}_{\text{final,min}}\ ,\ \text{SoC}_{\text{final,max}}] \tag{4.30}$$

$$x_N^{fob} \in [\text{FoB}_{\text{final,min}}\ ,\ \text{FoB}_{\text{final,max}}] \tag{4.31}$$

Objective Function:
The objective function for the optimization problem considers the total fuel consumption by the TES. Additionally, the control state changes are also considered, which correspond to turning the TES ON or OFF. Since turbine engines require time to become operational after starting, instantaneous switching is not feasible. To address this, the control state changes are penalized in the objective function. Consequently, the optimization aims to minimize the fuel consumption and also accounts and aims to minimize the control state changes. The resulting objective function is as follows,

$$J_k = \sum_{k=0}^{N-1} \Delta m_{fuel,k} + k_p \cdot \Delta u_{k,k-1} \tag{4.32}$$

where $\Delta m_{fuel,k}$ is the fuel mass consumed by TES in timestep k, $\Delta u_{k,k-1}$ is the control state change and k_p is the penalizing factor for the control state change.

Discretization:
The DP solver creates the state-space for the optimization problem based on the following parameters.

Discritization for control input = N_U
Discritization for SoC state = N_{soc}
Discritization for FoB state = N_{fob}
Sample time = Ts

The system models, including those for the aircraft, EPS, TES, and PBU were developed as MATLAB-based functions and scripts. Some simplifications had to be performed when converting the Simulink based complex models to MATLAB scripts to fit the DP solver.

The optimization problem was formulated in MATLAB to ensure a seamless workflow between the simulation model and the Dynamic Programming (DP) based optimization algorithm. The mission profile generated from the simulation model serves as an input for the optimization algorithm. The optimal control strategy, obtained as an output from the DP-based optimization algorithm, can be integrated into the HPACS model as a Simulink variant model. This integration serves two purposes. First, this enables the users to generate missions in the simulation model, run the DP-based optimization to obtain the optimal mode schedule and re-simulate the mission with the optimal mode schedule to assess the performance in a seamless way. Second, comparative analyses can be performed between the simulation model results and the DP-based optimal results conveniently.

5

Result and Discussion

This chapter presents and discuss the outcomes of the project in main three different sections: Simulation Model, Simulation Results and DP-Based Optimal Energy Management Strategy.

Simulation Model: As the simulation model is a key outcome of the project, the overall simulation model and the Simulink mask based GUI are described from the perspective of the user.

Simulation Results: The simulation results from the simulation model for various scenarios are presented and discussed in this section. These include the ES-30 typical mission simulation results, comparisons with baseline mission data, cruise altitude survey results and cruise speed survey results. Moreover, the impact of different weather parameters and the effect of accounting for varying aircraft weight are analysed and discussed.

DP-Based Optimal Energy Management Strategy: This section presents and discusses the results for the proposed DP-based optimal energy management strategy.

5.1 Simulation Model

The pre-flight mission simulation model based in Simulink environment is shown in Figure(5.1).

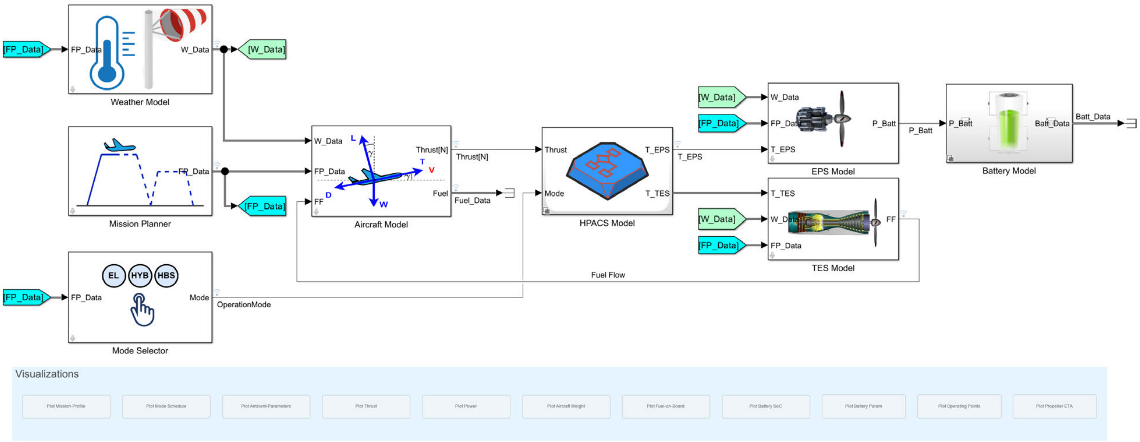


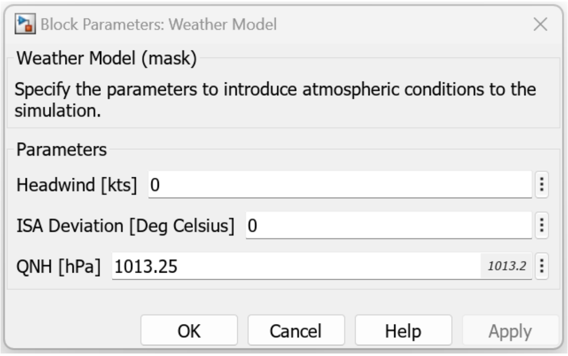
Figure 5.1: Pre-flight mission simulation model.

The user can perform a simulation by configuring the simulation environment in four steps:
Step 1: Set atmospheric conditions in the Weather Model mask.
Step 2: Generate the mission profile in the Mission Planner mask.
Step 3: Select the operation mode schedule in the Mode Selector mask.
Step 4: Specify initial parameters in the Aircraft Model mask.

The different model masks involved in the above four steps are described below.

Weather Model:

The mask of the Weather Model shown in Figure(5.2), serves as the GUI that allows the user to specify atmospheric conditions for the simulation. Within this interface, the user can adjust the values for headwind, temperature DISA, and QNH to configure the atmospheric conditions of the simulation to the desired conditions.


Block Parameters: Weather Model

Weather Model (mask)

Specify the parameters to introduce atmospheric conditions to the simulation.

Parameters

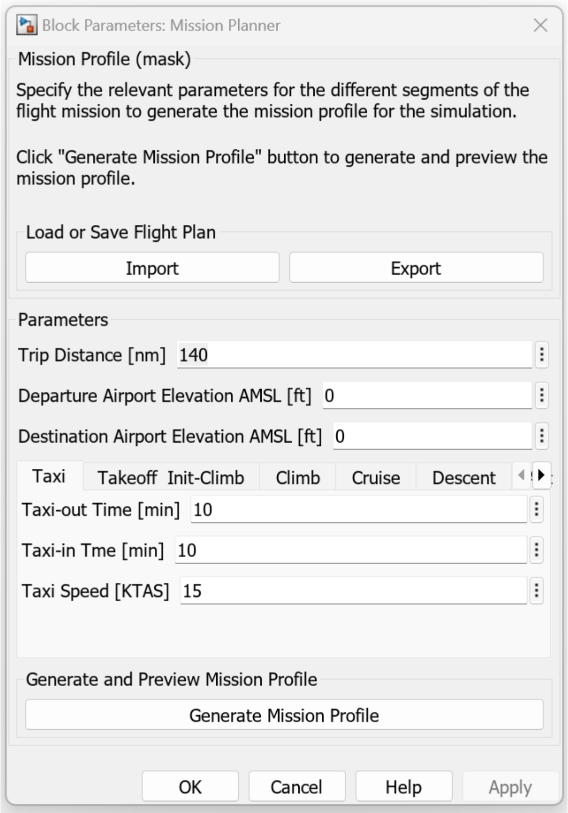
Headwind [kts] 0

ISA Deviation [Deg Celsius] 0

QNH [hPa] 1013.25

Figure 5.2: Weather Model mask which facilitates the user to specify atmospheric conditions for the simulation.

Mission Planner:


Block Parameters: Mission Planner

Mission Profile (mask)

Specify the relevant parameters for the different segments of the flight mission to generate the mission profile for the simulation.

Click "Generate Mission Profile" button to generate and preview the mission profile.

Load or Save Flight Plan

Parameters

Trip Distance [nm] 140

Departure Airport Elevation AMSL [ft] 0

Destination Airport Elevation AMSL [ft] 0

Taxi-out Time [min] 10

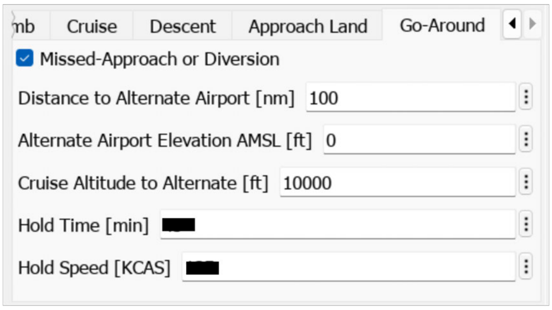
Taxi-in Tme [min] 10

Taxi Speed [KTAS] 15

Generate and Preview Mission Profile

Figure 5.3: Mission Planner mask which facilitates the user to generate mission profiles for the simulation.

The mask of the Mission Planner model shown in Figure(5.3), facilitates the user to create the mission profile for the simulation. The user can generate a mission profile that includes all the flight phases relevant to the regional aircraft typical mission. The GUI facilitates the user for both importing and exporting mission profiles. The user can import predefined mission profiles or export their configured mission profiles as .xlsx files. By using this GUI, the user can conveniently and efficiently configure mission profiles. The parameters for each segment in the mission is arranged in separate containers within the mask for convenience. The user has the ability to define different scenarios for the mission profile including scenario of go-around and flying to the alternate airport with hold, as shown in Figure(5.4).



☑ Missed-Approach or Diversion

Distance to Alternate Airport [nm] 100

Alternate Airport Elevation AMSL [ft] 0

Cruise Altitude to Alternate [ft] 10000

Hold Time [min] [redacted]

Hold Speed [KCAS] [redacted]

Figure 5.4: Enabling and specifying parameters for a go-around scenario.

The user can preview the altitude profile of the generated mission profile by clicking the "Generate Mission Profile" button. The preview result for a generated reference mission is shown in Figure(5.5).

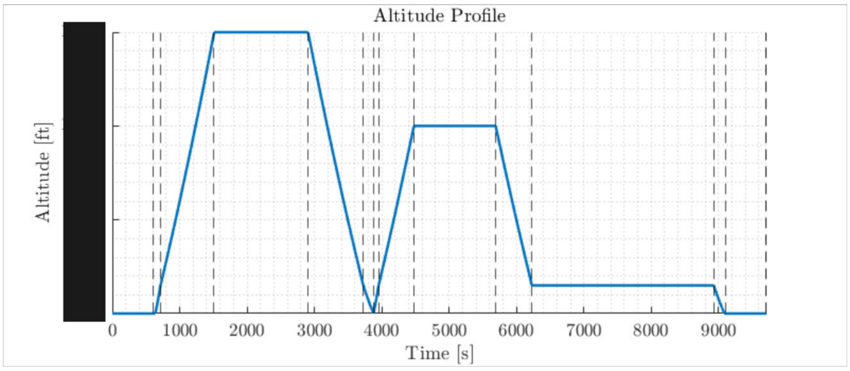
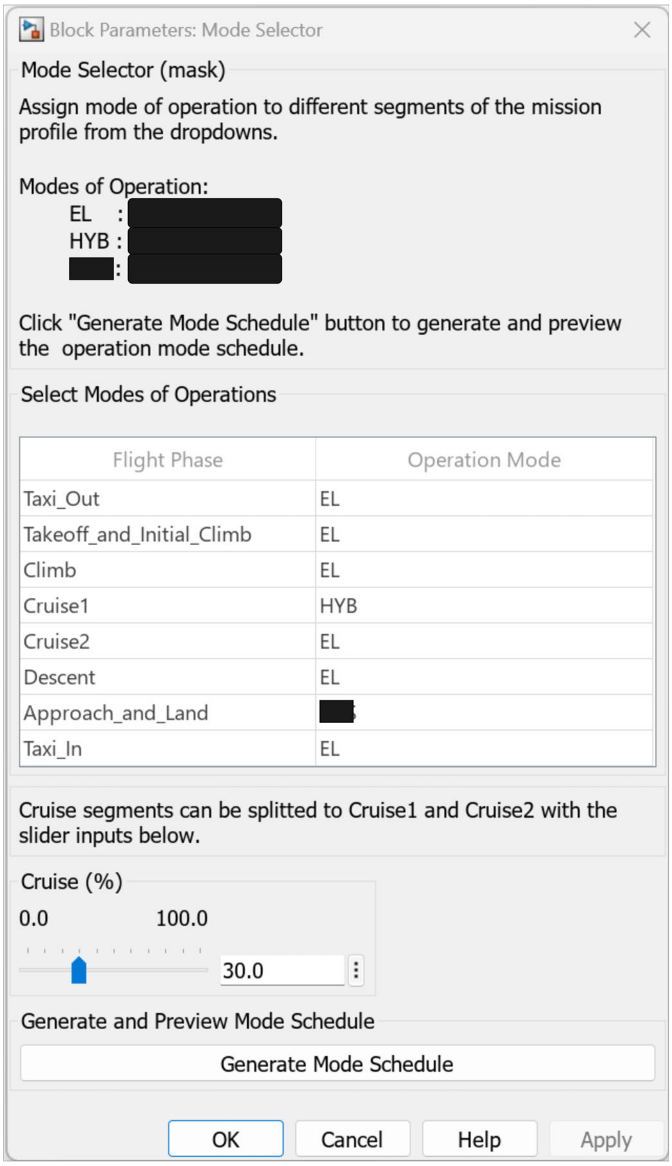


Figure 5.5: The altitude profile as the preview result of the generated mission profile.

Mode Selector:
The mask of the Mode Selector serves as the GUI, which facilitates the user to assign operation modes for different segments of the mission. The structure is automatically arranged according to the mission profile and the segments relevant for the mission profile are filtered and presented to the user. The Mode Selector mask for a reference mission without a go-around scenario is shown in Figure(5.6).



Block Parameters: Mode Selector

Mode Selector (mask)

Assign mode of operation to different segments of the mission profile from the dropdowns.

Modes of Operation:
EL : [redacted]
HYB : [redacted]
[redacted]: [redacted]

Click "Generate Mode Schedule" button to generate and preview the operation mode schedule.

Select Modes of Operations

Flight Phase	Operation Mode
Taxi_Out	EL
Takeoff_and_Initial_Climb	EL
Climb	EL
Cruise1	HYB
Cruise2	EL
Descent	EL
Approach_and_Land	[redacted]
Taxi_In	EL

Cruise segments can be splitted to Cruise1 and Cruise2 with the slider inputs below.

Cruise (%)
0.0 100.0
30.0

Generate and Preview Mode Schedule

Figure 5.6: Mode Selector mask for mission profile without go-around scenario.

The cruise segment is splitted to two segments as Cruise1 and Cruise2 for more flexible operation mode assignment. The percentage of the Cruise1 in the cruise segment can be adjusted using the slider or numeric input. The user can preview the assigned mode schedule by clicking "Generate Mode Schedule" button. The preview result corresponding to assigned modes in Figure(5.6) is shown in Figure(5.7).

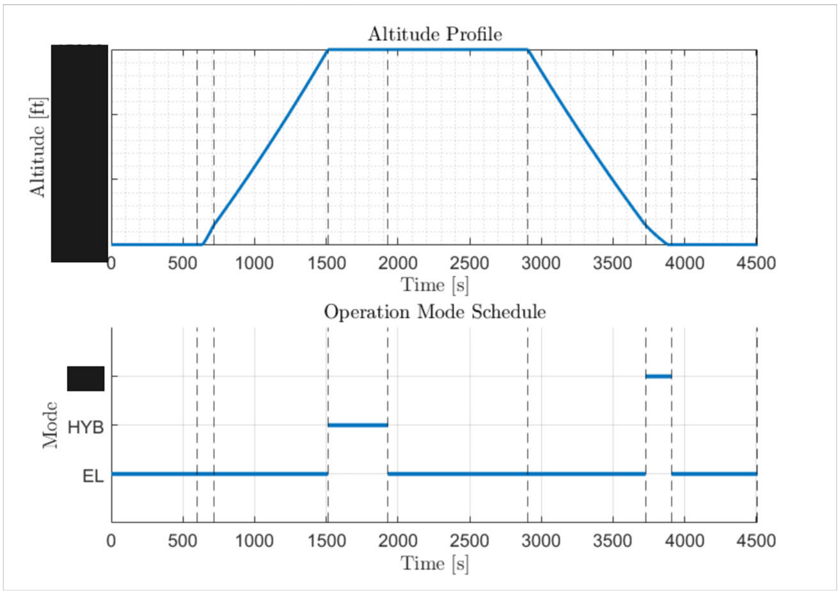
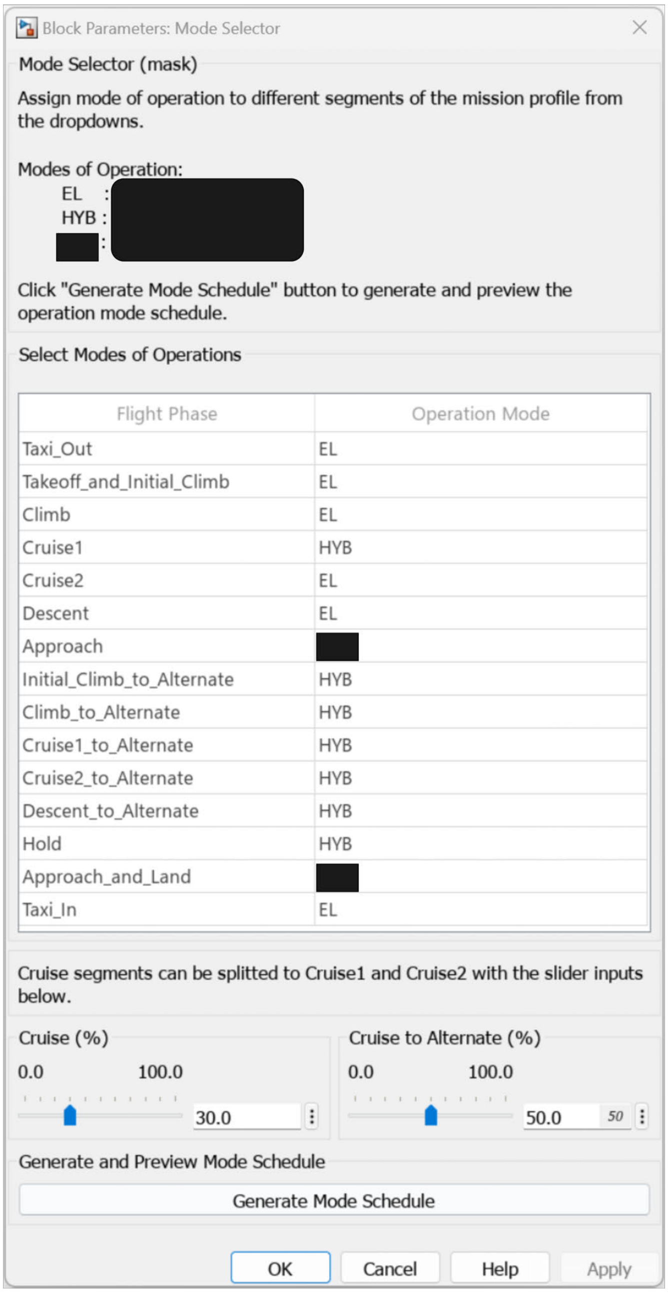


Figure 5.7: The altitude profile and mode schedule previewed by the Mode Selector mask.

Similarly, for a mission profile consists of a go-around scenario, the relevant mission segments are presented in the mask. The mask of the Mode Selector and the corresponding preview result for a mission profile with a go-around scenario is shown in Figure(5.8) and Figure(5.9) respectively.



Mode Selector (mask)

Assign mode of operation to different segments of the mission profile from the dropdowns.

Modes of Operation:
EL :
HYB :
[redacted] :
[redacted]

Click "Generate Mode Schedule" button to generate and preview the operation mode schedule.

Select Modes of Operations

Flight Phase	Operation Mode
Taxi_Out	EL
Takeoff_and_Initial_Climb	EL
Climb	EL
Cruise1	HYB
Cruise2	EL
Descent	EL
Approach	[redacted]
Initial_Climb_to_Alternate	HYB
Climb_to_Alternate	HYB
Cruise1_to_Alternate	HYB
Cruise2_to_Alternate	HYB
Descent_to_Alternate	HYB
Hold	HYB
Approach_and_Land	[redacted]
Taxi_In	EL

Cruise segments can be splitted to Cruise1 and Cruise2 with the slider inputs below.

Cruise (%)
0.0 100.0
30.0

Cruise to Alternate (%)
0.0 100.0
50.0 50

Generate and Preview Mode Schedule

Figure 5.8: Mode Selector mask for mission profile with a Go-Around scenario.

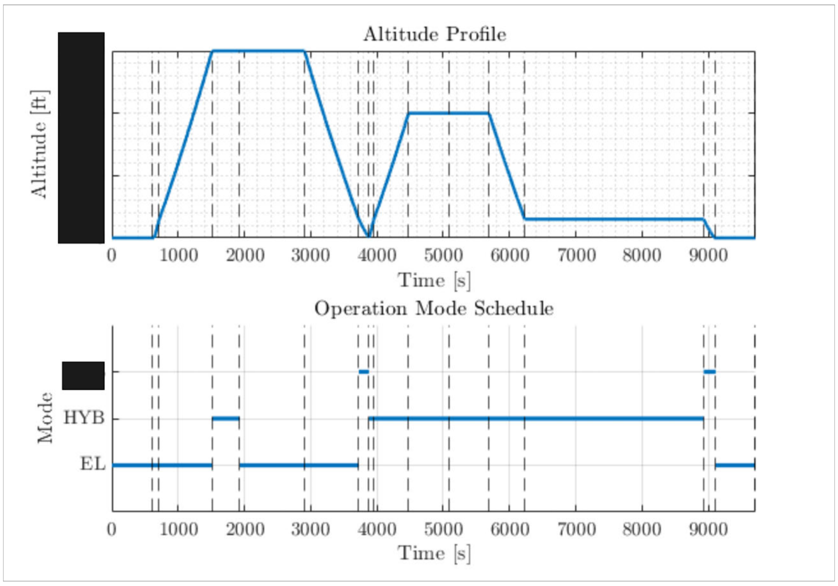
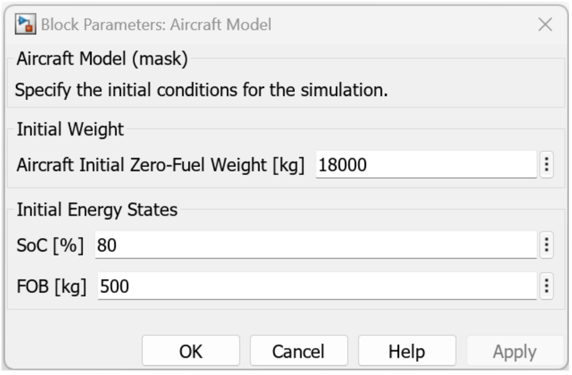


Figure 5.9: The altitude profile and mode schedule previewed by Mode Selector mask for a mission profile with a go-around scenario.

Aircraft Model:
The mask of the Aircraft Model facilitates the user to specify initial parameters related to aircraft weight and the on-board energy levels as shown in the Figure(5.10).



Aircraft Model (mask)

Specify the initial conditions for the simulation.

Initial Weight

Aircraft Initial Zero-Fuel Weight [kg] 18000

Initial Energy States

SoC [%] 80

FOB [kg] 500

Figure 5.10: The mask of the Aircraft Model.

Visualizations:
The simulation model facilitates the user to generate number of plots directly from the simulation model. Additionally, extensive number of simulation signals and parameters including the internal signals of different sub-models are logged facilitating in-depth analysis. Analysing the simulation results across multiple simulation runs by varying the parameters of interest can be performed using inbuilt tools such as Simulink Data Inspector. The plots which can be directly generated by the simulation models are summarized in Table(5.1).

Table 5.1: Buttons for generating simulation results in the simulation model and the corresponding plots.

Button	**Plots**
Plot Mission-Profile	- Altitude Profile - Speed Profile - Flightpath Angle Profile - Mission Segments
Plot Mode Schedule	- Mission Segments - Operation Mode Profile
Plot Ambient Parameters	- Pressure Altitude - Outside Air Temperature - Static Pressure - Air Density - Speed of Sound
Plot Thrust	- Aircraft-level Required Thrust - Required Thrust EPS - Required Thrust TES
Plot Power	- Aircraft-level Propulsive Power Required - Required Power EPS - Required Power TES
Plot Aircraft Weight	- Aircraft Weight
Plot Fuel-on-Board	- Fuel Consumption - Fuel Remaining
Plot Battery SoC	- Power Demand - Remaining Energy [kWh] - SoC [Ah] - SoC [%]
Plot Battery Param	- Power Demand - OCV - Current
Plot Operating Points	- Motor Operating Points - Engine Operating Points
Plot Propeller Efficiencies	- EPS Propeller Efficiency - TES Propeller Efficiency

5.2 Simulation Results

5.2.1 Hybrid Aircraft Typical Mission Simulation

A flight mission was simulated in the simulation model under ISA conditions without any headwind. The initial parameters listed in the Table(5.2) were used for the simulation. The main simulation results are presented and discussed in separate sections below.

Table 5.2: Parameter values used for the hybrid aircraft typical mission simulation.

Parameter	Value
Aircraft ZFW	x kg
Initial FoB	x kg
Initial PBU SoC	x%
PBU configuration	xpxs
PBU cell capacity	x Ah

5.2.1.1 Mission Profile

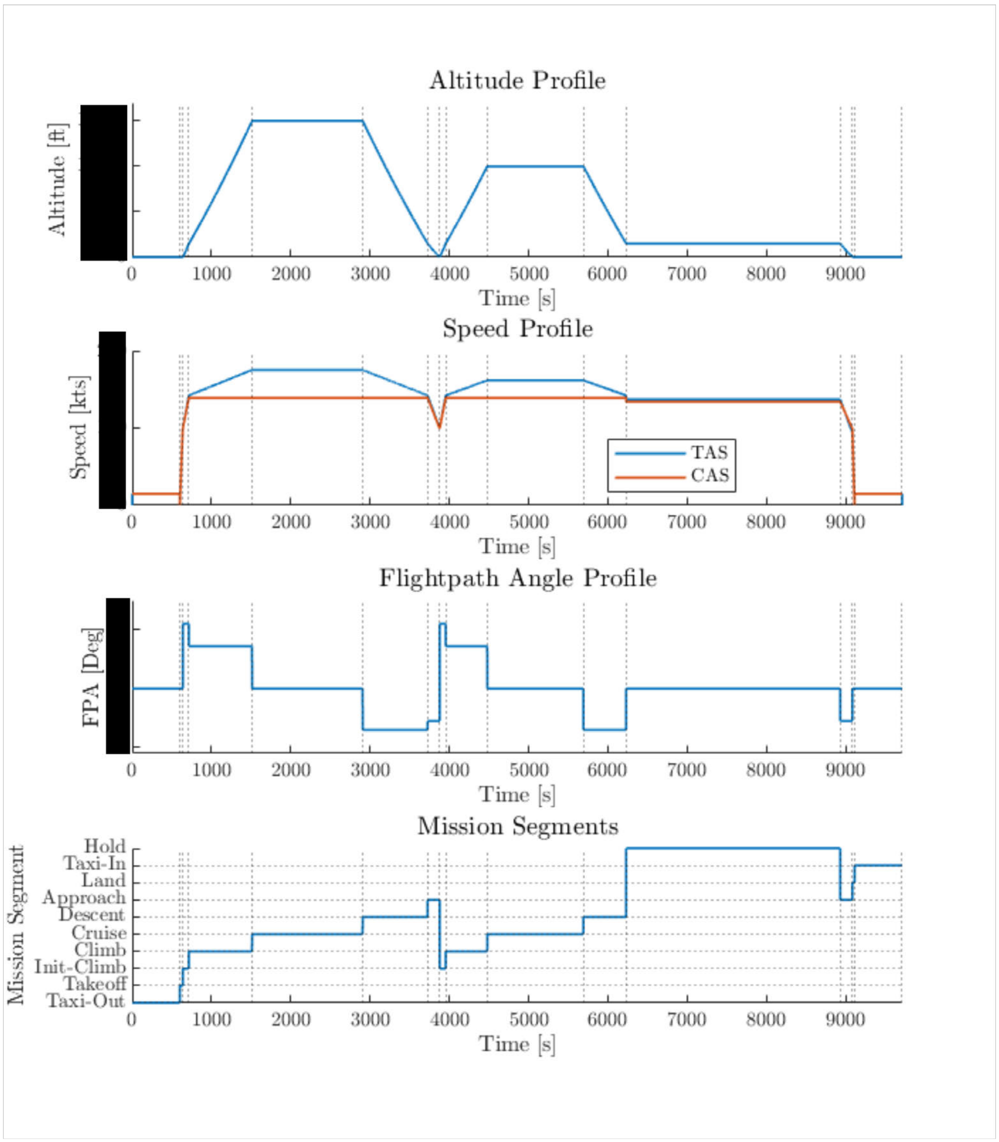


Figure 5.11: Typical mission simulation results: Mission Profile

The Mission Profile plot consists of the altitude, speed, FPA and mission segment profiles against flight time as shown in Figure(5.11). The first segment of the mission taxi-out is at x KTAS for 10 minutes. In the takeoff, the aircraft accelerates from 0 to x KCAS which is the takeoff speed. In the initial climb phase the aircraft climbs to x ft AGL while accelerating from x KCAS to x KCAS climb speed. As we can observe from the FPA plot, the initial climb is steep at an average FPA of x^o. In the climb segment, the aircraft climbs from x ft AGL to the cruise altitude of x ft AMSL at fixed x KCAS and relatively shallower FPA of x^o. As it can be observed from the speed profile plot, although the CAS is fixed throughout the climb segment, the change of ambient air density in climb results a varying TAS. This change of TAS is regarded as a constant acceleration throughout the climb segment.

The aircraft cruises at x ft AMSL at x KCAS where it is equivalent to approximately x KTAS and the simulation model is able to capture this accurately. The next segments of the mission are descent and approach, where the aircraft descends to the destination airport from x ft to x ft at x KCAS and x ft to 0 ft decelerating from x KCAS to x KCAS touchdown speed respectively.

The rest of the segments represent a missed-approach and diversion scenario where the aircraft is flying to an alternate airport at cruise altitude of x ft AMSL. Further, before approaching and landing to the alternate airport, the aircraft perform a holding at x AGL for x minutes at x KCAS.

5.2.1.2 Mode Schedule

The Mode Schedule plot consists of subplots for mission segments and the operation mode profile. The assigned operation modes for the simulation, as selected with the Mode Selector prior to the simulation, have been accurately captured and represented in Figure (5.12), which depicts the mode schedule of the simulation model. Notably, during the aircraft mission, the EL mode is active until reaching the top of climb (TOC). The initial 40% of the cruise phase (Cruise1) is in HYB mode and remaining cruise segment (Cruise2) and the descent segments are in EL mode. The approach segment operates in x mode, and all mission segments after a missed approach remain in HYB mode, except for the approach to the alternate airport, which switches to x mode. Upon landing at the alternate airport, the taxi-in mode is EL.

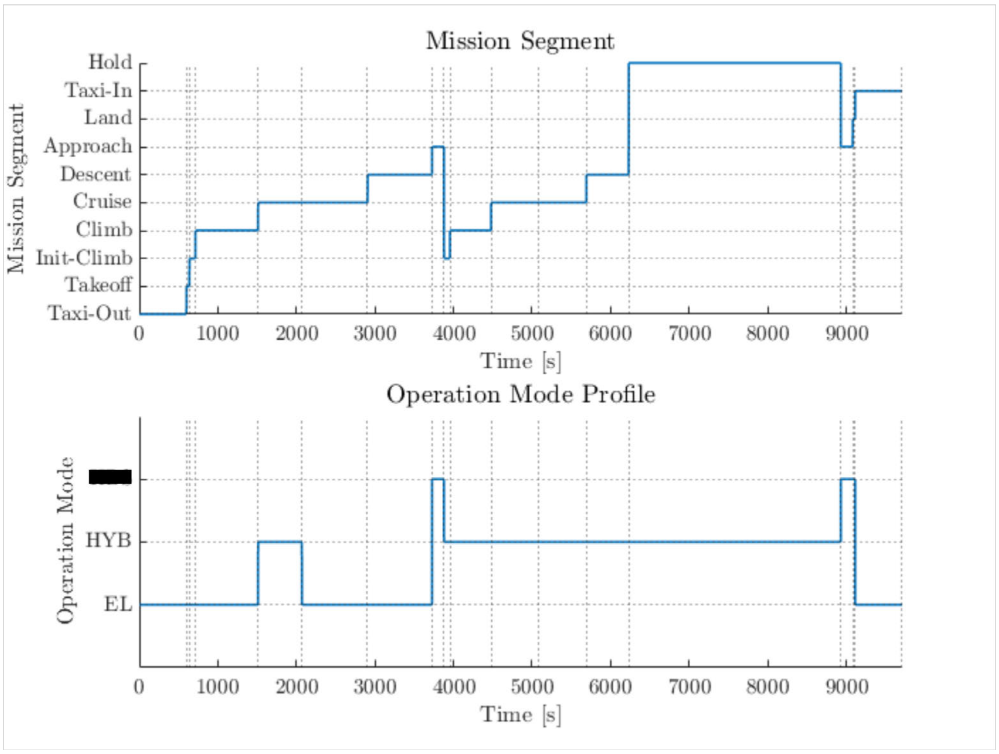


Figure 5.12: Hybrid aircraft typical mission simulation results: Mode Schedule

5.2.1.3 Ambient Parameters

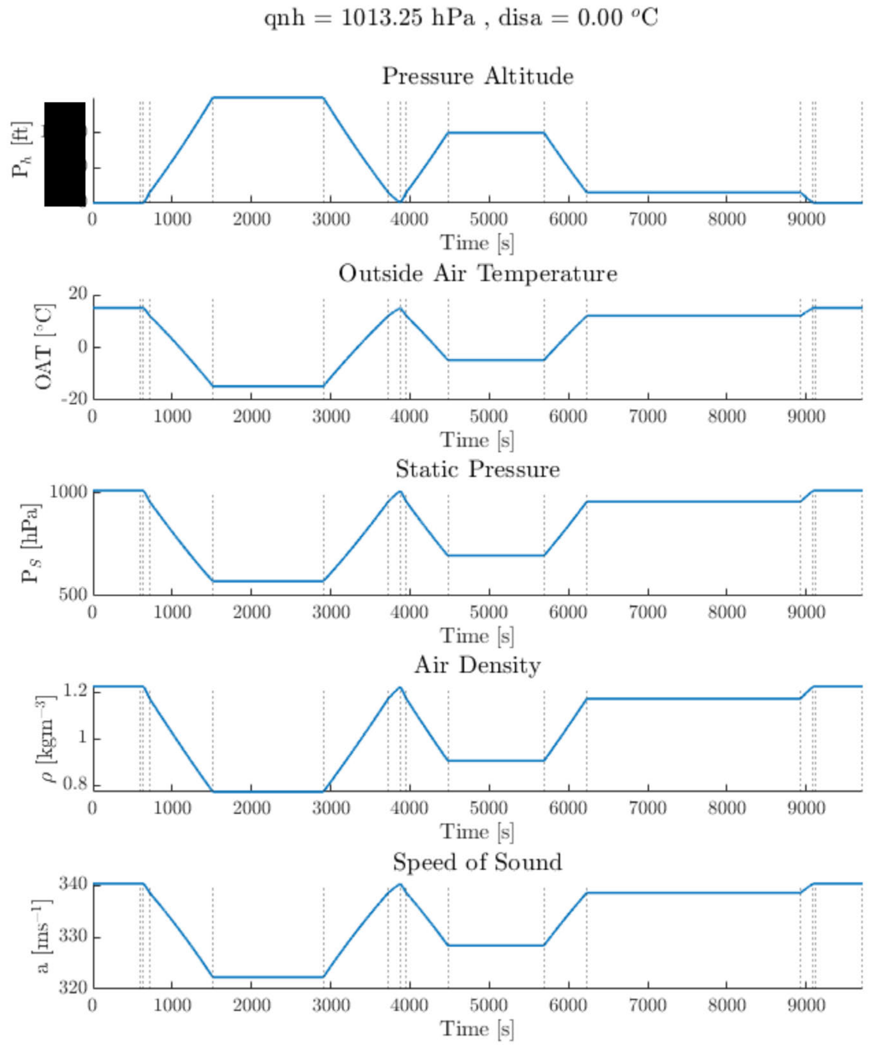


Figure 5.13: Hybrid aircraft typical mission simulation results: Ambient Parameters

As depicted in Figure (5.13), the Ambient Parameters plots illustrate the variations of the ambient parameters to which the aircraft was exposed over the course of the flight. As the considered atmospheric conditions is ISA, the pressure altitude is same as the altitude profile. The OAT, static pressure, air density and speed of sound values are reduced when the aircraft gains altitudes. Therefore, the ambient parameter plots show the expected behaviour.

5.2.1.4 Thrust

The Thrust plot includes subplots for aircraft-level thrust required, EPS thrust, and TES thrust against flight time, as depicted in Figure (5.14). Additionally, the EPS thrust plot corresponds to a single EPS unit, while the TES thrust values correspond to a single TES unit.

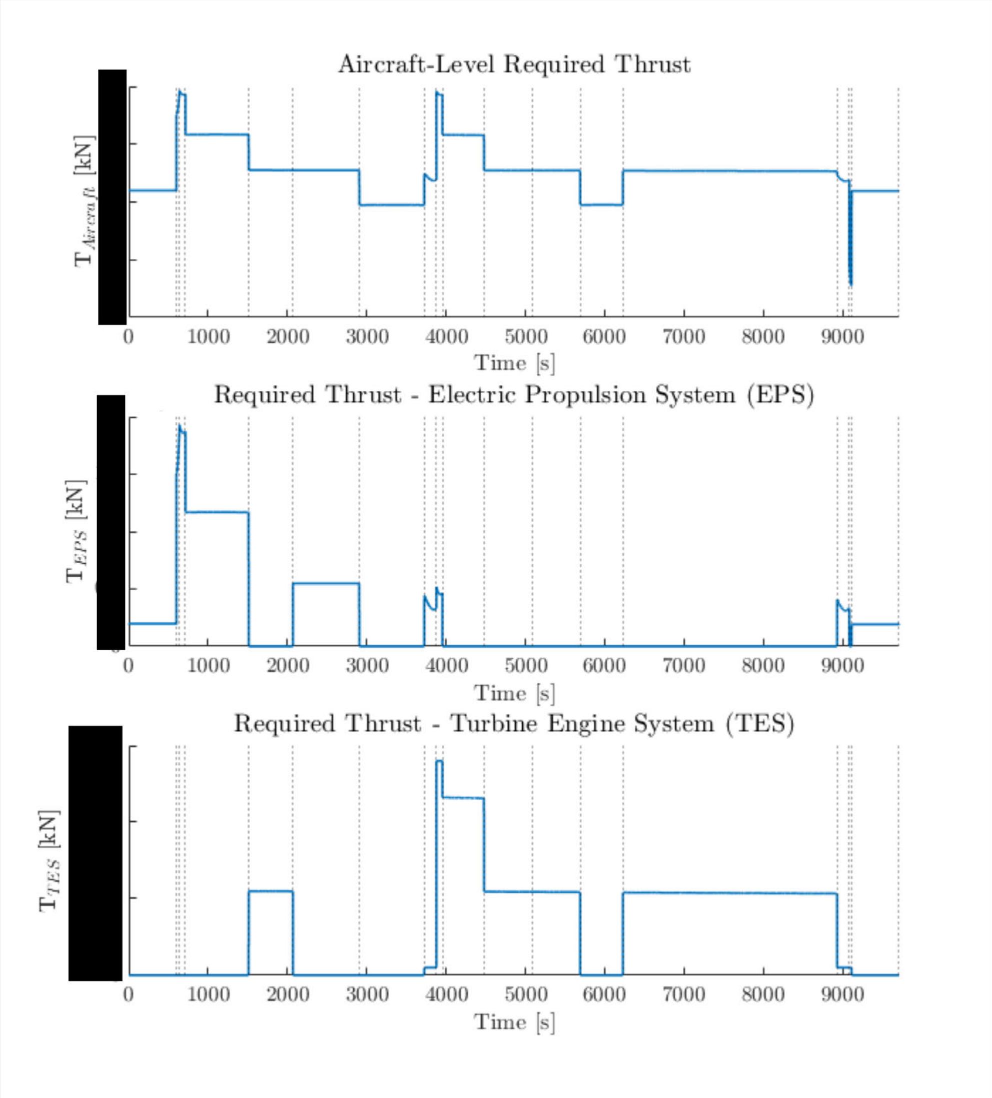


Figure 5.14: Hybrid aircraft typical mission simulation results: Thrust

It can be observed that in regions with EL mode, only EPS thrust is present, while in regions with HYB mode, only TES thrust is present. In the regions with HBS, both TES and EPS thrust are present, in agreement with the Master Controller logic used for simulating the split of aircraft-level thrust into EPS and TES components. Another noteworthy observation is that the negative thrust in the aircraft-level thrust required plot is not reflected in the TES or EPS thrust plots. This discrepancy arises from the fact that the negative thrust was neglected during the modeling of the propeller models as a simplification step.

The highest required thrust occurs during take-off, followed by the initial climb phases. Additionally, the climb phase demands relatively high thrust compared to the cruise phase. In contrast, the required thrust during the descent phase is in the negative region, while the thrust needed during the approach phase is considerably higher. This increase in thrust during approach is due to the extra drag introduced by the deployment of flaps and landing gear.

5.2.1.5 Power

The Power plot includes subplots for aircraft-level propulsive power, EPS power, and TES power where EPS power and TES power correspond to single unit of EPS and TES respectively. These power plots depicted in Figure(5.15) exhibit behavior similar to the corresponding thrust plots, with the highest power occurring during take-off and initial climb segments. During climb and cruise, power decreases. In the approach segment, power increases to compensate for the additional thrust required due to increased drag resulting from the deployment of flaps and landing gear. The plots for TES power and EPS power agree with the corresponding thrust plots as well as the operation mode schedule.

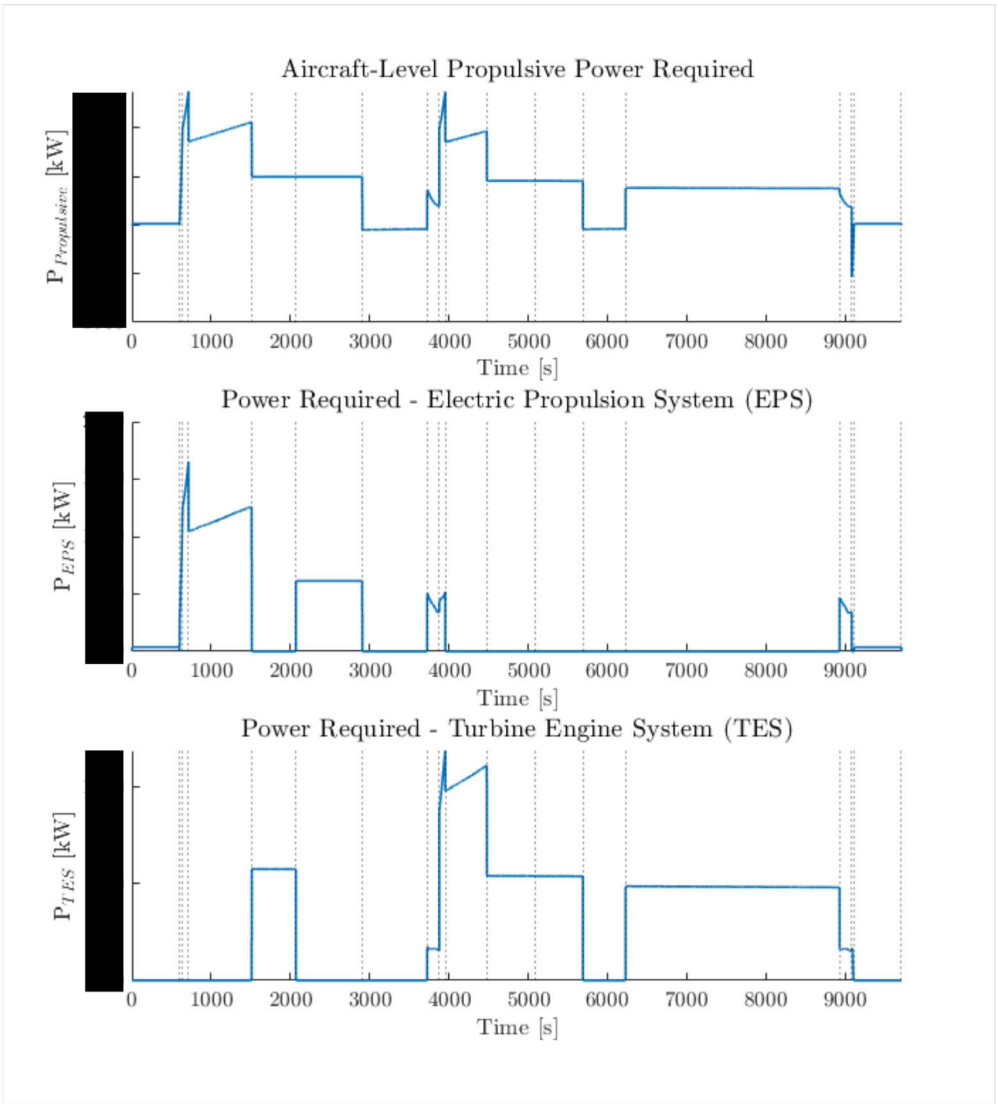


Figure 5.15: Hybrid aircraft typical mission simulation results: Power

5.2.1.6 Fuel-on-Board

The Fuel-on-Board plot comprises subplots for fuel consumption and fuel remaining, as depicted in Figure (5.16). The fuel consumption plot illustrates the fuel usage of a single unit of TES, while the fuel remaining plot provides information on the total remaining fuel onboard the aircraft. Notably, fuel consumption is zero during segments operated in EL mode. Conversely, the highest fuel consumption occurs during the missed-approach and initial climb, where power demand is at its peak, and these segments are operated in HYB mode. The drop in fuel remaining values corresponds to the fuel consumption values, with segments exhibiting high fuel consumption resulting in steep drops.

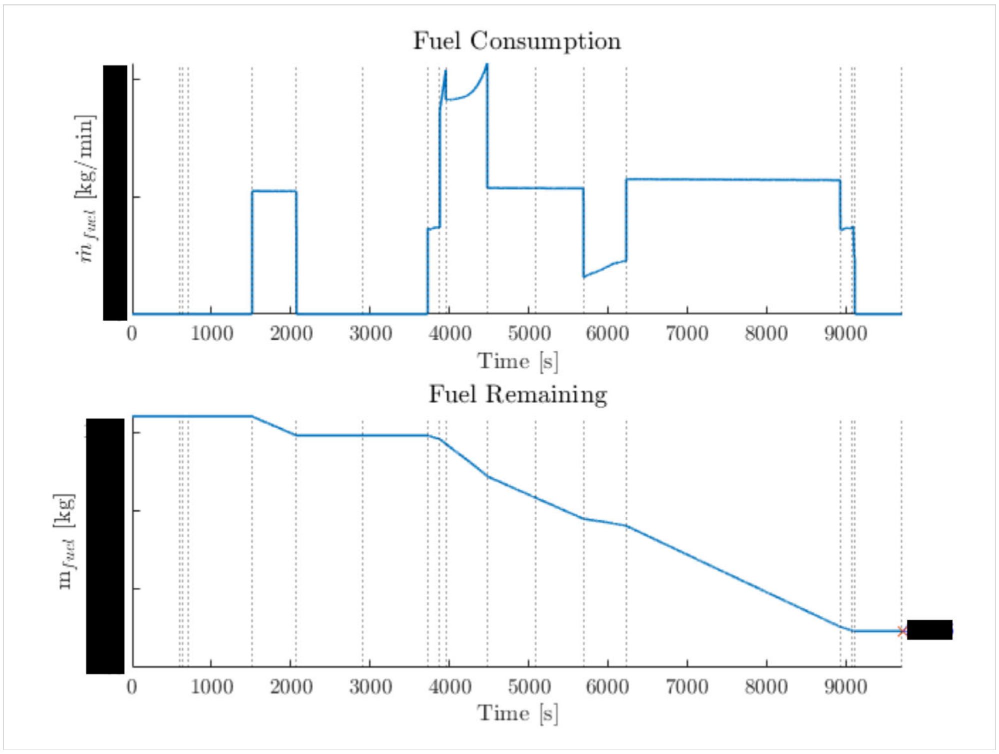


Figure 5.16: Hybrid aircraft typical mission simulation results: Fuel-on-Board

5.2.1.7 Aircraft Weight

The Aircraft Model in the simulation accounts for the weight change of the aircraft, which occurs due to fuel burn. The Aircraft Weight plot, depicted in Figure (5.17), provides information about the weight variation of the aircraft throughout the flight mission. This plot corresponds to the fuel remaining plot, as the aircraft weight is the sum of the fuel weight and the zero fuel weight of the aircraft.

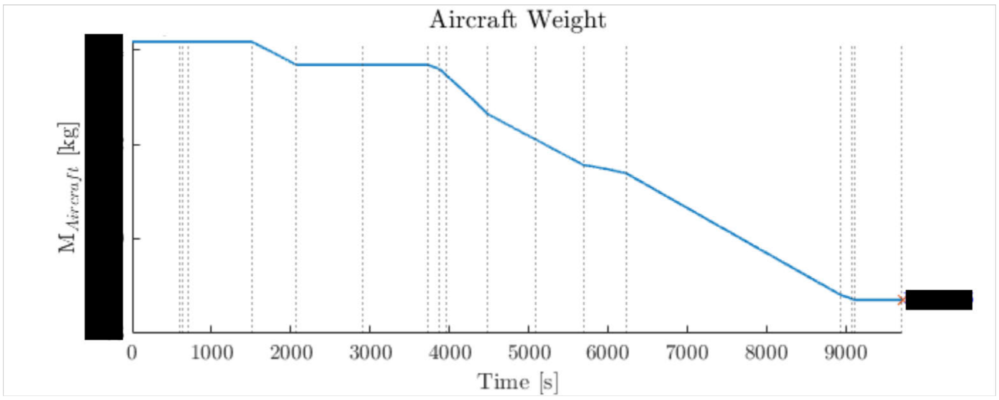


Figure 5.17: Hybrid aircraft typical mission simulation results: Aircraft Weight

5.2.1.8 Battery State-of-Charge

The Battery State-of-Charge (SoC) plot, depicted in Figure (5.18), comprises subplots for power demand, remaining energy in kWh, SoC in Ah, and SoC as a percentage value and all these plots corresponds to a single unit of battery pack in the aircraft. Notably, the highest battery power demand occurs during takeoff and initial climb segments, resulting in the greatest consumption of PBU energy in these phases.

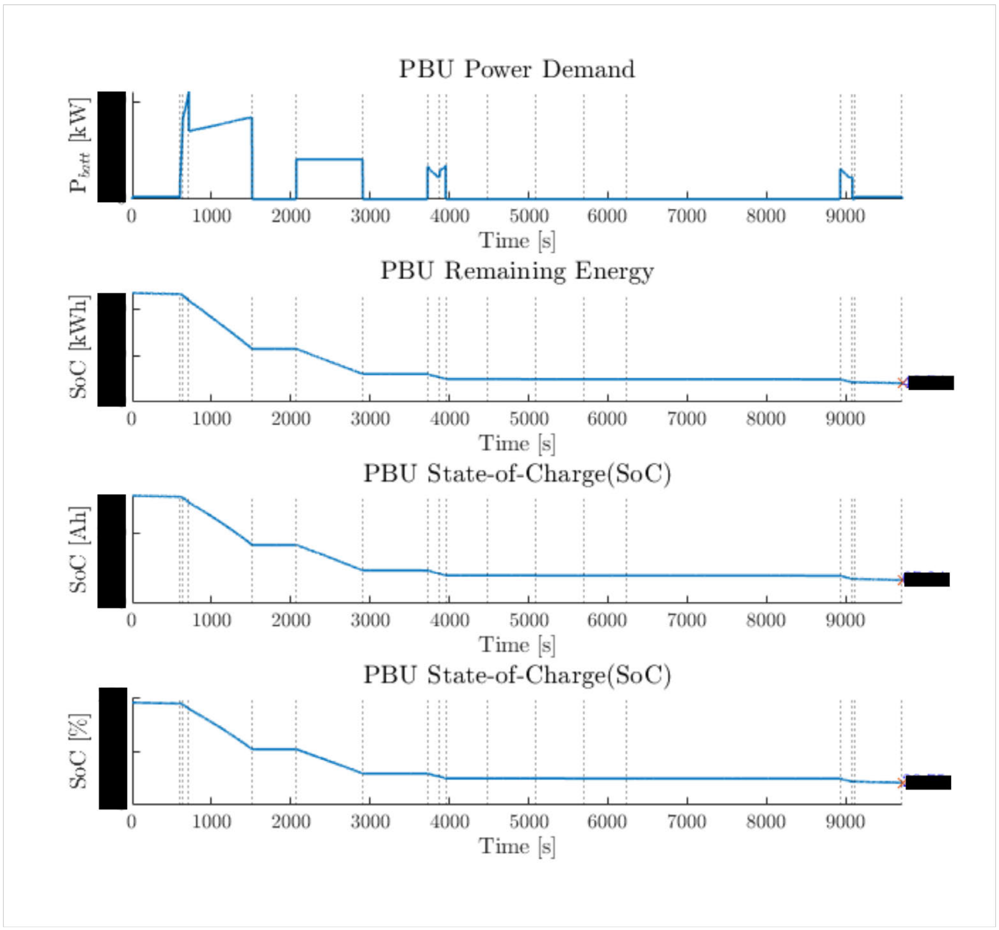


Figure 5.18: Hybrid aircraft mission simulation results: Battery SoC

5.2.1.9 Battery Parameters

The Battery Parameters plot comprises subplots for power demand, open circuit voltage (OCV), and current values. Notably, the highest current aligns with the peak power demand.

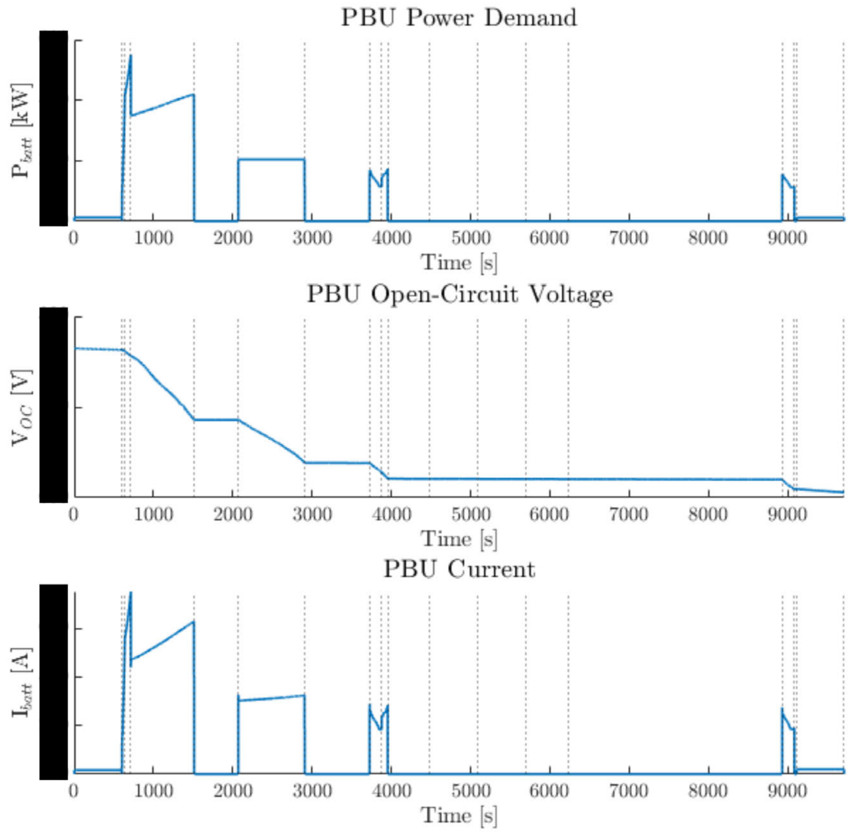


Figure 5.19: Hybrid aircraft typical mission simulation results: PBU Parameters

5.2.1.10 Operating Points

The Operating Points plots consists of operating points of the turbine engine and the PMSM as shown in Figure(5.20) and Figure(5.21) respectively.

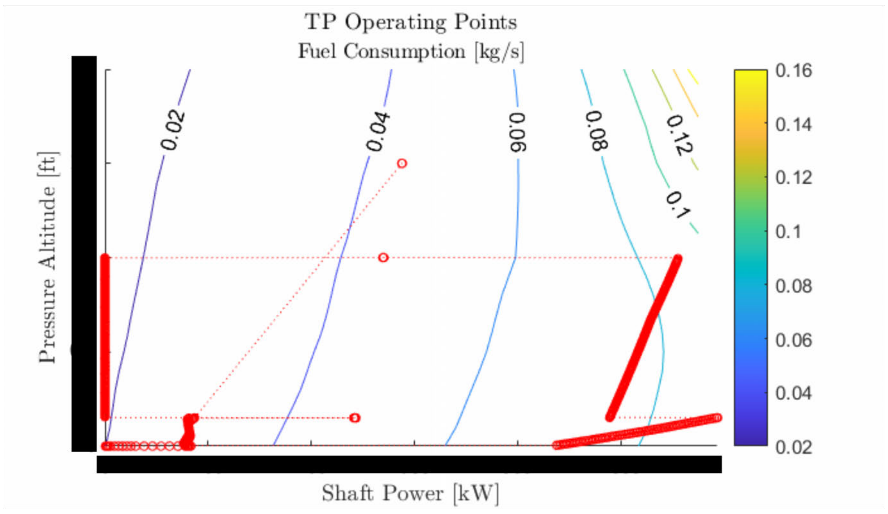


Figure 5.20: Hybrid aircraft typical mission simulation results: Turbine Engine Operating Points

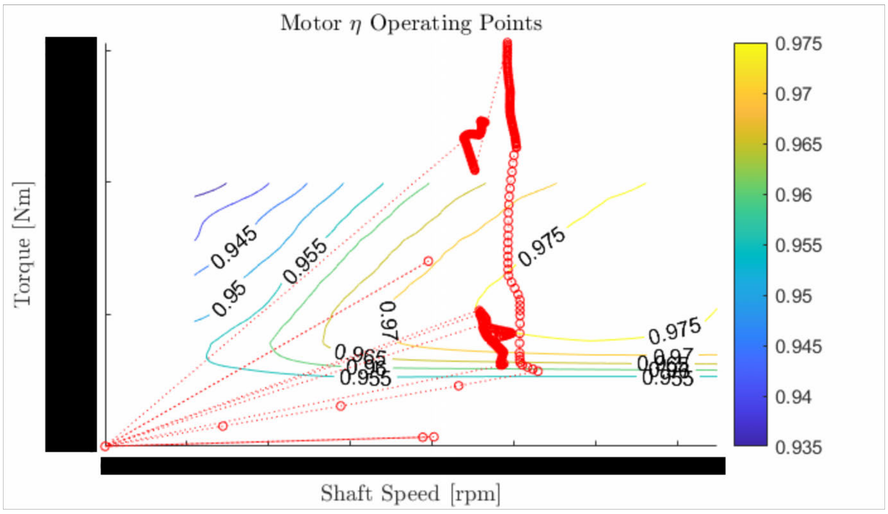


Figure 5.21: Hybrid aircraft typical mission simulation results: PMSM Operating Points

5.2.1.11 Propeller Operating Point Efficiency

The Propeller efficiency plot comprises subplots for TES and EPS propeller efficiencies against simulation time as shown in Figure(5.22).

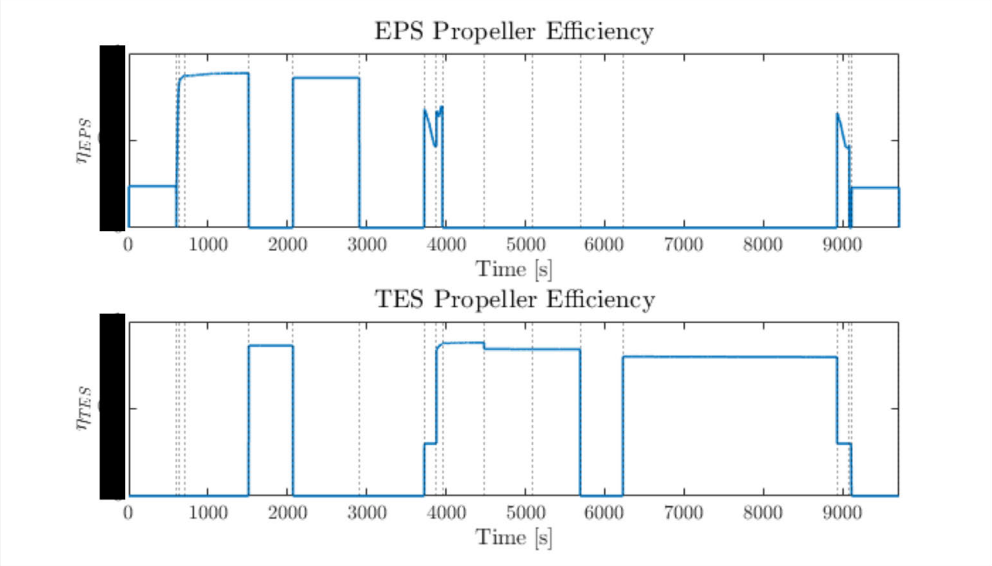


Figure 5.22: Hybrid aircraft typical mission simulation results: Propeller Operating Point Efficiencies.

5.2.2 Comparison Against Baseline Mission Data

To assess the performance of the simulation model, a comparative analysis was conducted using baseline mission data. Specifically, a baseline mission was simulated in the simulation model configuring initial conditions and system parameters to match those for the baseline data.

ISA atmospheric conditions with no headwind was considered for the weather model. MTOW of 19418 kg, initial FoB of 1040 kg and 90% of initial PBU SoC were used as the initial parameters for the simulation to match the corresponding values to the baseline. Furthermore, during the cruise segment, the first 40% was in HYB mode, while the remaining cruise segment was in EL mode and the approach and landing were in x mode.

The simulation results for power and energy consumption of the EPS and TES systems are plotted alongside the baseline mission data in Figure (5.23). The PBU Power plot reflects the total power demand from both EPS units, while the TES Power plot represents the combined values from both TES units.

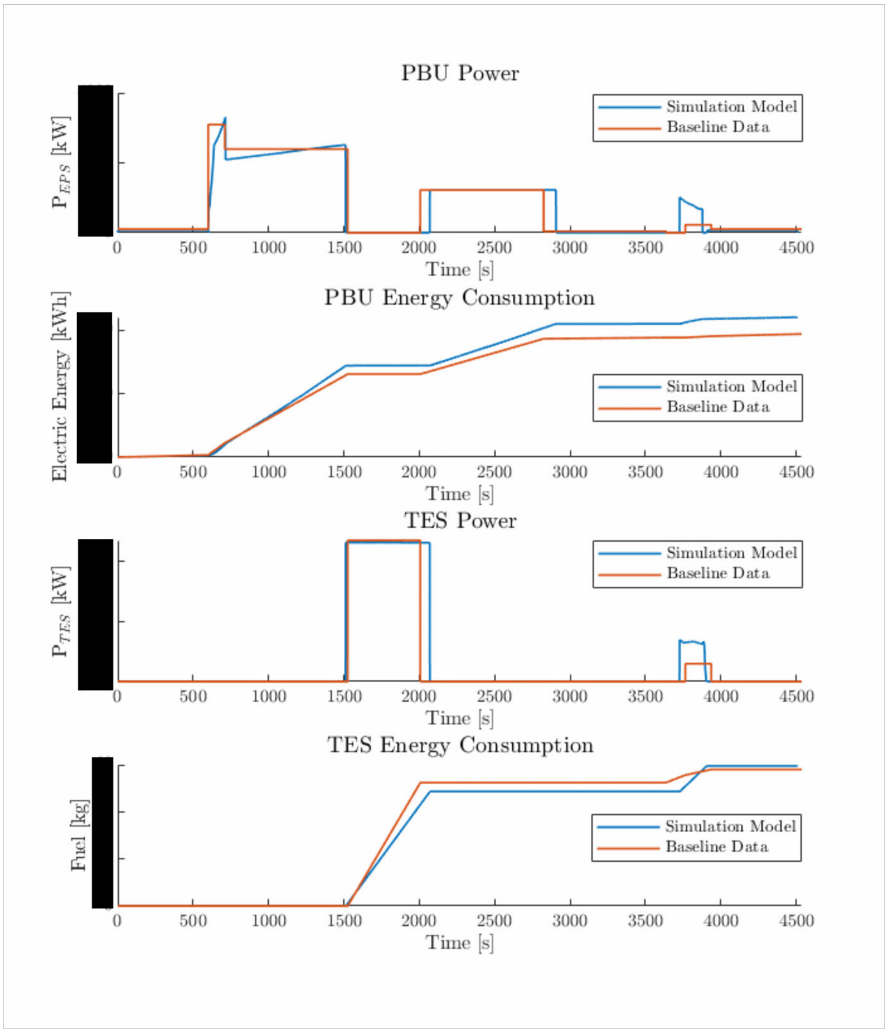


Figure 5.23: Comparison typical mission simulation results against baseline mission data.

Based on the results, the power drawn from the PBU in both the baseline and simulation models peaks during takeoff and initial climb. In the baseline data, the power requirement during takeoff and initial climb remains constant. However, the simulation model accounts for varying TAS, resulting in varying power requirements. Despite this difference, the values closely match. Additionally, the average power requirement during the climb phase aligns well between the simulation model and baseline mission data.

During the cruise phase, both the TES power and PBU power readings in the simulation model closely align with the values from the baseline mission data. Although there is a slight discrepancy in PBU energy consumption between the simulation model and baseline data, the overall results remain close. Furthermore, the fuel consumption plots for both the simulation model and the baseline provide similar results.

5.2.3 Cruise Altitude Survey

The altitude survey function in the FMS performs optimization of the vertical flight profile to achieve a certain goal such as minimizing time en-route, minimizing fuel consumption etc. Further, energy states and flight time estimates for various altitude levels are provided, allowing pilots to select the most efficient cruise altitude. The objective of this simulation analysis is to evaluate energy consumption and flight time estimates for different cruise altitude levels, similar to the FMS altitude survey function for hybrid aircraft.

The simulation was conducted using the initial parameters and configuration as specified in Table(5.2). For the mission profile, the hybrid aircraft typical mission, covering only from the departure airport to the destination airport was considered for the simulation. The simulation followed a mode schedule where the first x% of cruise in HYB mode while the remainder of the cruise in EL mode, x mode for the approach and landing. No headwind under ISA atmospheric condition was considered for the simulation.

In real-world flight operations, several factors influence the choice of cruise altitude. Typically the cruise altitude is selected to ensure that the cruise phase constitute more than x% of the entire flight. The cruise altitude also depends on the direction of flight: an Instrument Flying Rules(IFR) aircraft flying a magnetic track of 0°-179° should cruise at odd thousands altitudes (i.e. 13000 ft, 15000 ft, 17000 ft etc.), while IFR aircraft flying a magnetic track of 180°-359° should cruise at even-thousands altitudes (i.e. 12000 ft, 14000 ft, 16000 ft etc.) [25]. For this simulation, different cruise altitudes ranging from x ft to x ft in increments of 1000 ft were considered.

The simulation result for the mission profiles with varying cruise altitudes are depicted in Figure(5.24).

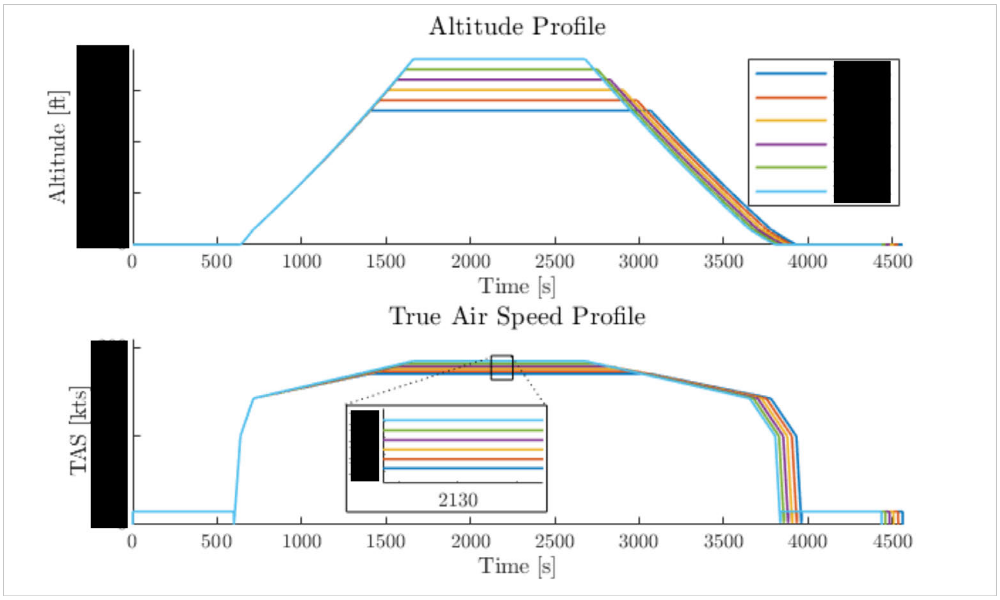


Figure 5.24: Cruise altitude survey: Mission profile.

As observed in the mission profile, flying the same trip at different cruise altitudes have resulted varying flight times. The longest flight time is recorded when cruising at x ft, while the shortest flight time occurs when cruising at x ft. This can be explained with reference to the TAS plot. When cruising at x ft, the cruising TAS reaches the peak value of x kts, whereas at x ft, it is only x kts. This TAS difference impacts the overall flight time to complete the mission. The variation in cruising TAS can be explained by examining the simulation result for ambient parameters as depicted in Figure(5.25).

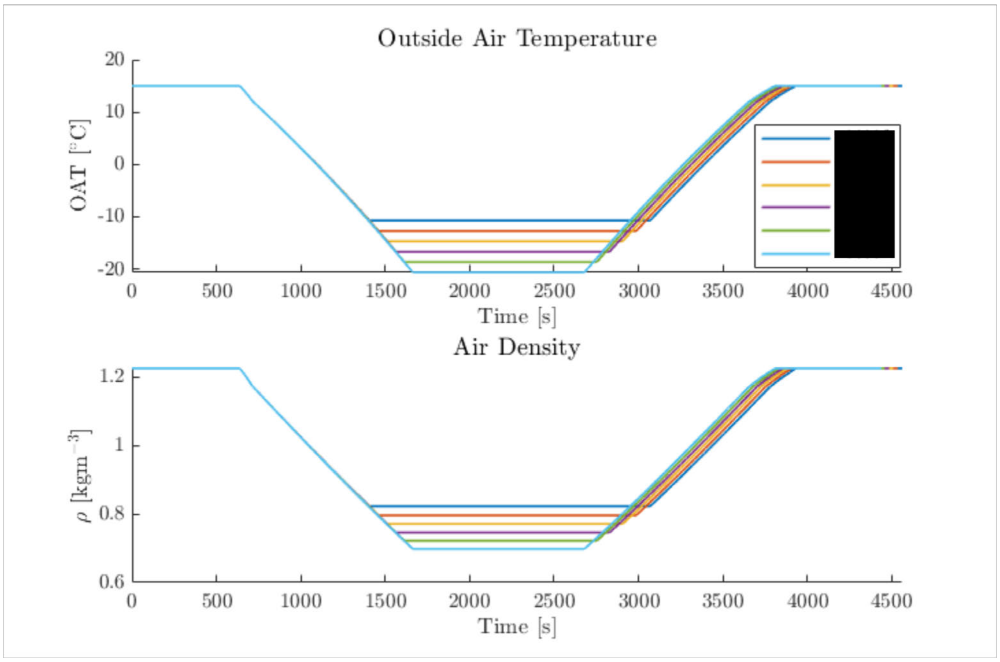


Figure 5.25: Cruise altitude survey: OAT and air density ambient parameters.

Referring to Figure(5.25), it can be observed that the OAT decreases when the altitude is increased. Consequently, the aircraft experiences the lowest OAT, cruising at x ft compared to the other considered altitudes. The air density is also decreased with altitude and the air density also takes the lowest value at x ft in this simulation. Since the aircraft maintains the same CAS for cruise across the different scenarios in the simulation, the TAS increases with decreasing air density. As a result, the TAS of the aircraft is highest when cruising at x ft, aligning with the finding from Figure(5.24).

Figure(5.26) depicts the simulation results illustrating the aircraft-level required thrust and propulsion power for various cruise altitude scenarios. While the aircraft-level thrust in cruise phase remains nearly the same across different altitudes, a closer examination reveals a slight increase in thrust demand when cruising at higher altitudes.

The mathematical relationship for the required thrust during cruise phase can be derived using Equation(3.18) and Equation(3.17) as follows:

$$T_{\text{required}} = D \tag{5.1}$$

$$T_{\text{required}} = \frac{1}{2} C_D \rho v^2 S_{ref} \tag{5.2}$$

Based on the simulation results for ambient parameters and the mission profile, it is observed that air density decreases at higher cruise altitudes, while the TAS of the aircraft increases. The impact of reduced air density (ρ) on the required thrust is linear (order of 1), whereas the effect of TAS (v) is quadratic (order of 2) according to the Equation(5.2). Consequently, the simulation shows a slight increase in required thrust at higher altitudes compared to the other altitudes. This effect becomes more pronounced in the propulsive power plot, as propulsive power is calculated by multiplying the required thrust by TAS.

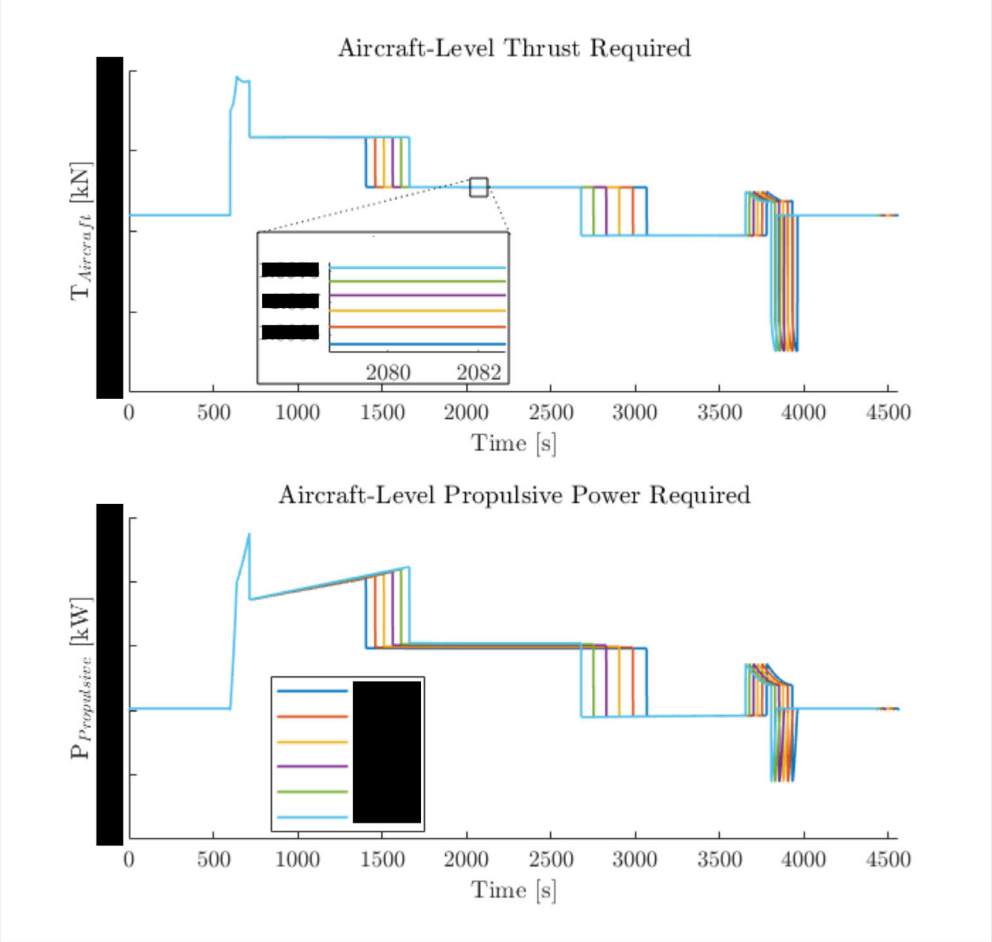


Figure 5.26: Cruise altitude survey: Aircraft-level required thrust and propulsive power.

The simulation results for the SoC and FoB for the cruise altitude survey is shown in Figure(5.27).

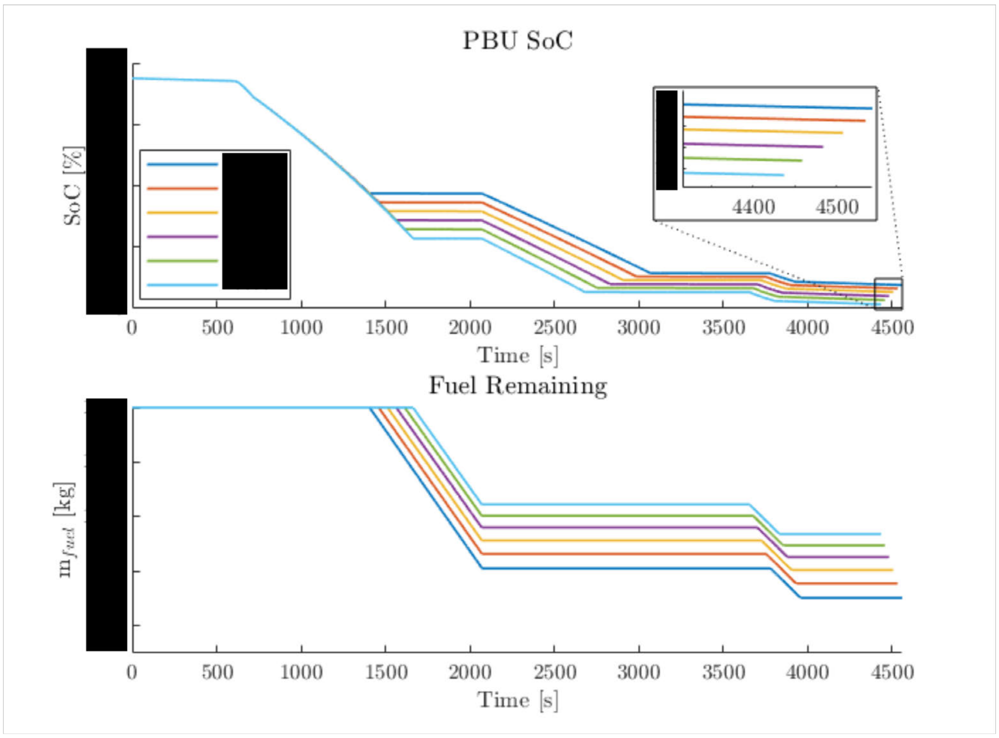


Figure 5.27: Cruise altitude survey: PBU SoC and FoB.

According to Figure(5.27), the end of flight SoC takes the maximum value of x% for the scenario cruising at x ft. Conversely, the minimum SoC value of x% occurs at the end of the flight when cruising at x ft. However, cruising at x ft results the minimum FoB level of x kg at the end of the flight, while the highest remaining FoB of x kg at the end of the flight is recorded for the scenario cruising at x ft.

When cruising at x ft, the aircraft spends an extended duration in climb relative to the other scenarios, and the climb is performed in EL mode. Consequently, the PBU energy is consumed during this climb, resulting in a lower PBU SoC at the end of the flight. Additionally, the increased TAS leads to a shorter time duration for the cruise phase, resulting in lower fuel consumption. As a result, the maximum remaining FoB at the end of the flight is observed when cruising at x ft. Another reason that contributes for lower fuel consumption in this scenario is that the fuel consumption of the turbine engine is low at higher altitudes as evident in Figure(5.28)

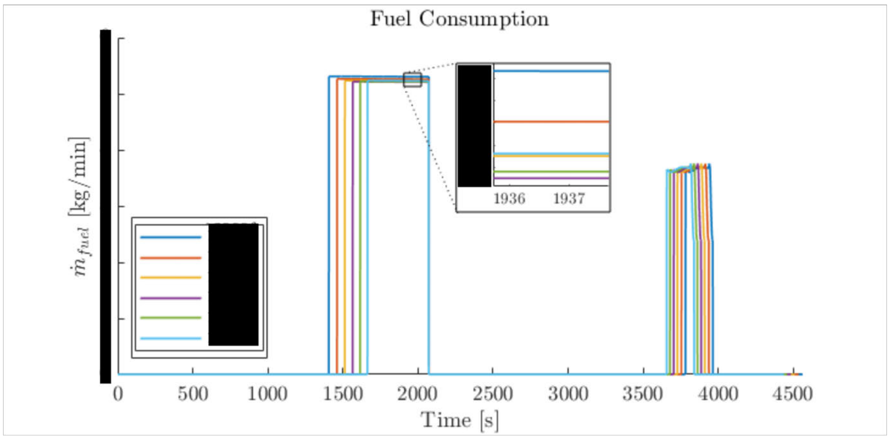


Figure 5.28: Cruise altitude survey: Turbine engine fuel consumption.

Conversely, the climb segment is shorter for the scenario cruising at x ft, while the cruise segment is extended. Therefore the PBU energy consumption is lower because the aircraft spends less time in climb in EL mode. However, the extended cruise time has an impact on the SoC as the final x% of cruise segment is operated in EL mode for this simulation and this can be observed in the SoC trajectories. Although the SoC trajectories start far apart on the y-axis at the beginning of the cruise, the trajectories of SoC converge toward the end of the flight. Moreover, the extended cruise time results more fuel consumption resulting lowest remaining FoB at the end of the flight when flying at x ft.

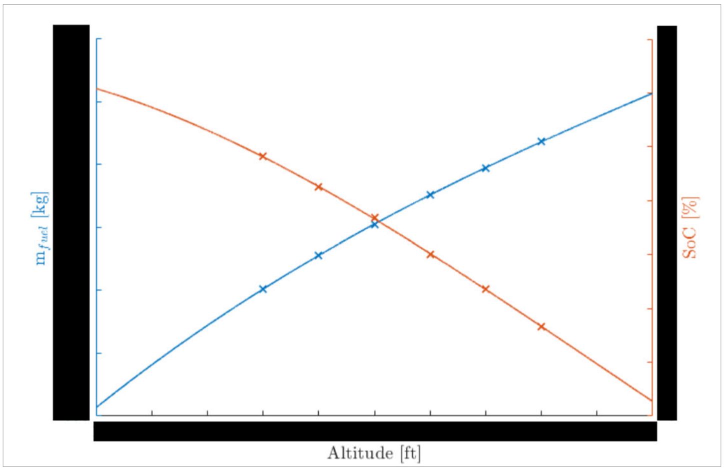


Figure 5.29: Cruise altitude survey: PBU SoC and FoB at the end of flight for different cruise altitudes.

The SoC and FoB values at the end of flight from the simulation results were plotted against altitude values with curve fitting using polynomials of order 3, and depicted in Figure(5.29). This result reveals an interesting trade-off. While gaining altitude can save fuel, it adversely impacts the PBU SoC if the climb is in EL mode. Conversely, cruising at lower altitude conserves PBU SoC but increases fuel consumption. However, this result is highly dependent of the mode schedule for the flight. Nevertheless, for the given mode schedule, the optimal cruise altitude is around x ft.

5.2.4 Cruise Speed Survey

The performance function of a Flight Management System (FMS) includes optimizing cruise speed to achieve specific goals, such as minimizing fuel consumption or reducing time en-route. The purpose of this cruise speed survey simulation is to analyze performance and energy consumption parameters across various cruise speeds.

The simulation was conducted using the initial parameters and configuration as specified in Table(5.2). The typical mission, from the departure airport to the destination airport was considered for the simulation. Similarly to the altitude survey simulation, this simulation followed a mode schedule where the first x% of cruise in HYB mode while the remainder of the cruise in EL mode, x mode for the approach and landing. No headwind under ISA atmospheric condition was considered for the simulation. Four different cruise speed values ranging from x KCAS to x KCAS in increments of 10 kts were considered for the simulation.

The simulation result obtained for the mission profiles with the varying cruise speeds are shown in Figure(5.30).

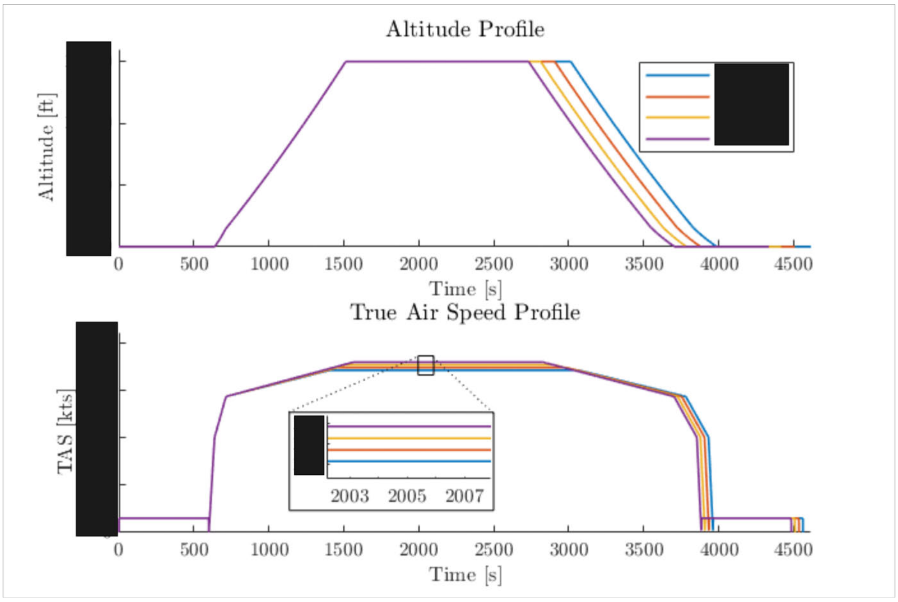


Figure 5.30: Cruise speed survey: Mission profile.

Based on the mission profile results, the aircraft maintains the same altitude in all four scenarios but with varying CAS values, resulting in different TAS values at cruise phase. The reason TAS exceeds KCAS is due to the lower air density at cruise altitude. These different TAS values lead to varying flight times for the four scenarios. Specifically, the scenario with a cruise speed of x KCAS has the minimum flight time, while the scenario with x KCAS has the maximum flight time.

The simulation results for the aircraft-level required thrust and the propulsion power are plotted in Figure(5.31).

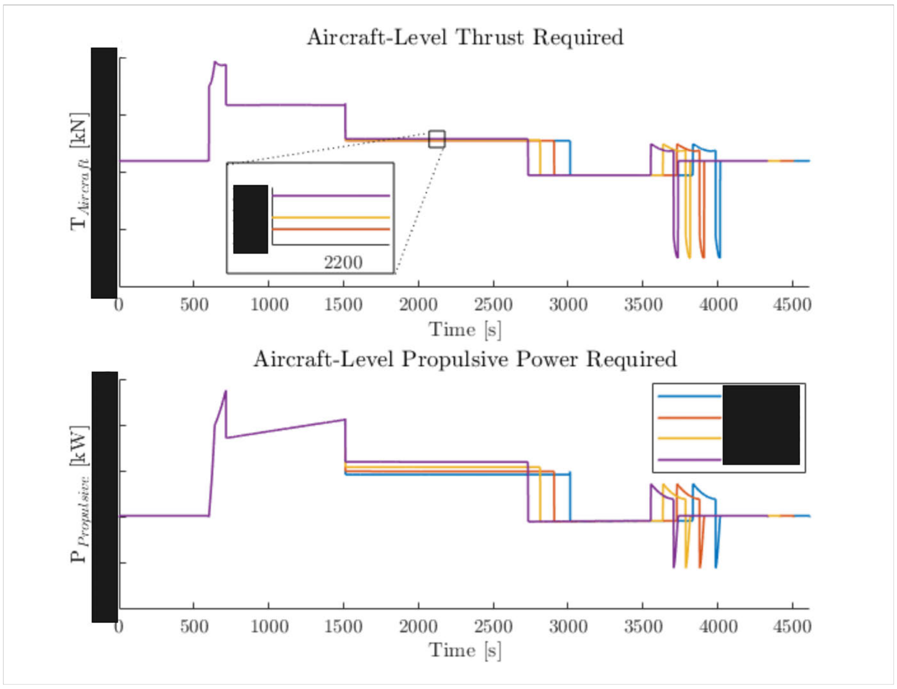


Figure 5.31: Cruise speed survey: Aircraft-level required thrust and propulsive power.

Figure (5.31) indicates that the maximum required thrust during the cruise phase corresponds to cruising at x KCAS, while the minimum occurs when cruising at x KCAS. The required thrust at x KCAS is higher than at x KCAS. This behavior arises because of the behaviour of induced and parasite drag. The induced drag is higher at lower TAS while it reduces with increasing TAS. Conversely, the parasite drag is low at low TAS and increases with TAS. As a result of that, the total drag is higher at both low and high TAS as illustrated in Figure(3.9). This results in higher thrust demand at lower and higher TAS. Additionally, the required propulsive power is higher when cruising at higher TAS, as evident from the plot.

The simulation results for PBU SoC and the remaining FoB values throughout the flight, corresponding to different cruise speeds, are depicted in Figure (5.32).

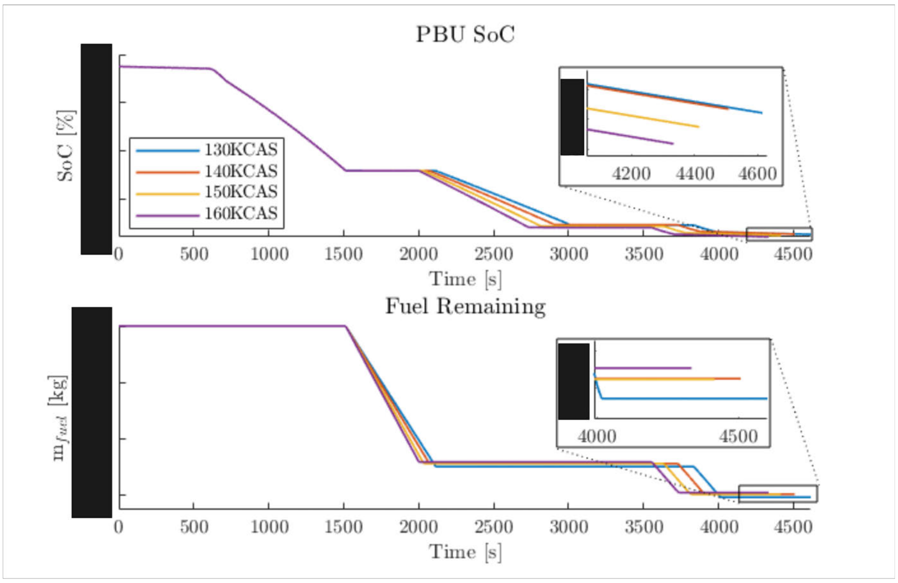


Figure 5.32: Cruise speed survey: PBU SoC and remaining FoB.

Based on the SoC plot for the simulation, it is observed that the rate of SoC depletion is higher when cruising at higher TAS, resulting in a lower SoC value at the end of the flight. The fuel consumption rate is directly related to the delivered power, and the higher cruise TAS corresponds to increased power demand, as evident from the propulsive power plot. Consequently, fuel consumption is also higher at higher cruise TAS. The simulation results for turbine engine fuel consumption are depicted in Figure (5.33).

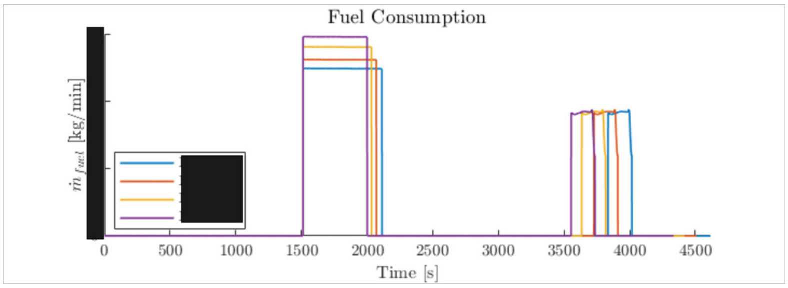


Figure 5.33: Cruise speed survey: Turbine engine fuel consumption.

The cruise speed survey results for PBU SoC and remaining FoB values at the end of flight were supplemented with additional simulation data. This allowed deriving results corresponding to CAS values ranging from x kts to x kts in 5 kts increments. Curve fitting using polynomials of degree 3 was employed and the resulting plot is shown in Figure(5.34).

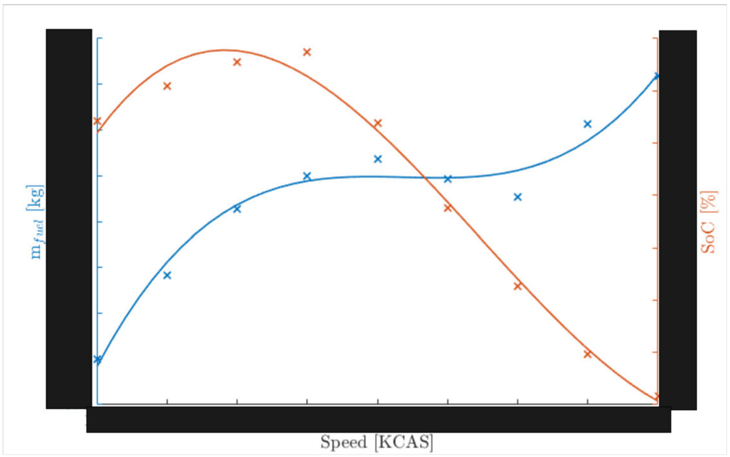


Figure 5.34: Cruise speed survey: PBU SoC and FoB at the end of flight for different cruise speeds.

According to this result, the lower cruise speeds lead to both lower SoC and FoB, while the higher cruise speeds results in higher FoB but lower SoC for the considered mode schedule. Based on these findings, it is evident that the cruise speeds in the range of x KCAS - x KCAS are optimal in-terms of overall energy consumption for the given mission profile and the ambient conditions, as well as for the provided mode schedule. Higher CAS can achieve greater FoB and shorter flight time, but at the cost of increased PBU energy consumption.

5.2.5 Effect of Accounting for Varying Aircraft Weight

One important aspect of the simulation model, in contrast to most existing simulation tools, is that it accounts for varying aircraft weight during the simulation. The focus of this simulation is to analyse the impact of considering the variation of aircraft weight for short-haul flight. The impact of considering aircraft weight variation on simulation results was analyzed by simulating the same typical mission both with and without accounting for varying weight and comparing the results for these two scenarios.

The Figure(5.35) shows the simulation result of the aircraft weight over the time of the flight.

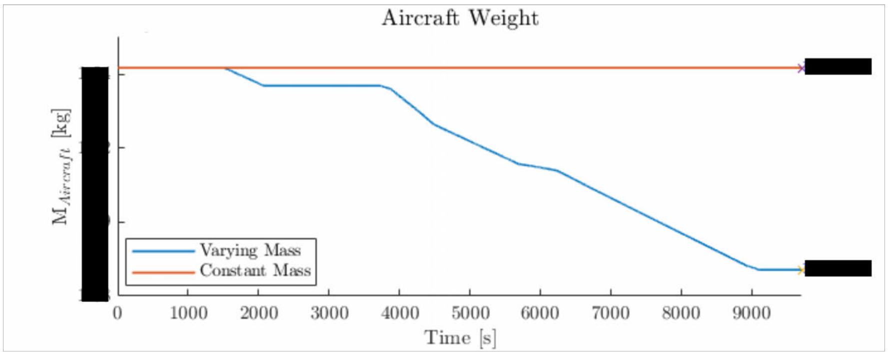


Figure 5.35: Impact of accounting for varying weight: Aircraft weight.

Referring to Figure(5.35), in the constant mass scenario, the aircraft weight remains steady at the takeoff weight of x kg. However, in the varying mass scenario, the aircraft weight gradually decreases due to fuel consumption, resulting in a final weight of x kg at the end of the flight.

The aircraft-level thrust required and propulsive power required is plotted in Figure(5.36). Notably, the plots for varying mass and constant mass scenarios closely overlap. Although a slight disparity emerges after the missed approach, this difference appears negligible. Consequently, we conclude that considering the varying mass model has not significantly altered the requirements for thrust and power.

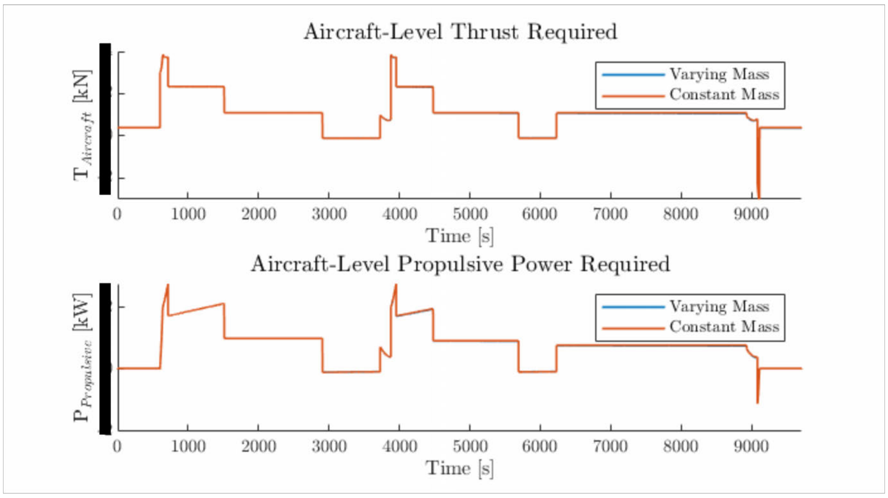


Figure 5.36: Impact of accounting for varying weight: Aircraft-level thrust required and propulsive power required.

The results for accounting for changes in aircraft weight during the simulation on the PBU SoC depicted in Figure(5.37). The plot ΔSoC represents the difference between the SoC results when considering varying aircraft weight and when ignoring weight changes. Notably, the simulation result plots for SoC in both scenarios closely overlap. The difference in SoC at the end of the flight is x% for this flight mission.

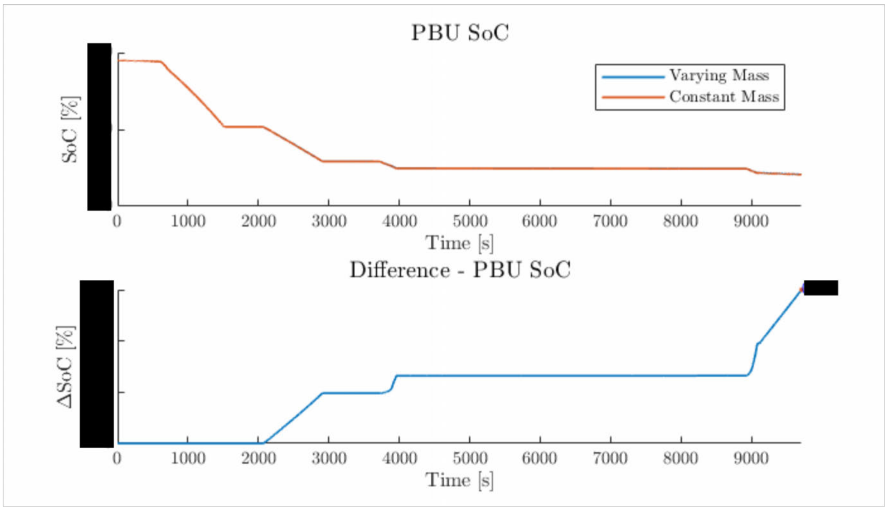


Figure 5.37: Impact of accounting for varying weight: PBU SoC.

Figure(5.38) illustrates the impact of accounting for changes in aircraft weight during the simulation on the FoB. The Δm_{fuel} indicates the difference between the remaining FoB result when considering varying aircraft weight versus assuming constant aircraft weight. Initially, the two plots overlap, but after the go-around, noticeable differences emerge. This discrepancy occurs because the flight operates in HYB mode after go-around, leading to fuel burn. Moreover, the remaining fuel onboard at the end of the flight differs by x kg.

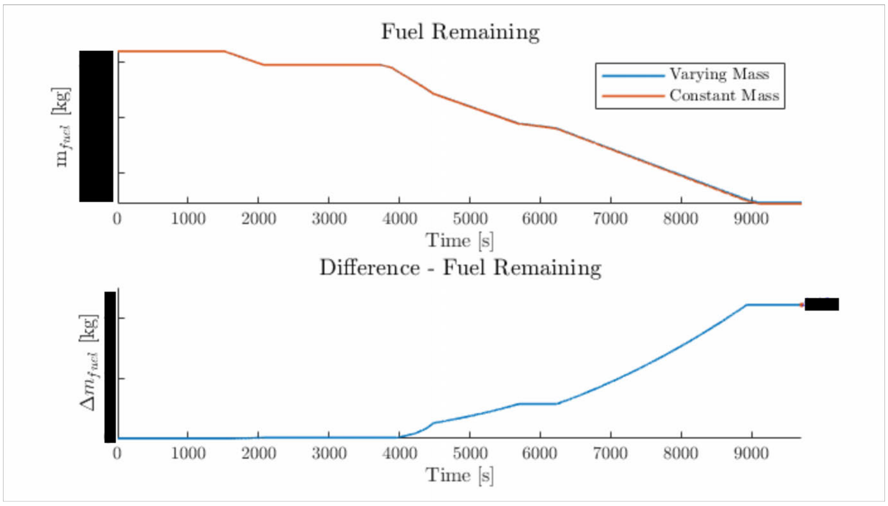


Figure 5.38: Impact of accounting for varying weight: Fuel onboard.

5.2.6 Effect of Weather Parameters

One of the novelties of the simulation model compared to the available tools in the literature is the Weather Model that enables the user to simulate various atmospheric conditions and explore their impact on aircraft missions. Specifically, different simulation experiments were conducted to analyze the impact of the following three parameters on the simulation results.

1. Impact of headwind
2. Impact of DISA
3. Impact of QNH

The simulation setup and the simulation results of each of the above scenarios are presented below.

5.2.6.1 Impact of Headwind

The focus of this simulation experiment is to assess the impact of the wind conditions to the flight mission and the SoE of the aircraft. The typical mission was simulated for ISA atmospheric conditions (QNH = 1013.25 hPa and $T_0 = 15\,^{o}C$) with wind conditions varying from 20 kts of tailwind to 20 kts of headwind in steps of 10 kts. The simulation result for mission profile is shown in Figure(5.39).

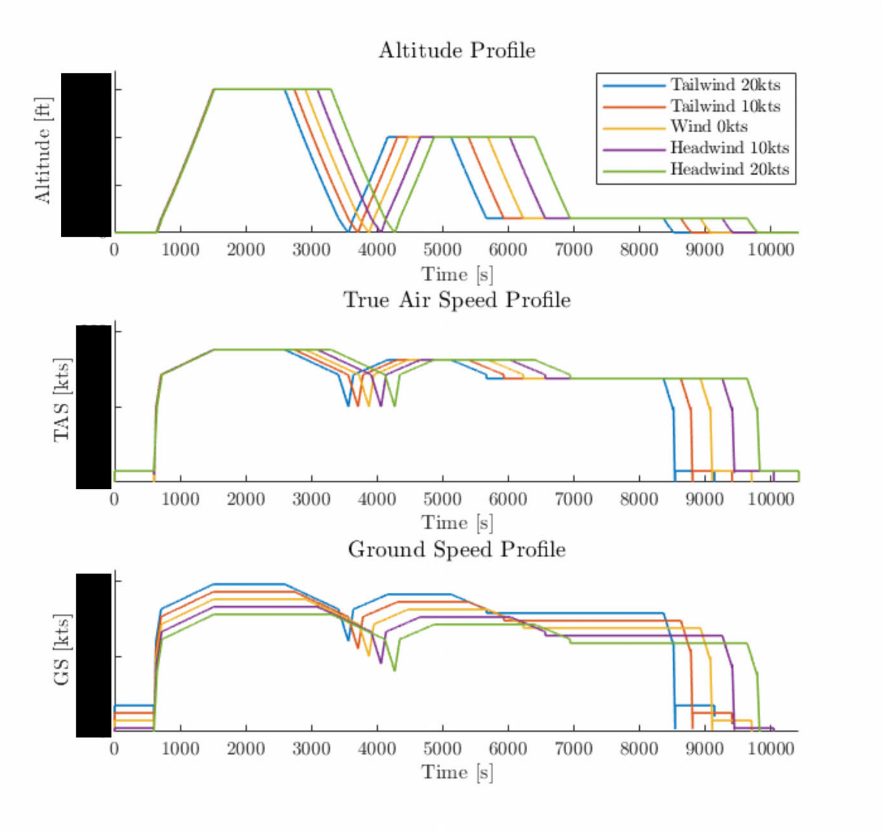


Figure 5.39: Impact of headwind: Mission profile.

The mission profile includes subplots for altitude, aircraft TAS, and ground speed (GS) of the aircraft. Notably, varying wind conditions have resulted in different total flight times. The flight times corresponding to different wind condition from the simulation results are presented in the Table(5.3).

Table 5.3: ES-30 typical mission flight times for different wind conditions.

Wind [kts]	Flight Time [s]
Tailwind 20	x
Tailwind 10	x
Wind 0	x
Headwind 10	x
Headwind 20	x

The flight time values in the table are supported by the GS plot from the simulation results. When encountering a tailwind condition, the ground speed of the aircraft increases and the aircraft takes less time to travel the same distance, leading to a shorter flight time. Conversely, in the presence of headwind, the GS decreases, resulting in an extended duration for flying the same mission. Notably, since the CAS for each flight segments are the same across the different scenarios, the presence of headwind or tailwind does not impact the TAS as observed in the plot.

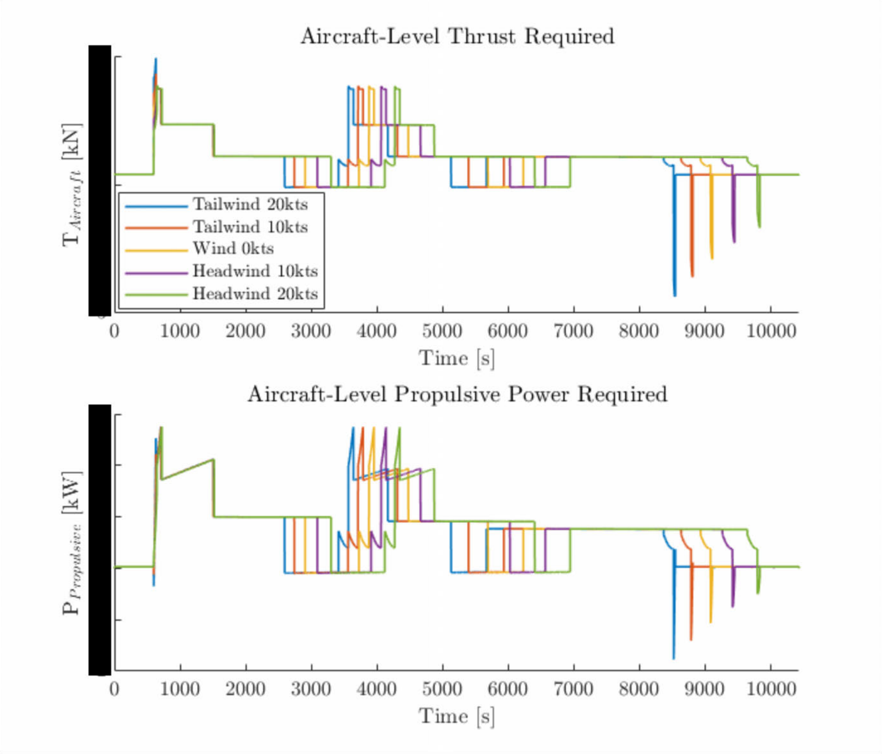


Figure 5.40: Impact of headwind: Aircraft-level required thrust and propulsive power.

The Figure(5.40) depicts the simulation results illustrating the aircraft-level required thrust and propulsion power under various wind conditions. The highest thrust demand occurs during take-off, particularly when the 20 kts tailwind presents. The take-off distance specified in the mission profile for all the cases is the same. Therefore the takeoff GS of the aircraft when tailwind presents is higher as the take-off GS is equal to the sum of takeoff TAS and the tailwind component. As a result of that, the aircraft has to accelerates to the this high GS during the takeoff roll leading to this higher thrust required observed in the plot. Conversely, the thrust demand diminishes when facing headwinds, as the take-off GS under such conditions equals the take-off TAS minus the headwind component. Therefore, the acceleration of the take-off roll is low when the headwind presents, resulting the low thrust value in the plot. This phenomenon becomes more pronounced during the landing phase. When the tailwind presents, the landing GS is increased necessitating a more aggressive deceleration. Conversely, when encountering headwinds, the landing ground speed decreases, resulting in a relatively lower deceleration during the landing roll. For the remaining phases of the flight, the headwind does not affect the required thrust. Because, the TAS is not affected by the headwind and the CAS values corresponding to each phase across the simulations are identical.

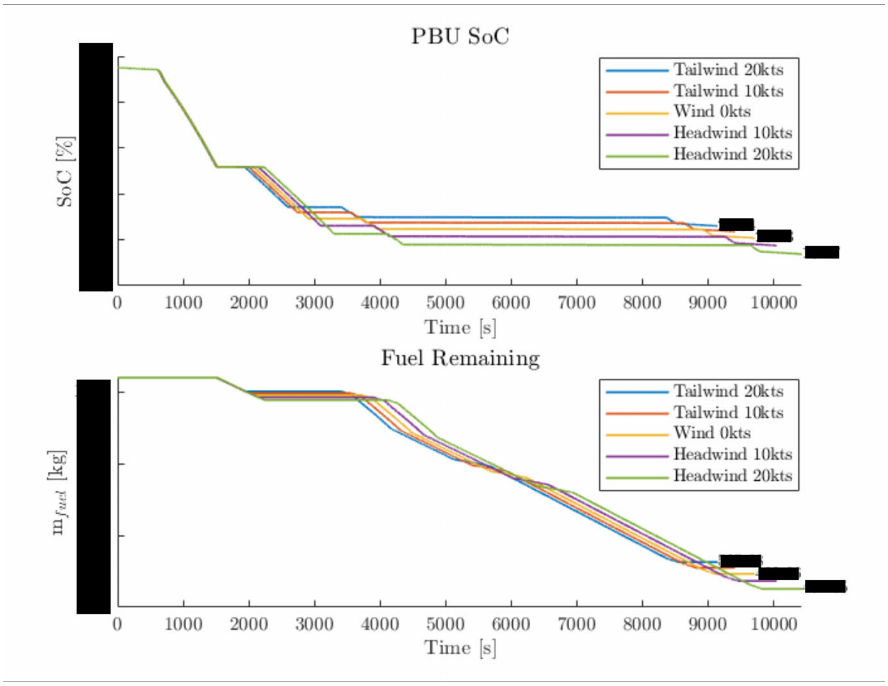


Figure 5.41: Impact of Headwind: SoC and FoB.

The Figure(5.41) consists of PBU SoC and remaining FoB subplots for the simulation under various wind conditions. Moreover the plot depicts the final SoC and FoB values for tailwind 20 kts, wind 0 kts and headwind 20 kts scenarios. The SoC and FoB values for all the wind conditions considered for the simulation are summarized in Table(5.4).

Table 5.4: End of flight SoC and FoB values for the ES-30 typical mission under different wind conditions.

Wind [kts]	**SoC [%]**	**FoB [kg]**
Tailwind 20	x	x
Tailwind 10	x	x
Wind 0	x	x
Headwind 10	x	x
Headwind 20	x	x

When there is no headwind, the PBU SoC and the FoB at the end of the flight are x% and x kg, respectively. With a 20 kts tailwind, the decreased flight time has resulted x% SoC and x kg FoB at the end of the flight. Conversely, with a x kts headwind, the extended flight time has resulted in higher energy usage resulting SoC and FoB at the end of the flight to take x% and x kg respectively. The aircraft SoE is significantly impacted by the headwind conditions, as evident from these simulation results.

5.2.6.2 Impact of DISA

The impact of temperature deviation from ISA conditions (DISA) to the flight mission and the SoE of the aircraft is the focus of this simulation experiment. No wind condition (Headwind = 0kts) and ISA atmospheric pressure (QNH = 1013.25hPa) was considered for the simulation. The ES-30 typical mission was simulated with the simulation model for temperature deviation values of -30^oC to $+20^o$C with steps of 10^oC. The simulation result for the mission profile for various temperature DISA conditions is shown in Figure(5.42). The total flight time for each scenario is summarized in Table(5.5).

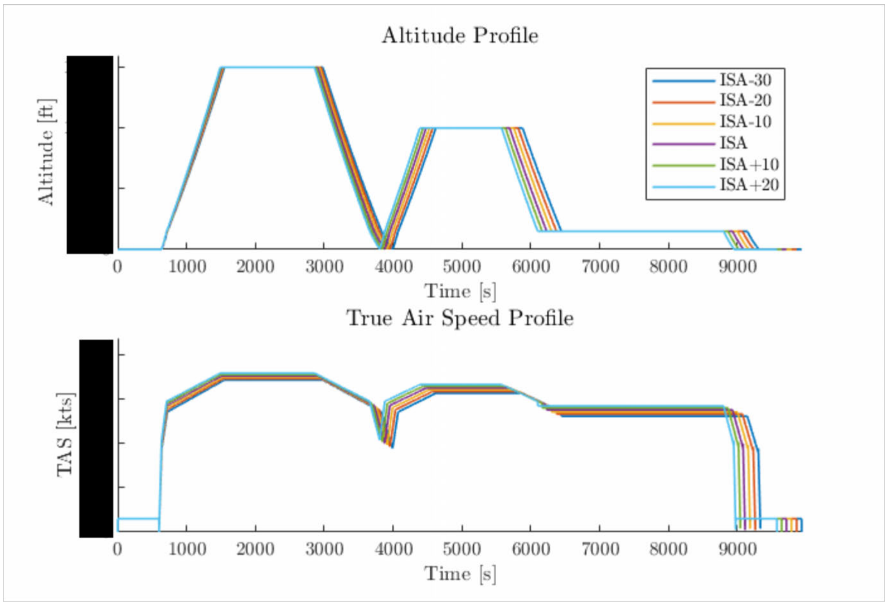


Figure 5.42: Impact of DISA: Mission Profile.

Table 5.5: ES-30 typical mission flight times for different DISA values.

DISA [^{o}C]	Flight Time [s]
-30	x
-20	x
-10	x
0	x
+10	x
+20	x

According to the Figure(5.42) and the Table(5.3), the longest flight time occurs under ISA-30 condition, with a duration of x seconds. In contrast, when flying under ISA+20 atmospheric condition, the same mission is completed in x seconds. Notably, the aircraft TAS is consistently highest during ISA+20 scenarios compared to other conditions. This TAS difference directly contributes to the shorter flight time observed in ISA+20 scenario. The underlying reason becomes evident when the simulation results for ambient conditions are examined, specifically OAT and air density, as depicted in Figure(5.43).

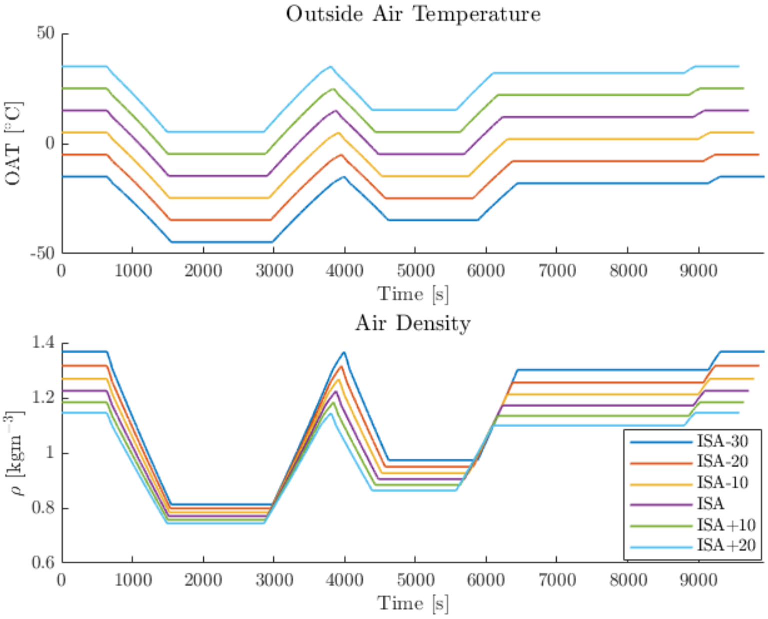


Figure 5.43: Impact of DISA: OAT and Air Density.

According to the Figure(5.43), the OAT values are highest in ISA+20 scenario and lowest in ISA-30 scenario. Furthermore, the air density values are higher in ISA-30 scenario while the air density values are lowest in ISA+20 scenario. From this result, it can be observed that when the OAT is higher, air density decreases and conversely, lower OAT leads to higher air density. Now, considering that the aircraft CAS for the corresponding mission segments across the different scenarios is the same, for a given CAS, the corresponding TAS is lower when air density is high and TAS is higher when air density is low. This insight explains the observed flight times: the

ISA-30 scenario results in longer flight times due to the higher air density, while the ISA+20 scenario yields shorter flight times because of the lower air density specially in the cruise phase of the flight.

The simulation results for required thrust and propulsive power is shown in Figure(5.44).

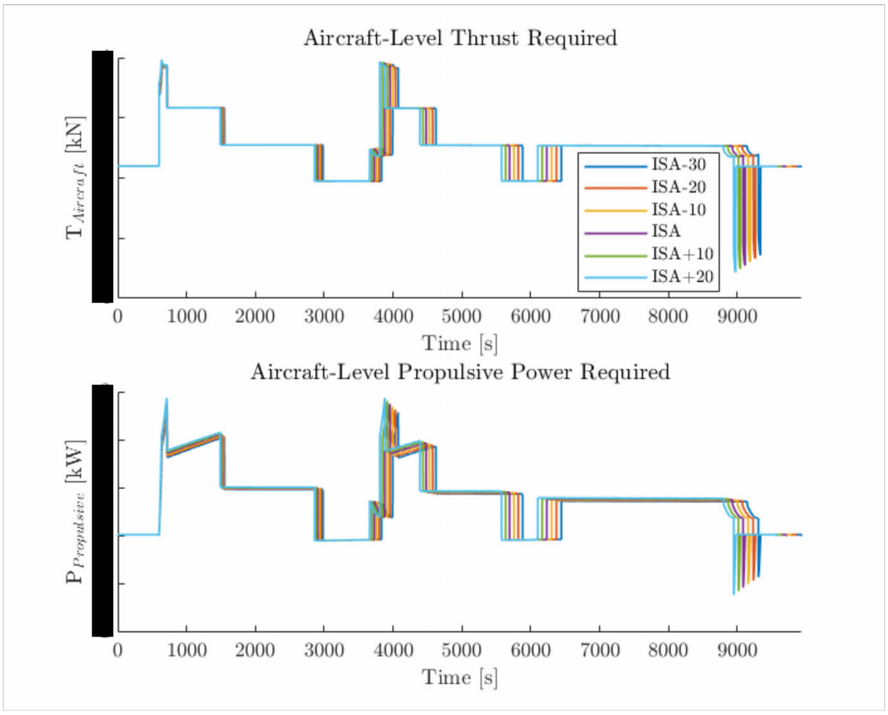


Figure 5.44: Impact of DISA: Aircraft-level thrust and propulsive power.

From the simulation results, it is evident that the thrust and power required are highest in the ISA+20 scenario compared to the other scenarios. This phenomenon is attributed to the ambient conditions affecting the aircraft performance. The higher TAS in ISA+20 scenario results more drag which leads to greater required thrust force and propulsive power.

The simulation results for PBU SoC and FoB of the aircraft for different DISA conditions are presented in Figure(5.45).

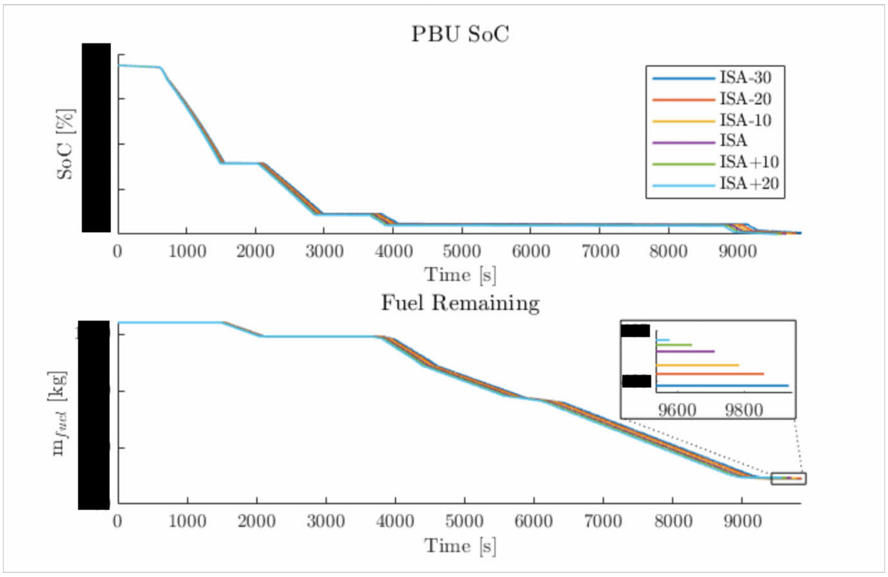


Figure 5.45: Impact of DISA: SoC and FoB.

The end of flight PBU SoC and FoB values are summarized in Table(5.6).

Table 5.6: End of flight SoC and FoB values for the ES-30 typical mission under different temperature DISA conditions.

DISA [^{o}C]	**SoC [%]**	**FoB [kg]**
-30	x	x
-20	x	x
-10	x	x
0	x	x
+10	x	x
+20	x	x

At ISA-30, the PBU SoC and remaining FoB values at the end of flight are x% and x kg, respectively. Interestingly, despite the longer flight time during ISA-30, the power requirement is lower, especially during takeoff and initial climb. Consequently, the SoC remains high at the end of the flight. However, the extended flight duration leads to increased fuel consumption in HYB mode, resulting in a lower FoB.

In the ISA+20 scenario, the SoC and FoB values at the end of the flight are x% and x kg, respectively. The higher power demand during ISA+20 causes the SoC to reach law value by the end of the flight compared to the other DISA conditions. Furthermore, the shorter flight duration has resulted the higher FoB value at the end of the flight for this condition.

5.2.6.3 Impact of QNH

The objective of this simulation experiment is to investigate the influence of atmospheric pressure on the flight mission and the SoE of the aircraft. The typical mission was simulated under no wind conditions (Headwind = 0 kts) and at ISA temperature (DISA = 0 °C), considering five different QNH values ranging from 985 hPa to 1040 hPa.

The simulation result for mission profile obtained from the simulation is shown in Figure(5.46).

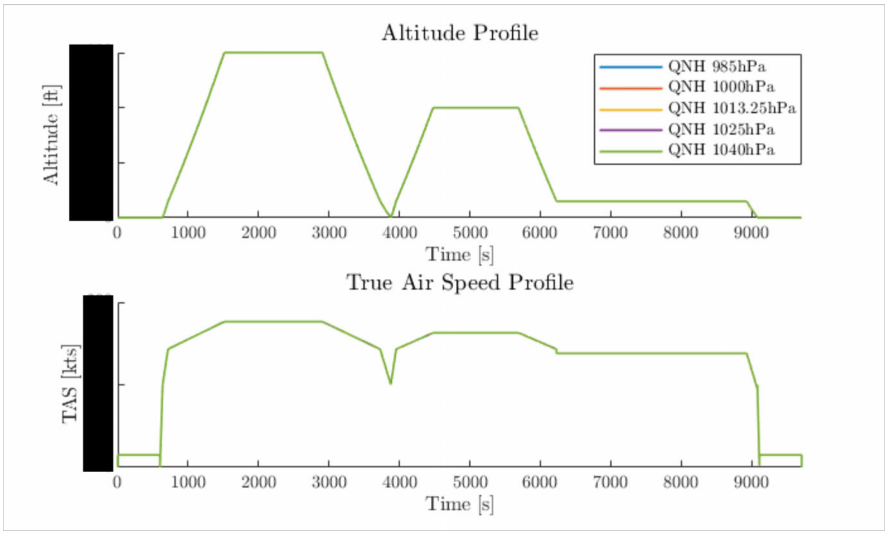


Figure 5.46: Impact of QNH: Mission Profile.

As we can observe, the altitude and TAS profiles from the simulation overlap for various QNH conditions. Consequently, the different QNH values have not affected either the TAS profile or the altitude profile when the headwind and temperature conditions remain constant.

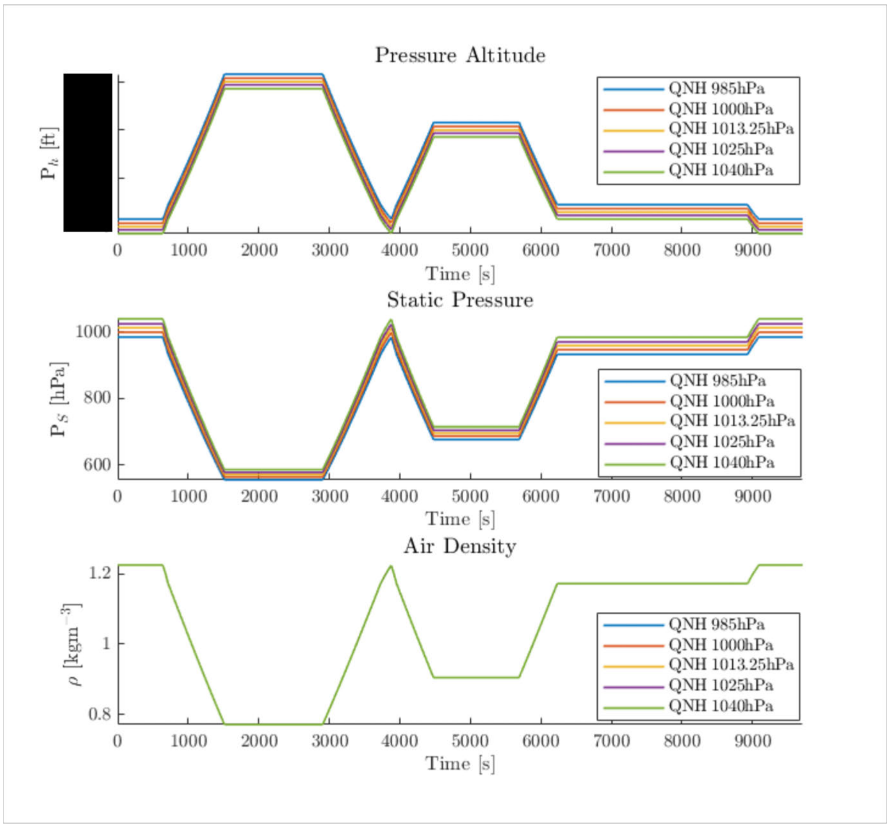


Figure 5.47: Impact of QNH: Ambient parameters.

The simulation results for ambient parameters, specifically pressure altitude, static pressure, and air density, are depicted in Figure (5.47). Notably, in the QNH 985 hPa scenario, pressure altitudes are higher compared to the other QNH scenarios. Conversely, the static pressure at aircraft altitude is greater in the QNH 1040 hPa scenario. Since air density is a function of ambient temperature, and the temperature remains consistent across all simulation conditions, the air density plots overlap for all QNH conditions.

The aircraft-level required thrust and aircraft propulsive power plots from the simulation is shown in Figure(5.48). The required thrust and propulsive power across different QNH conditions overlap according to the simulation result. The reason for this is the fact that the mission profiles and the corresponding air density profiles are the same regardless of the different QNH values as seen from the above simulation results.

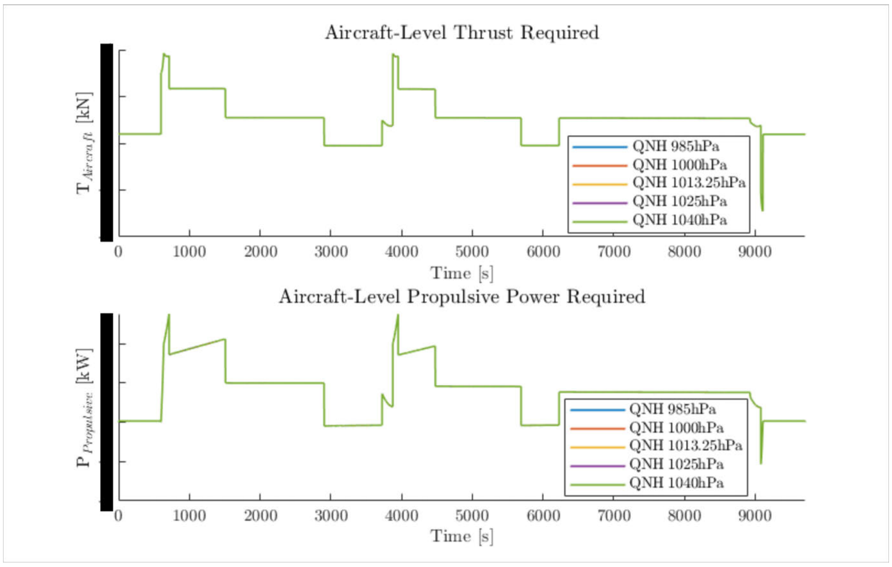


Figure 5.48: Impact of QNH: Aircraft-level required thrust and propulsive power.

The simulation results for PBU SoC and FoB evolution under different QNH conditions are shown in Figure (5.49).

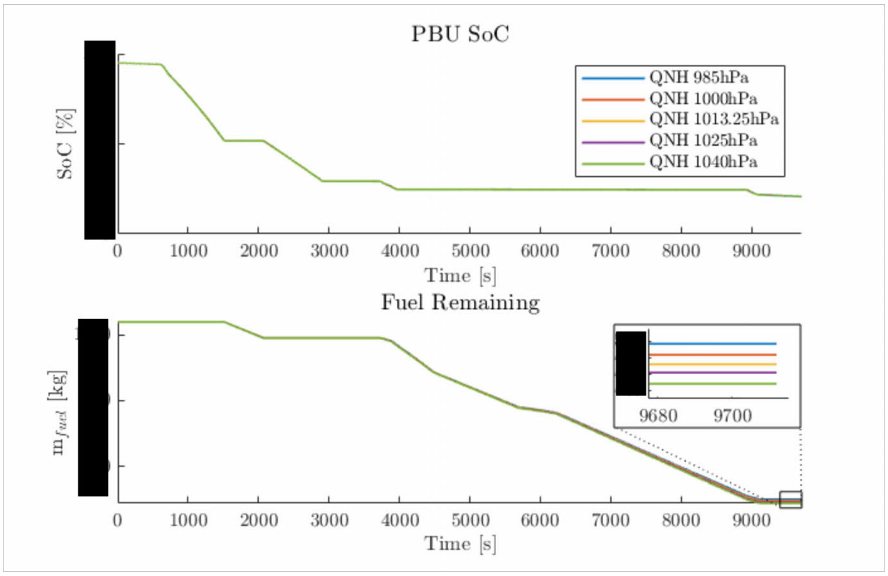


Figure 5.49: Impact of QNH: SoC and FoB.

The SoC evolution remains consistent regardless of the QNH condition. However, the FoB values at the end of the flight are higher for the QNH 985 hPa condition compared to the other scenarios. This discrepancy arises from the fact that the fuel consumption of the turbine engine depends on the pressure altitude. Since the pressure altitude for QNH 985 hPa is greater, the fuel consumption is more efficient, resulting in lower fuel consumption and a higher FoB value compared to the other scenarios. The end of flight PBU SoC and remaining FoB values for each QNH condition are summarized in Table(5.7).

Table 5.7: End of flight SoC and FoB values for the typical mission under different QNH conditions.

QNH [hPa]	**SoC [%]**	**FoB [kg]**
985	x	x
1000	x	x
1013.25	x	x
1025	x	x
1040	x	x

5.3 Optimal Energy Management Strategy

5.3.1 DP-based Optimal Mode Schedule

The simulation experiment aimed to analyze the energy management strategy resulting from the dynamic programming (DP) based optimization algorithm, in comparison with a user-specified mode schedule. Initially, the regional hybrid aircraft typical mission was simulated under ISA atmospheric conditions with no headwind, using the simulation model, incorporating the following parameters and initial conditions.

Table 5.8: Initial parameter values for the simulation to analyze DP-based optimization algorithm.

Parameter	**Value**
Aircraft ZFW	x kg
Initial FoB	x kg
Initial Battery SoC	85%

Further, during the cruise segment, the initial x% was operated in HYB mode, while the remaining x% of cruise was in EL mode. The simulation model yielded results for mode schedule, PBU SoC and remaining FoB are depicted in Figure(5.50).

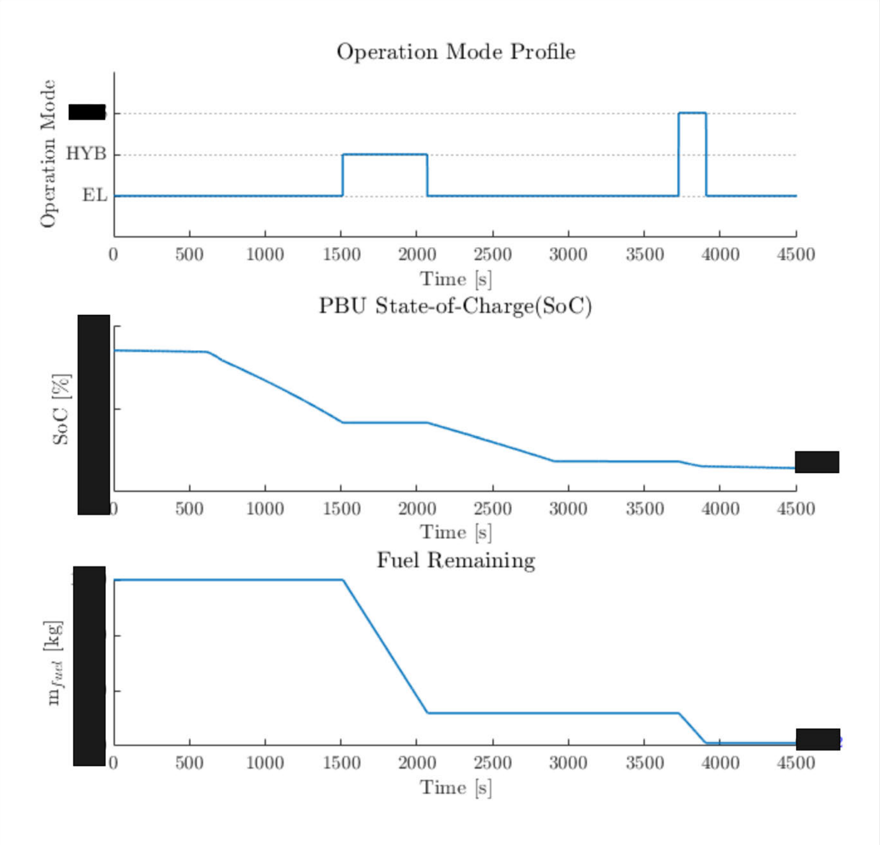


Figure 5.50: Simulation results with with user specified mode schedule.

Based on the simulation results, the final SoC and FoB values for the specified mission, according to the user-defined mode schedule are x% and x kg respectively.

Then, a scenario was considered where the required remaining SoC level at the end of the flight, driven by operational requirements is 20%. In this context, the DP-based optimization algorithm was executed with a 20% constraint on the minimum terminal SoC ($\mathrm{SoC}_{\mathrm{final,min}}$) as specified by Equation(4.30). Additional to the parameters retrieved directly from the simulation model along with the mission profile, the discretization related parameters in the Table(5.9) were considered for the DP-based optimization.

Table 5.9: Discretization parameters for the DP solver.

Parameter	**Value**
N_U	2
N_{soc}	201
N_{fob}	201
k_p	0.1
Ts	1 s

The obtained result from the DP-based optimization algorithm for the control and state trajectories are plotted along with the altitude profile of the mission in Figure(5.51).

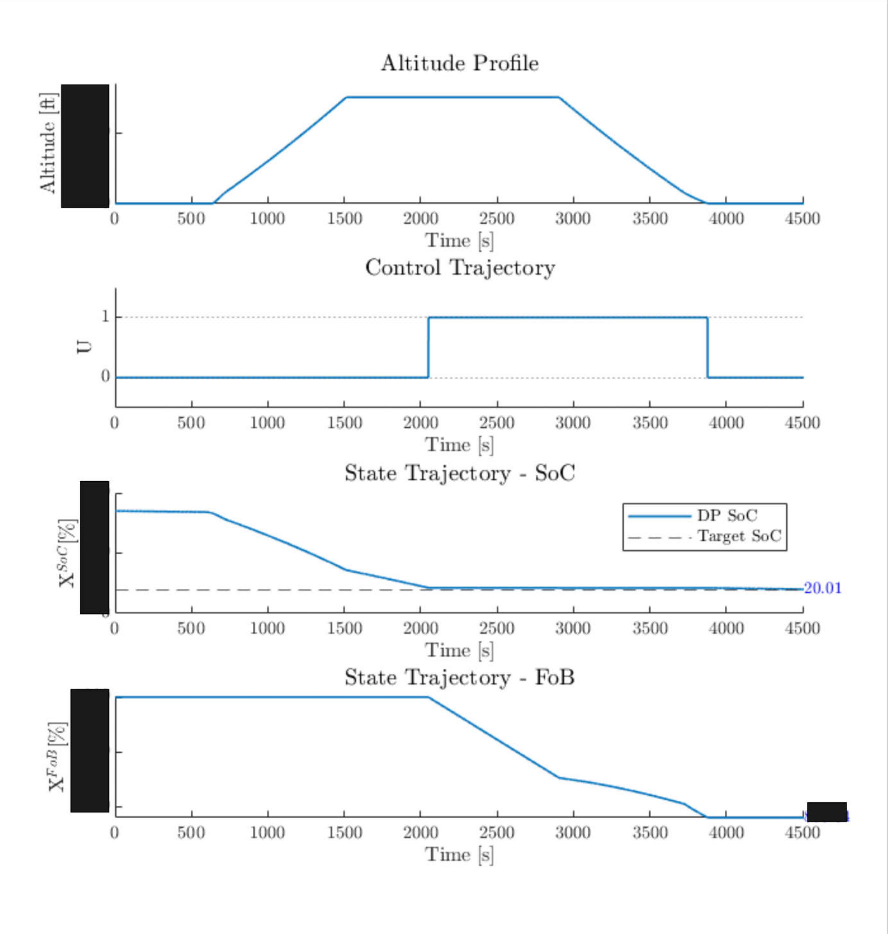


Figure 5.51: Control and state trajectories results from the DP based optimization algorithm.

Based on the DP-based optimization results, the control trajectory indicates that during the initial 38.42% of the cruise phase, the turbine engine remains OFF and the remaining duration of the flight until landing, the turbine engine is ON. At the end of the flight, the PBU SoC has reached the minimum permissible SoC value of 20%, and the remaining FoB is x kg.

The optimal trajectory resulted from the DP-based optimization algorithm, from the initial point to the final point, is visualized in the state-space of SoC, FoB, and time in Figure(5.52).

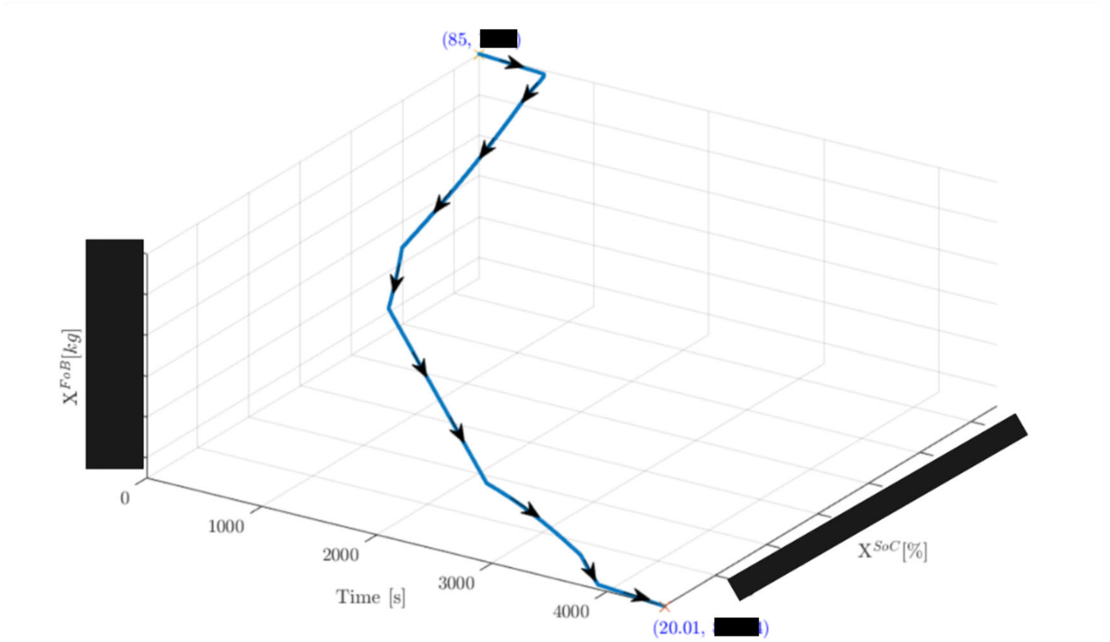


Figure 5.52: The optimal state trajectory in the state-space of SoC, FoB and time.

To evaluate the performance of the optimal mode schedule obtained through the DP-based optimization algorithm, the control trajectory (U) was fed into the simulation model as the mode schedule with x mode enforced for the approach phase and simulated. The simulation results obtained from the simulation model, for PBU SoC and remaining FoB are presented in Figure(5.53), along with the altitude profile and mode schedule for reference.

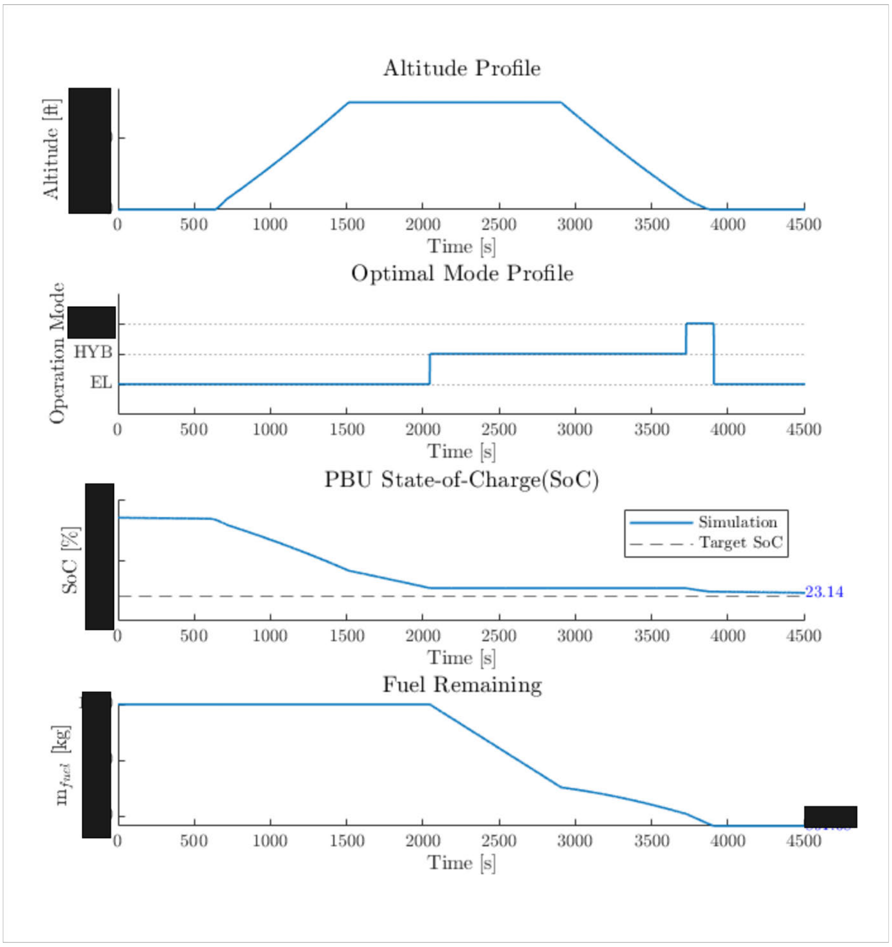


Figure 5.53: SoE evolution with optimal mode schedule from DP-based optimization algorithm.

According to the simulation result, the PBU SoC closely follows the DP result as expected and reaches a final SoC value of 23.14% at the end of the flight. The remaining FoB at the end of the simulation stands at x kg. These results indicate that the optimal mode schedule derived from the DP-based optimization algorithm effectively handles constraints and delivers the expected outcome. However, discrepancies between the SoC and FoB values in the DP state trajectories and the simulation may arise from several factors. Firstly, simplifications made during the conversion of system models into Matlab scripts for the DP solver could introduce discrepancies between the models used in optimization and the Simulink based models in the simulation. Additionally, while the sample time remains consistent between simulation and DP optimization, the discretization of the SoC and FoB state-space involves only 201 points in DP optimization.

To assess the robustness of the DP-based algorithm and its performance against different missions, three different missions were generated by adjusting the mission profile related parameters as follows.

Mission 1: A trip distance of x nm at a cruise altitude of x ft.

Mission 2: A trip distance of x nm at a cruise altitude of x ft.
Mission 3: A trip distance of x nm at a cruise altitude of x ft.

The same initial parameters in Table(5.8) were used for the the simulation while the DP related parameters in Table(5.9) were used for the DP-based optimization. Further, to assess the robustness of the algorithm under varying terminal SoC constraint values, the $\text{SoC}_{\text{final,min}}$ was adjusted to 30%. The optimal mode schedule obtained from the optimisation, for the different missions were fed to the simulation model, with x mode enforced during the approach phase. The simulation results from the simulation model, for the three different missions are plotted in Figure(5.54). Specifically, the altitude profile is plotted along with the mode schedule, PBU SoC and the remaining FoB plots.

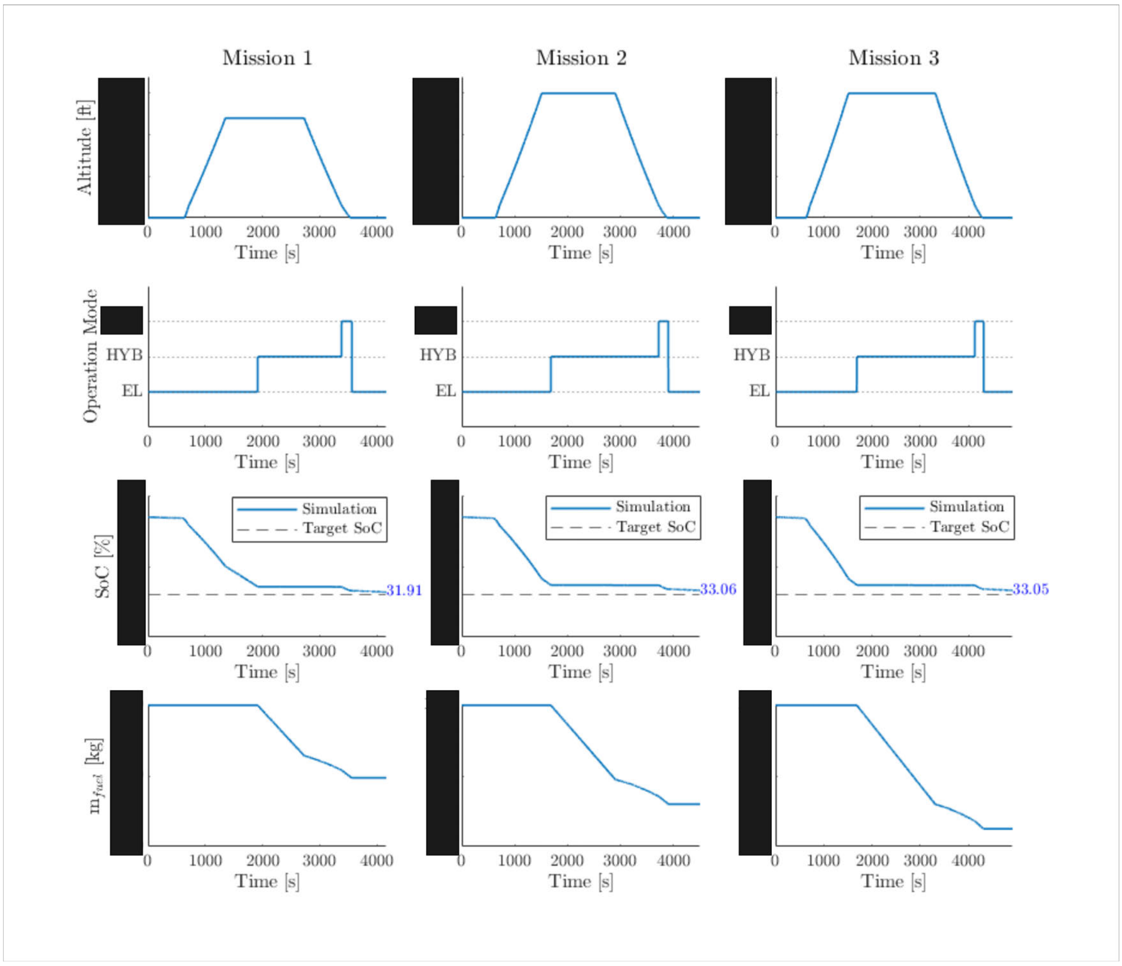


Figure 5.54: Simulation results for different mission profiles with DP-based optimal mode schedules with $\text{SoC}_{\text{final,min}} = 30\%$.

Based on the simulation results, it is evident that the PBU SoC closely approached the $\text{SoC}_{\text{final,min}}$ of 30% for all three missions. However, the terminal FoB values varied among the missions. Mission 3, being the longest mission, recorded the lowest terminal FoB, while Mission 1, the shortest mission, had the highest FoB at the end of the flight.

The DP-based optimal mode schedule effectively utilized the permissible PBU energy for each mission, with the remaining energy requirements met through fuel consumption. This outcome underscores the ability of the DP-based algorithm to consistently provide an optimal mode schedule, regardless of changes in the mission profile.

The DP-based optimization process was executed on an Intel Core i7-11800H @ 2.30 GHz processor and the execution time to solve the optimization problem for each mission is summarized in Table(5.10).

Table 5.10: Processing times for the DP-based optimization for different missions.

Mission	**Processing Time**
Mission 1	810 s
Mission 2	870 s
Mission 3	967 s

5.3.2 Impact of Control State Penalizing Factor (k_p)

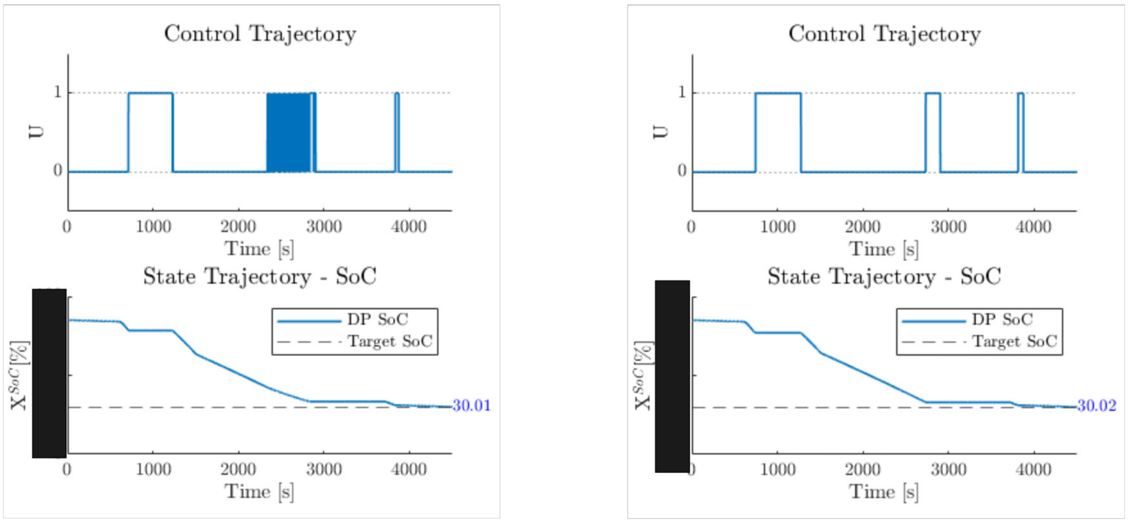
To analyse the effect of the penalizing factor for the control state change (k_p), the same mission was simulated with $\text{SoC}_{\text{final,min}} = 30\%$ and for three different scenarios as follows.

$$\text{Case1}: \ k_p = 0$$

$$\text{Case2}: \ k_p = 0.01$$

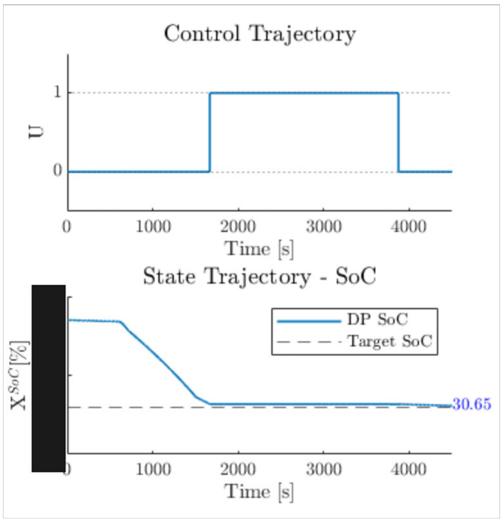
$$\text{Case3}: \ k_p = 0.1$$

The DP optimization result for control state U and PBU SoC X^{SoC} for the three different scenarios are shown in Figure(5.55).



(a) Case1: $k_p = 0$

(b) Case2: $k_p = 0.01$



(c) Case3: $k_p = 0.1$

Figure 5.55: DP-based optimization results for control and SoC trajectories for varying k_p.

Based on the results in Figure(5.55), all three scenarios reached the permissible minimum SoC level. When $k_p = 0$, where there is no penalization on control state changes, the control input U exhibits numerous On/Off occurrences. When $k_p = 0.01$, the penalization factor significantly reduces control state changes. With $k_p = 0.1$, the algorithm ensures that only necessary control state changes are employed to achieve the expected system output ensuring minimum possible control state changes.

The k_p value can be used as the weighing factor of the cost, for turning the turbine engine On/Off during the operation. Turning On/Off the turbine engine incurs certain cost such as the fuel consumed during warm-up time after start, until the engine is operable. This cost can be incorporated to the optimizer algorithm with the k_p value. Using the k_p value, the algorithm can determine whether it is relatively efficient to keep the turbine engine idle, or turn it On/Off based on the energy requirement of the mission.

6

Conclusion

This project centers around designing and evaluating an aircraft performance and mission simulation model for the ES-30, a clean-sheet hybrid-electric aircraft developed by Heart Aerospace. The primary focus of this work lies in pre-flight estimation of the state-of-energy (SoE) function within the Flight Management System (FMS) for hybrid-electric aircraft. The simulation model comprises several critical modules:

- Mission Planner - Enables users to generate detailed flight missions.
- Weather Model - Generates atmospheric conditions for simulation.
- Aircraft Model - Accounts for weight variations due to from fuel consumption.
- Propulsion System Models - Adapted to the novel independent hybrid propulsion architecture of the ES-30.

An inverse simulation approach was used for developing the simulation model, combining physics-based models informed by performance maps and system-specific data. The verification of the simulation model was based on ES-30 typical mission baseline data.

The simulation analysis results reveal several key insights regarding the state-of-energy (SoE) in hybrid-electric aircraft. The short distance flights show negligible SoE sensitivity to varying aircraft weight, however for longer flights weight becomes relevant for accurate SoE estimations. The headwind significantly affects the aircraft SoE, while the atmospheric pressure and temperature also play a role, albeit to a lesser extent and the simulation model is able to quantify these effects.

The simulation model supports development of FMS energy management and performance functions. It can be used to determine the optimal vertical flight profile, minimizing en-route time, fuel consumption or battery energy usage. Specifically, a simulation-based analysis for optimal cruise altitude and cruise speed have been presented and discussed. The simulation model includes simplifications and assumptions, due to absence of actual data, primarily due to proprietary reasons. To achieve greater accuracy, complementing the model with actual system-specific data is essential.

A significant achievement of this work lies in proposing an optimal energy management strategy based on Dynamic Programming (DP). This algorithm generates an ideal mode-of-operation profile for a given flight mission, ensuring that the specified battery energy reserves remain intact while minimizing the fuel consumption. Moreover, the simulation model yields comprehensive data that can be leveraged in developing FMS data tables. Further, it provides a quantitative framework for evaluating mission specific requirements, aircraft and system performance. As such, it becomes a valuable analysis tool for teams specializing in flight science, propulsion, and energy storage. Furthermore, the proposed DP-based energy management strategy can be used as a foundational concept for developing the energy management function within the FMS in ES-30.

Bibliography

[1] International Air Transport Association. "Air Passenger Monthly Analysis – January 2024." IATA Sustainability & Economics, March 6, 2024. www.iata.org/monthly-traffic-statistics.

[2] Ritchie, Hannah. "What Share of Global CO_2 Emissions Come from Aviation?" Our World in Data, 2024. https://ourworldindata.org/global-aviation-emissions.

[3] Adu-Gyamfi, Bright, and Clara Good. "Electric Aviation: A Review of Concepts and Enabling Technologies." Transportation Engineering 9 (2022): 100134. https://doi.org/10.1016/j.treng.2022.100134.

[4] Montlaur, Adeline, Luis Delgado, and César Trapote-Barreira. "Analytical Models for CO_2 Emissions and Travel Time for Short-to-Medium-Haul Flights Considering Available Seats." Sustainability 13, no. 18 (2021): 10401. https://doi.org/10.3390/su131810401.

[5] European Commission, Directorate-General for Mobility and Transport, Directorate-General for Research and Innovation. Flightpath 2050: Europe's Vision for Aviation: Maintaining Global Leadership and Serving Society's Needs. Publications Office, 2011. https://data.europa.eu/doi/10.2777/50266.

[6] Bird, Jonathan Z. "A Review of Electric Aircraft Drivetrain Motor Technology." IEEE Transactions on Magnetics 58, no. 2 (2022): 1-8. https://doi.org/10.1109/TMAG.2021.3081719.

[7] Xie, Ye, Al Savvarisal, Antonios Tsourdos, Dan Zhang, and Jason Gu. "Review of Hybrid Electric Powered Aircraft, Its Conceptual Design and Energy Management Methodologies." Chinese Journal of Aeronautics 34, no. 4 (2021): 432-450. https://doi.org/10.1016/j.cja.2020.07.017.

[8] Doff-Sotta, Martin, Mark Cannon, and Marko Bacic. "Optimal Energy Management for Hybrid Electric Aircraft." IFAC-PapersOnLine 53, no. 2 (2020): 6043-6049. https://doi.org/10.1016/j.ifacol.2020.12.1672.

[9] Zhao, X., Sahoo, S., Kyprianidis, K., Sumsurooah, S., Valente, G., Rashed, M., Vakil, G., Hill, C.I., Jacob, C., Gobbin, A. and Bardenhagen, A., 2019, June. A framework for optimization of hybrid aircraft. In Turbo Expo: Power for Land, Sea, and Air (Vol. 58608, p. V003T06A012). American Society of Mechanical Engineers.

[10] Cameretti, Maria Cristina, Andrea Del Pizzo, Luigi Pio Di Noia, Michele Ferrara, and Ciro Pascarella. 2018. "Modeling and Investigation of a Turboprop Hybrid Electric Propulsion System" Aerospace 5, no. 4: 123. https://doi.org/10.3390/aerospace5040123

[11] Batteh, J., Gohl, J., Sielemann, M., Sundstrom, P., Torstensson, I., MacRae, N., & Zdunich, P. (2018). Development and Implementation of a Flexible Model Architecture for Hybrid-Electric Aircraft.

[12] Donateo, T., Ficarella, A., & Spedicato, L. (2016). Development and validation of a software tool for complex aircraft powertrains. Advances in Engineering Software, 96, 1-13. DOI: 10.1016/j.advengsoft.2016.01.001

[13] Palaia, Giuseppe, and Karim Abu Salem. 2023. "Mission Performance Analysis of Hybrid-Electric Regional Aircraft" Aerospace 10, no. 3: 246. https://doi.org/10.3390/aerospace10030246

[14] Lammen, W.F., & Vankan, W.J. (2019). Electrification studies of single aisle aircraft: A ‘retrofit’ investigation including parallel hybrid electric propulsion (Report No. NLR-TP-2019-312).

[15] Njoya Motapon, S., Dessaint, L.-A., & Al-Haddad, K. (2014). “A Comparative Study of Energy Management Schemes for a Fuel-Cell Hybrid Emergency Power System of More-Electric Aircraft.” IEEE Transactions on Industrial Electronics, 61(3), 1320-1334. doi: 10.1109/TIE.2013.2257152.

[16] Bellman, Richard E. 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.

[17] Bongermino, E., Tomaselli, M., Monopoli, V. G., Rizzello, G., Cupertino, F., & Naso, D. (2017). Hybrid Aeronautical Propulsion: Control and Energy Management. IFAC-PapersOnLine, 50(2), 169–174.

[18] Bradley, T., Moffitt, B., Parekh, D., Fuller, T., & Mavris, D. (2009). Energy management for fuel cell-powered hybrid-electric aircraft. 7th International Energy Conversion Engineering Conference. American Institute of Aeronautics and Astronautics.

[19] Pinto Leite, J.P.S. and Voskuijl, M. (2020), “Optimal energy management for hybrid-electric aircraft,” Aircraft Engineering and Aerospace Technology, Vol. 92, No. 6, pp. 851-861. DOI: 10.1108/AEAT-03-2019-0046

[20] Smith, R. W. (2001). Flight Management Systems. CRC Press LLC.

[21] Federal Aviation Administration. 2021. “Compliance Criteria for 14 CFR section 33.28, Aircraft Engines, Electrical and Electronic Engine Control Systems.” Advisory Circular 33.28-1. [Accessed April 22, 2024]. https://go.usa.gov/xn89k.

[22] Simon, Donald L. “System-Level Control Concepts for Electrified Aircraft Propulsion Systems.” Technical Memorandum. Cleveland, Ohio: Glenn Research Center, 2022. NASA/TM-20210026284.

[23] International Organization for Standardization. Standard Atmosphere. 1975. Ref. No. 2533-1975 (E).

[24] Airbus: Getting to Grips with Aircraft Performance. Published by Airbus Flight Operations Support and Line Assistance; 2002.

[25] Federal Aviation Administration. Pilot’s Handbook of Aeronautical Knowledge. 2008.

[26] EUROCONTROL EXPERIMENTAL CENTRE. User Manual for the Base of Aircraft Data (BADA) Revision 3.12. EUROCONTROL; 2014. EEC Technical/Scientific Report; No. 14/04/24-44.

[27] Alligier, R., Hamed, M. G., Gianazza, D., & Serrurier, M. (2011). Ground-based prediction of aircraft climb: point-mass model vs regression methods. In: Complex World 2011, 1st Annual Complex World Conference.

[28] Lammen, W. F., & Vankan, W. J. (2019). Electrification studies of single aisle aircraft. Netherlands Aerospace Centre NLR.

[29] McCormick BW. Aerodynamics, Aeronautics, and Flight Mechanics. 2nd ed. John Wiley & Sons; 1995.

[30] König, P., Sharma, D., Konda, K.R., Xie, T., Höschler, K. "Comprehensive Review on Cooling of Permanent Magnet Synchronous Motors and Their Qualitative Assessment for Aerospace Applications." Energies 16, no. 22 (2023): 7524. https://doi.org/10.3390/en16227524.

[31] Bouzidi, I., Bianchi, N., and Masmoudi, A. "An Approach to the Sizing of Electric Motors Devoted to Aerospace Propulsion Systems." COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 33, no. 5 (2014): 1527-1540.

[32] De Donno, G. (2017, May 16). High efficiency motors. Metal Working World Magazine.

[33] Pratt & Whitney Canada Corp. “PT6A Turboprop Information.” Pratt & Whitney Canada Customer FIRST Centre. Accessed May 10, 2024. www.pwc.ca.

[34] Chapman, J. W., Lavelle, T. M., & Litt, J. S. (2016). Practical techniques for modeling gas turbine engine performance. In 52nd AIAA/SAE/ASEE joint propulsion conference (p. 4527).

[35] El-Sayed, A.F. (2017). Aircraft Propulsion and Gas Turbine Engines (2nd ed.). CRC Press. https://doi.org/10.1201/9781315156743

[36] Guzzella, L., & Sciarretta, A. (2005). Vehicle Propulsion Systems: Introduction to Modeling and Optimization.

[37] Nejad, S., Gladwin, D. T., & Stone, D. A. (2016). A systematic review of lumped-parameter equivalent circuit models for real-time estimation of lithium-ion battery states. Journal of Power Sources, 316, 183-196. DOI: 10.1016/j.jpowsour.2016.03.042.

[38] Estaller, J., Kersten, A., Kuder, M., Thiringer, T., Eckerle, R., & Weyh, T. (2022). Overview of battery impedance modeling including detailed state-of-the-art cylindrical 18650 lithium-ion battery cell comparisons. Energies, 15(10), 3822. DOI: 10.3390/en15103822.

[39] Madani, S. S., Schaltz, E., & Kær, S. K. (2018). A Review of Different Electric Equivalent Circuit Models and Parameter Identification Methods of Lithium-Ion Batteries. ECS Transactions, 87(1), 23. DOI: 10.1149/08701.

[40] Campagna, N., Castiglia, V., Miceli, R., Mastromauro, R., Spataro, C., Trapanese, M., & Viola, F. (2020). Battery Models for Battery Powered Applications: A Comparative Study. Energies, 13(10), 4085. DOI: 10.3390/en13164085

[41] EUROCAE.ED-309: Guidance on VTOL Energy Level Information Provided to the Crew. Issued in February 2023.

[42] Smita, Smita, Chandrima Kollara Pradeep, Stefan Lundberg, Johan Hellsing, and Giacomo Mingardo. 2023. “Analysis of the Propulsion System of an Electric Aircraft.” Chalmers tekniska högskola, Institutionen för elektroteknik. https://odr.chalmers.se/items/233cde8f-9fb1-4c84-8068-874c7bddcb13

[43] Estaller, Julian, Anton Kersten, Manuel Kuder, Torbjörn Thiringer, Richard Eckerle, and Thomas Weyh. 2022. “Overview of Battery Impedance Modeling Including Detailed State-of-the-Art Cylindrical 18650 Lithium-Ion Battery Cell Comparisons†.” Energies 15 (10). MDPI. doi:10.3390/en15103822.

[44] Wood, A. 2023. Density Altitude Calculator. https://aerotoolbox.com/density-altitude

[45] E6BX. Pressure Altitude Calculator. https://e6bx.com/pressure-altitude

[46] O. Sundstrom and L. Guzzella, "A generic dynamic programming Matlab function," 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), St. Petersburg, Russia, 2009, pp. 1625-1630, doi: 10.1109/CCA.2009.5281131.

[47] International Civil Aviation Organization. Manual of the ICAO Standard Atmosphere: extended to 80 kilometers (262,500 feet). 1993.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se