

This is a postprint of the following article:

Shugo Kaneko, Maxfield Arnson, Hannah Hajdik, Gokcin Cinar, Joaquim R. R. A. Martins, and Alejandra Uranga. Gradient-Based MDAO Framework for Advanced Aircraft Concepts. *AIAA SciTech Forum*, 2026.

The published article may differ from this postprint and is available at:

<https://arc.aiaa.org/doi/10.2514/6.2026-1302>.

# Gradient-Based MDAO Framework for Advanced Aircraft Concepts

Shugo Kaneko<sup>a,1</sup>, Maxfield Arnson<sup>a,2</sup>, Hannah Hajdik<sup>b,3</sup>,  
Gokcin Cinar<sup>a,4</sup>, Joaquim R. R. A. Martins<sup>a,5</sup>, and Alejandra Uranga<sup>b,6</sup>

<sup>a</sup>*University of Michigan, Ann Arbor, MI, 48109*

<sup>b</sup>*Electra, Manassas, VA, 20110*

## Abstract

Aircraft that incorporate novel technologies have the potential to unleash step changes in energy efficiency. These novel aircraft are highly integrated, can take advantage of electrification and closer airframe-propulsion integration, and leverage complex interactions between subsystems and new possibilities for aircraft operations. To support the need for advanced aircraft conceptual design studies, we develop and integrate a series of subsystem models in a multidisciplinary design analysis and optimization (MDAO) framework. Our framework, built upon NASA's OpenMDAO and Aviary, leverages gradient-based optimization to efficiently explore the expanded aircraft design space that new technologies offer. The framework includes component models of various subsystems, including electrified propulsion, batteries, thermal management, aerodynamics, and aero-propulsive interactions. This paper presents the system-level integration of these subsystem models into an aircraft-level MDAO framework. We demonstrate its application to the conceptual design optimization of electrified aircraft for single-aisle and commuter classes. This MDAO framework will be made open source to contribute to ongoing research on next-generation aircraft design and sustainable aviation.

## 1 Introduction

Advancements in aircraft design will play a major role in a more energy-efficient future for the aviation industry. These novel designs must meet existing standards while exceeding current performance. Early-stage conceptual design studies are crucial for evaluating the viability of new technologies and ensuring their functionality at the system and subsystem levels. Such studies provide

---

<sup>1</sup>Postdoctoral Research Fellow, Department of Aerospace Engineering.

<sup>2</sup>Ph.D. Candidate, Department of Aerospace Engineering. Student Member AIAA.

<sup>3</sup>Senior Multidisciplinary Optimization Engineer.

<sup>4</sup>Assistant Professor, Department of Aerospace Engineering. Senior Member AIAA.

<sup>5</sup>Pauline M. Sherman Collegiate Professor, Department of Aerospace Engineering. Fellow AIAA.

<sup>6</sup>Chief Engineer for Research and Future Concepts. Senior Member AIAA.

confidence in advancing conceptual designs to the preliminary stage, curb risks in premature experimental development, and benchmark requirements for a vehicle that meets future needs without compromising proven performance.

Multidisciplinary design, analysis, and optimization (MDAO) methods efficiently explore the complex aircraft design space, considering various aircraft subsystems and their system-level interactions. MDAO is especially important for next-generation aircraft designs that incorporate a variety of advanced technologies, such as electrified propulsion, which benefits from integration with other systems. It also enables explicit consideration of environmental impact metrics and operational flexibility as part of the design requirements and objectives, appropriately balancing complex design objectives alongside other conventional metrics, such as fuel burn, weight, and cost.

Aircraft conceptual design exploration with an electrified propulsion system has emerged as a promising research topic in the past decades [1]. We use the term *electrified* to designate an aircraft or propulsion system that uses electrical components in the primary thrust-producing chain of the propulsion system. NASA has proposed and extensively studied novel electrified aircraft concepts, including STARC-ABL [2, 3] and SUSAN [4]. Cinar et al. [5, 6] developed a tool for electrified propulsion architecture sizing and synthesis (E-PASS). Brelje and Martins [7] developed OpenConcept, an open-source tool for electric aircraft design and mission analysis using OpenMDAO [8]. Kruger and Uranga [9] presented the library for unified conceptual aircraft synthesis (LUCAS) and discussed design space exploration for electric commuter aircraft. Recently, NASA developed the open-source aircraft conceptual design tool Aviary [10, 11], whose main goal is to support MDAO studies of aircraft concepts.

In this work, we develop a series of aircraft subsystem models and an aircraft MDAO integration framework, focusing on electrified propulsion and closer subsystem integration. A primary goal of our framework is to provide the capability to model aircraft equipped with various advanced technologies that would enable ultra-efficient air transportation, such as the concept shown in Fig. 1. The subsystem model fidelity ranges from first-order approximations, typical in the conceptual design literature, to higher-fidelity physics-based models that allow more detailed design space explorations.



**Figure 1:** Concept aircraft illustrating the advanced technologies being modeled, including electrified propulsion, distributed propulsion, boundary layer ingestion, and blown lift. It serves as a research baseline and does not represent an optimized future configuration.

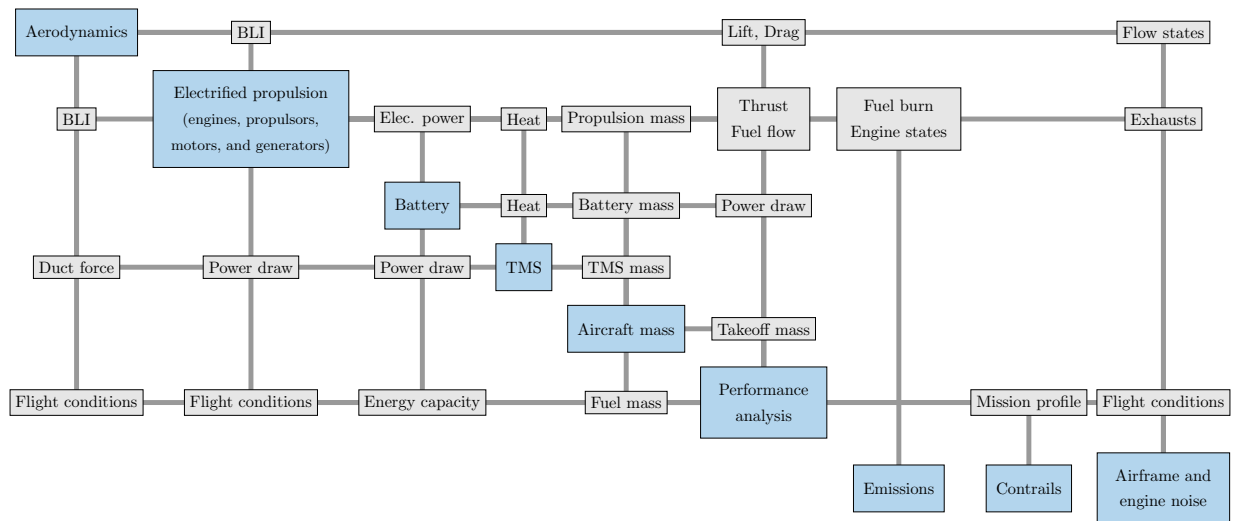
We build our tool on top of NASA’s Aviary [11] and OpenMDAO [8] and utilize their capabilities to perform fully-coupled, large-scale MDAO. Our tool will become open source for public use by the research community. This research effort is supported by NASA’s Advanced Aircraft Concepts for Environmental Sustainability (AACES) 2050 Program, and this paper presents the first version of our MDAO framework. Subsequent publications by the authors and collaborators will present detailed modeling approaches and the applications of the framework to design space exploration.

This paper is organized as follows. Section 2 provides an overview of the aircraft system-level MDAO integration, the underlying methodology, and the aircraft performance analysis. Section 3 presents a brief overview of the subsystem models, which will be detailed in subsequent publications. Finally, Section 4 demonstrates the use of the framework for electrified aircraft design optimizations.

## 2 Aircraft MDAO Framework: System Integration

### 2.1 Overview

To properly assess the performance, economic viability, and environmental footprint of future aircraft concepts, our MDAO framework models system-level interactions of various subsystems, including aerodynamics, electrified propulsion, batteries, and thermal management systems (TMS). Figure 2 shows the relevant subsystems and how they interact with each other. Aerodynamics and propulsion subsystems are coupled by the aero-propulsive technologies we consider, namely boundary layer ingestion (BLI) and blown-lift. An electrified propulsion system involves multiple subsystems, such as engines, motors, and generators, whose couplings are not shown in Fig. 2 but are discussed in Sec . 3.1. The TMS handles the heat produced by the electrified propulsion components and draws power from the powertrain while adding duct drag or thrust. All these subsystems affect the aircraft mass and performance, requiring a fully coupled analysis and optimization at the aircraft level.



**Figure 2:** Aircraft-level coupling of subsystems in MDAO framework. Diagonal blue blocks are major subsystems, and gray lines and off-diagonal blocks show interactions between subsystems.

## 2.2 Gradient-Based Optimization

Design optimization is an effective tool for exploring aircraft design spaces. Conceptual design engineers can perform a series of optimizations with different technology assumptions and/or design objectives to better understand the design space and relevant trade-offs without needing to manually sweep every design variable. In that sense, optimization serves as a “dimensionality reduction” tool that extracts the most important parameters and trends from an otherwise too-high-dimensional design space.

Computational efficiency in design optimization is a key factor in supporting rapid conceptual design studies and is particularly important when the design space is wide and complex. This is often challenging because design optimization problems can have many design variables and constraints, as well as a large number of subsystem models within an aircraft optimization loop. Thus, there is usually a trade-off between model fidelity and computational efficiency. We use two strategies to improve the computational efficiency. First, our framework utilizes gradient-based optimization because it is the most computationally efficient approach for large-scale optimization problems. Secondly, we develop subsystem models with varying fidelity levels, applying higher or lower fidelity models depending on the design space, research questions, and the need for speed versus accuracy when exploring promising regions.

## 2.3 Foundation: OpenMDAO, Dymos, and Aviary

We implement our MDAO framework using NASA’s open-source Python packages: OpenMDAO [8], Dymos [12], and Aviary [11]. Figure 3 shows an overview of the tools used to develop our framework.

OpenMDAO is a gradient-based optimization framework for general MDAO problems. Its mathematical backbone is the *modular analysis and unified derivatives architecture* [13; 14, Sec. 13.2] and the *unified derivative equations* [15; 14, Sec. 6.9]. These enable efficient multidisciplinary analysis and coupled derivative computation using a generalization of the implicit analytic method (the adjoint method). OpenMDAO also has an interface to state-of-the-art gradient-based optimizers, including SNOPT [16] and IPOPT [17] via the pyOptSparse wrapper [18].

Aviary is an aircraft conceptual design tool built on OpenMDAO. Subsystem models in Aviary— aerodynamics, propulsion, geometry, empty weight, and aircraft dynamics—are inherited from

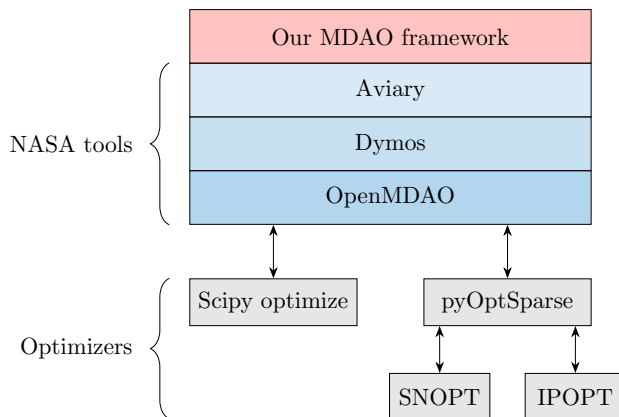


Figure 3: Tools used to develop our aircraft MDAO framework.

NASA’s Flight Optimization System (FLOPS) [19] and the General Aviation Synthesis Program (GASP) [20]. Aviary modernizes the implementation of the FLOPS and GASP models using OpenMDAO. Its modular implementation allows us to incorporate our custom subsystem models (for example, electrified propulsion and TMS models) and combine them with Aviary’s standard models. The outcome is a set of modular capabilities built upon the foundation of Aviary that supports the evaluation of novel aircraft concepts, including more integrated and electrified concepts.

Aviary uses Dymos [12, 21], a trajectory optimization library built on OpenMDAO, for mission analysis and optimization. When solving aircraft design optimization with a fixed mission profile, Aviary uses Dymos as an ordinary differential equation (ODE) solver to integrate time-varying quantities, such as aircraft dynamics, fuel burn, and battery energy usage. Aviary and Dymos also support simultaneous optimization of aircraft design, mission profile, and dynamic controls. For example, we can optimize the dynamic power management of an electrified propulsion system along the flight path while concurrently sizing the aircraft, propulsion components, and batteries.

Using these NASA tools, we develop and implement a series of aircraft subsystem models for electrified aircraft conceptual design with closer subsystem integration. All of our models are formulated with analytic derivatives and implemented or wrapped as OpenMDAO components, enabling fully-coupled gradient-based optimization. Section 3 describes the subsystem models available in our MDAO framework and how we integrate them into Aviary.

## 2.4 Performance Analysis

### 2.4.1 Mission Analysis

We use Aviary’s height-energy mission analysis to compute the mission fuel burn and battery energy consumption [22]. The simplest mission profile would consist of climb, cruise, and descent phases, but Aviary supports more detailed mission parametrization. The mission profile can be either prescribed and fixed, or it can be optimized via the Mach number and altitude profiles to optimally fly an aircraft. Thanks to Aviary’s multi-mission capability, aircraft design can be optimized for a weighted objective across multiple missions (*e.g.*, the most-demanding sizing mission and economy mission), where each mission is defined by a different payload weight, fuel weight, and electrification setting.

Our framework also supports nominal takeoff and landing analysis in addition to the primary climb-cruise-descent mission. Takeoff and landing analysis is crucial when including noise constraints in the design optimization problem. It also computes the fuel burn and battery energy consumption more accurately, which is especially important for short-range aircraft. We use Aviary’s FLOPS-based detailed takeoff and landing capability, which requires a two-degree-of-freedom (2DoF) equations of motion (EoM) model instead of the height-energy model. In our framework, we combine these 2DoF takeoff and landing phases with the primary height-energy mission by connecting the end-of-takeoff states (such as mass, battery states, and component temperature) to the start-of-climb states, and likewise connecting the end-of-descent and start-of-landing states. We chose to combine the 2DoF EoM and the height-energy model instead of using the 2DoF EoM across the entire mission. This is because the height-energy model is more robust and computationally efficient than the 2DoF EoM model.

## 2.4.2 Takeoff Balanced Field Length

We include an extended version of Aviary’s balanced field length (BFL) analysis in our framework. The BFL serves as a sizing constraint for the propulsion system and wing area. Our framework extends Aviary’s BFL analysis capability, which assumes one-engine failure for twin-engine aircraft, to more general failure modes that are relevant to electrified aircraft. Available failure modes include one or multiple component failures for engines, motors, generators, and batteries, as well as the emergency switching of electrification mode (*e.g.*, engine failure and switching to all-electric mode, or vice versa).

## 2.4.3 Other Sizing Conditions

Point performance requirements are added to Aviary’s pre-mission group. These point conditions serve as additional sizing constraints for the sizing of propulsion power, battery power, and wing. By default, we include the sea-level static maximum thrust condition, where the required thrust value is computed based on the maximum takeoff weight (MTOW) and user-provided lower bound on the thrust-to-weight (T/W) ratio. This condition is an inequality constraint, meaning that the aircraft’s T/W is at least the user-provided value, but it may be higher than that if more demanding sizing conditions are present. Users can add other operating points, such as top-of-climb and ceiling conditions.

## 2.5 Optimization Objective Function

Any performance metric discussed in this paper can be used as an objective function of the aircraft’s design optimization. Common metrics for the objective include fuel burn, total fuel and battery energy consumption, takeoff weight, or any weighted sum of these metrics. For multi-mission optimization, users can set any weighting factors between missions.

At this moment, a user is responsible for selecting an appropriate objective function for their design problem. We plan to propose effective objective functions (*e.g.*, an appropriate balance between multiple metrics) for next-generation aircraft design in our upcoming publications.

## 3 Aircraft Subsystem Models

This section discusses the aircraft subsystem models that our MDAO framework currently includes, as summarized in Table 1. We use Aviary’s *external subsystem builder* interface to integrate the majority of the subsystem models; however, we couple the high-fidelity battery model (Sec . 3.5.3) to Aviary at the OpenMDAO layer (what is called the *level-3* interface in Aviary.) Aviary organizes component computations into *pre-mission*, *mission*, and *post-mission* groups. The pre-mission group is responsible for computations prior to mission analysis such as weight buildup, the mission group hosts the mission analysis, and the post-mission group performs post-processing computations. Table 1 indicates which group each subsystem model is integrated into. Some subsystems have several performance models, each corresponding to a different fidelity level. Lower-fidelity models are simpler and computationally faster but often involve calibration constants, while higher fidelity models provide a better representation of the component’s performance in a variety of conditions. Finally, each subsystem’s weight is estimated via a model processed in the pre-mission group.

**Table 1:** Summary of subsystem models

Section	Subsystem	Model	Aviary integration group(s)	Fidelity level
Sec. 3.1	Propulsion system	Unified propulsion system	Pre-mission and Mission	-
Sec. 3.2	Turboshaft/Turbofan core	Constant PSFC	Mission	Low
		Engine deck	Mission	Mid
		pyCycle	Pre-mission and Mission	High
Sec. 3.3	Propulsor	Constant efficiency	Mission	Low
		Propeller	Mission	Mid
		Ducted fan deck	Mission	Mid
		pyCycle ducted fan	Pre-mission and Mission	High
Sec. 3.4.1	Motor and generator	Constant efficiency	Mission	Low
Sec. 3.4.2	Converter	Constant efficiency	Mission	Low
Sec. 3.4.3	Voltage limit	Safe operating voltage	Mission	Mid
Sec. 3.4.4	Cables	Power loss	Mission	High
Sec. 3.4.5	TMS	Lumped model	Pre-mission and Mission	Low
		Component build-up	Pre-mission and Mission	High
Sec. 3.5	Battery	Constant efficiency	Mission	Low
		Equivalent circuit model	Mission	Mid
		DFM model via PyBaMM	Outside of Aviary	High
Sec. 3.6	Empty weight	Aviary build-up	Pre-mission	Mid
		Custom build-up	Pre-mission	Mid
		Aircraft-level regression	Pre-mission	Low
Sec. 3.7	Aerodynamics	Aviary build-up	Mission	Mid
		Custom $C_{D_0}$ build-up	Mission	Mid
		High lift and stall	Pre-mission	Mid
		BLI	Mission	Mid
Sec. 3.8.1	Power-offtake	Regression	Pre-mission	Mid
Sec. 3.8.2	Geometry	Wing area	Pre-mission	Low
		Tail area	Pre-mission	Low

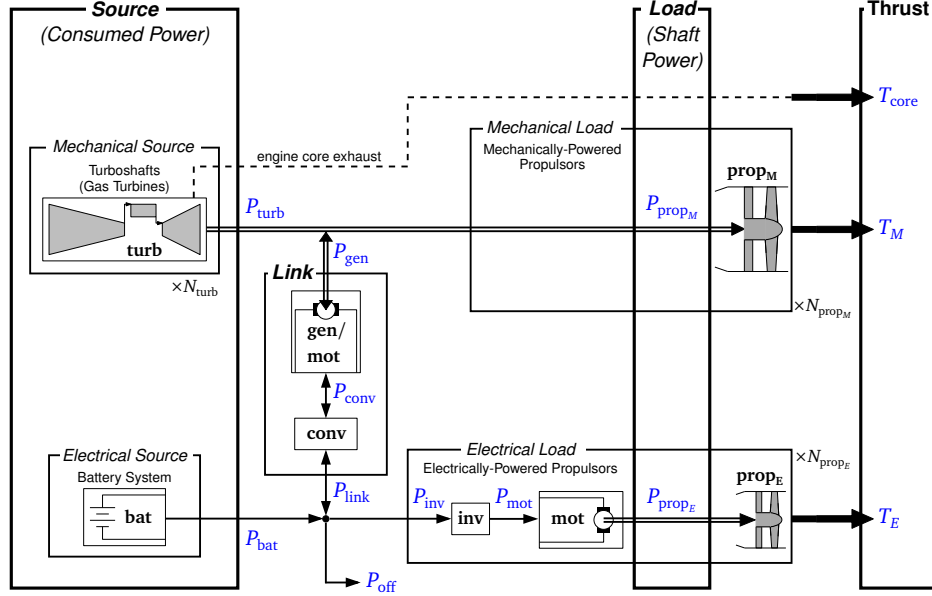
## 3.1 Unified Propulsion System

### 3.1.1 Model Overview

We use the *unified propulsion system model* [9, 23] as a top-level propulsion system model to represent conventional, all-electric, and various electrified propulsion architectures (e.g., full and partial turboelectric, series hybrid, and parallel hybrid) as a continuum. Two nondimensional factors, the *source electrification factor*  $f_S$  and the *load electrification factor*  $f_L$ , fully determine the propulsion system architecture by defining the power split at the source and propulsor (load) levels as

$$f_S \equiv \frac{P_{\text{bat}}}{P_{\text{bat}} + P_{\text{turb}}}, \quad f_L \equiv \frac{P_{\text{prop, E}}}{P_{\text{prop, E}} + P_{\text{prop, M}}}, \quad (1)$$

via the power supplied by the battery (electrical source)  $P_{\text{bat}}$ , the power from any gas turbines  $P_{\text{turb}}$ , the shaft power to (electrical) motor-driven propulsors  $P_{\text{prop, E}}$ , and the shaft power supplied to propulsors by turbine (mechanical) shafts  $P_{\text{prop, M}}$ . In this work, we use the term *propulsor* to refer to any thrust-producing component, whether it be a fan (ducted or un-ducted) or a propeller.



**Figure 4:** Unified propulsion system model, adapted and expanded from Kruger et al. [23] to show this work’s terms and definitions.

Kruger and Uranga [9] originally defined the load electrification factor from the *flow power* split, but we define it for the propulsor *shaft power* in this work because in our framework, the shaft power is always an input to a propulsor model while not all propulsor models compute the flow power. We have also added a thrust contribution from the gas turbine core(s)  $T_{core}$ , as well as a power offtake  $P_{off}$  to account for the aircraft’s non-propulsive power requirements (see Secs. 3.4.5 and 3.8.1). It will be useful to also consider the *thrust electrification factor*

$$f_T \equiv \frac{T_E}{T_E + T_M + T_{core}}, \quad (2)$$

which represents the electrification level in terms of useful thrust work for the aircraft.

Setting  $f_S = f_L = 0$  yields conventional turbofan or turboprop propulsion, the use of  $f_S = f_L = 1$  represents an all-electric propulsion system, and  $0 \leq f_S, f_L \leq 1$  results in a hybrid-electric architecture. These factors can be static design parameters or dynamic control variables that vary across the mission, so we can optimize a dynamic electrification strategy. A detailed explanation of these factors can be found in Kruger and Uranga [9].

Figure 4 shows the overview of the unified propulsion system model. Under this system’s model, we have propulsion component models for turboshafts (or gas turbines) for the mechanical power source, batteries for the electrical source, propulsors to produce thrust, motors to electrically-driven propulsors, generators to convert mechanical power into electrical power, and electrical converters (inverters or rectifiers as needed). These components are connected by the power that flows through, into, and out of them, which is determined by the unified propulsion system based on the overall aircraft thrust required and the two electrification factors.

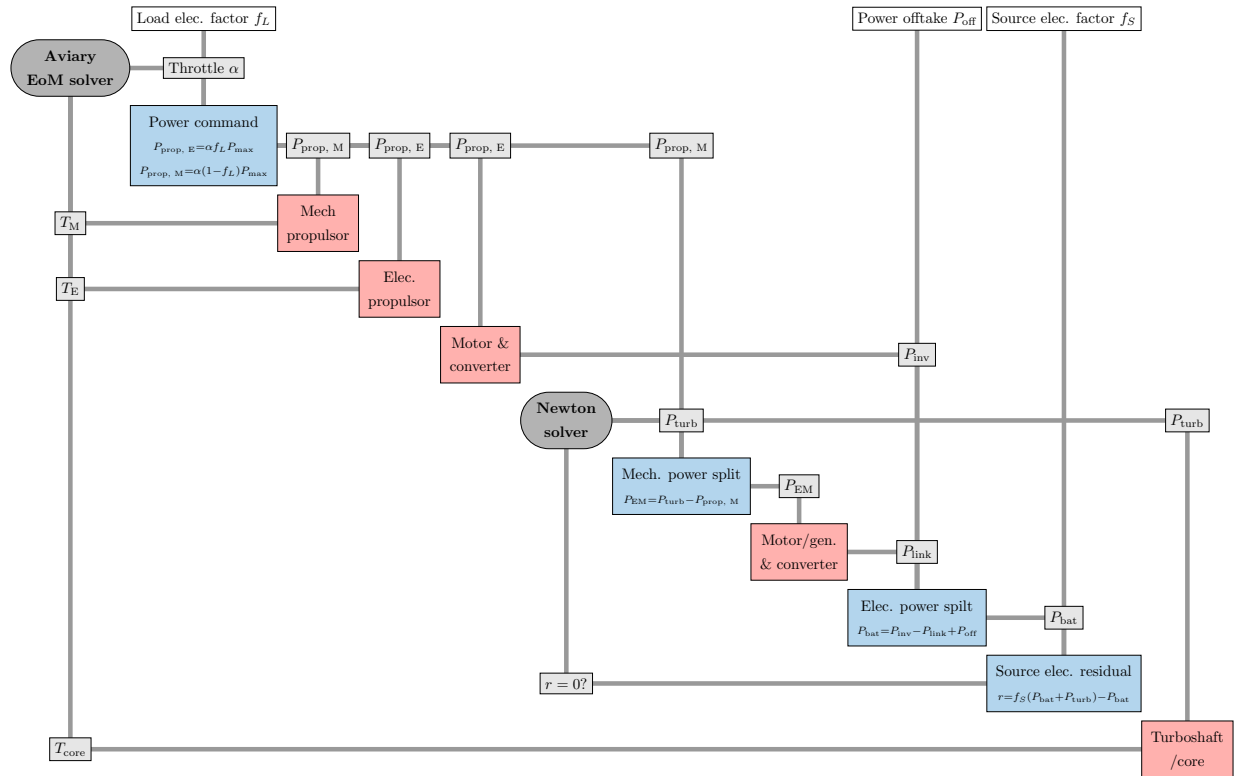
### 3.1.2 Implementation

We implement the unified propulsion system model in OpenMDAO and incorporate it into Aviary using its *EngineBuilder* interface. Our implementation is modular, allowing it to incorporate any fidelity of propulsion component model, as long as each component adheres to the required inputs and outputs listed in Table 2. The optional inputs and outputs are connected between components when available. Table 2 shows the input and output of OpenMDAO components in our implementation, but these do not necessarily follow the physical power flow directions shown in Fig. 4. This is because we define the component inputs and outputs to simplify the implementation and the nonlinear solver structure.

Figure 5 illustrates the propulsion system implementation using the extended design structure matrix (XDSM) [24]. At the top level, there is an Aviary-level Newton solver that varies the throttle  $\alpha$  to match the thrust  $T$  required at each flight condition throughout the mission. This Aviary-level

**Table 2:** Dynamic input and output variables of each component in the unified propulsion system.

Component	Inputs (required)	Outputs (required)	Inputs (optional)	Outputs (optional)
Turboshaft/core	Shaft power	Fuel flow	Inlet flow state	Core thrust, station flow properties, shaft speed
Propulsor	Shaft power	Thrust	-	Exhaust flow properties, shaft speed
Motor	Shaft power	Electric power	Shaft speed	Heat
Generator	Shaft power	Electric power	Shaft speed	Heat
Converter	Power input	Power output	-	Heat
Battery	Power drawn	Energy consumption rate	-	Heat, voltage



**Figure 5:** XDSM diagram showing the component coupling and solver structure for the unified propulsion system. Red blocks generally involve implicit equations and an internal solver, whereas blue blocks represent explicit computations.

EoM solver also involves other subsystems such as aerodynamics, but these are not shown in Fig. 5 for simplicity. The propulsion system as a whole returns thrust, fuel flow, and battery power draw given the throttle and the electrification factors. It may also output other optional variables, including heat and gas turbine exhaust flow properties, that are used in downstream analysis.

Within the unified propulsion system, the “power command” component first computes the propulsor shaft powers for both mechanically and electrically driven propulsors ( $P_{\text{prop, M}}$  and  $P_{\text{prop, E}}$ , respectively). This component refers to the load electrification factor  $f_L$  and an arbitrary scalar  $P_{\text{max}}$ ; the value of  $P_{\text{max}}$  has no effect on the solution because the Aviary throttle is left unbounded in our setup, but an appropriately-scaled value improves solver convergence. The shaft power commands are connected to propulsor components that return thrust to the Aviary’s EoM component. The shaft power for electrically-driven propulsors also connects to the motor and inverter components, which then compute the inverter input power and connect it to the electrical power splitter/merger. The shaft power for mechanically-driven propulsors connects directly to the mechanical power splitter/merger. We then have an inner Newton solver that varies the turboshaft (or engine core) power  $P_{\text{turb}}$  to match the source electrification factor by solving the following residual equation

$$r = f_S(P_{\text{bat}} + P_{\text{turb}}) - P_{\text{bat}} = 0, \quad (3)$$

which is simply a residual form of the original definition in Eq. (2). This inner Newton solver includes the generator and converter components in the loop. Once the solver converges, it provides the shaft power to the turboshaft component, which then computes the fuel flow. The propulsion system also outputs the battery power, given the turboshaft power, inverter power, and non-propulsive power offtake.

Generally, propulsion components have internal implicit equations that require a component-level solver. Also, the input and output of a component model are not always the same as the inputs and outputs defined by the propulsion system. For example, the system defines the shaft power as an input of a propulsor component, but an actual propulsor model may have the power as an output, while the shaft speed is an input. In such a case, we add a Newton solver at the component-level solver to vary the shaft speed to match the power.

As a result, the propulsion system model has multiple layers of Newton solvers (at least two layers as shown in Fig. 5, and usually more when a component-level solver is required). We solve this system using OpenMDAO’s hierarchical Newton solver [14, Sec. 13.2], which is more robust than a monolithic Newton solver. In the future, our implementation may benefit from a hybrid approach that combines the hierarchical and monolithic Newton solvers, potentially improving computational efficiency without sacrificing robustness.

### 3.2 Turboshaft: Thermal Engine Core

The unified propulsion system defines “turboshaft” as a component that consumes fuel to generate shaft power. This component can be a turbofan engine core, or more generally a gas turbine, rather than a stand-alone turboshaft engine. To represent a turbofan, we model the turbofan’s core and fan separately to generalize the propulsion system architecture. This section summarizes the models that relate fuel flow and shaft power, as well as the weight estimation of the propulsion system’s turboshaft component.

### 3.2.1 Constant-PSFC Model

The lowest fidelity turboshaft model uses a constant, user-defined power-specific fuel consumption (PSFC) to compute the fuel flow rate as

$$\dot{m}_{\text{fuel}} = \text{PSFC} \cdot P_{\text{turb}} , \quad (4)$$

which the mission analysis integrates to compute the mission fuel burn.

Given  $P_{\text{turb}}$  across missions and at every flight condition, we impose the following *power margin* constraint

$$P_{\text{margin, turb}} \equiv P_{\text{rate, turb}} - P_{\text{turb}} \geq 0 \quad \text{for all flight conditions} \quad (5)$$

that sizes the turboshaft power rating  $P_{\text{rate, turb}}$ . This model ignores altitude power lapse.

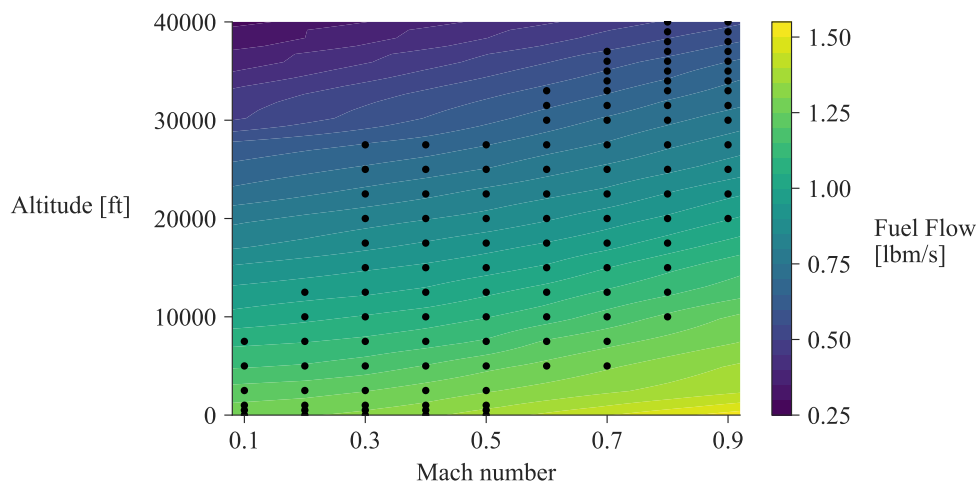
### 3.2.2 Engine Deck

Our framework also supports a standard engine-deck approach. This interpolates user-provided engine performance data to compute the shaft power and fuel flow rate as a function of altitude, freestream Mach number, and throttle. We then use a nonlinear solver to vary the throttle to match the deck's shaft power output to the shaft power requested by the propulsion system. Figure 6 shows an example of an engine deck interpolation.

The engine data table should include the tabulated data of shaft power and fuel flow rate at various altitudes, Mach numbers, and throttle settings. Users can generate the data table from any engine modeling tool or experimental data. Our framework provides several engine data tables that we generated using pyCycle [25]. Our engine deck also supports interpolation of additional outputs, such as combustor inlet temperature and pressure that can be used in downstream analysis.

For sizing purposes, the optimizer varies the turboshaft scaling factor  $s_{\text{turb}}$  that linearly scales the engine deck outputs. This is essentially a rubber engine approach based on the underlying deck data. We then impose the power margin constraint to size the scaling factor:

$$P_{\text{margin, turb}} \equiv s_{\text{turb}} \bar{P}_{\text{turb}} - P_{\text{turb}} \geq 0 \quad \text{for all flight conditions.} \quad (6)$$



**Figure 6:** Example of a turboshaft engine deck: slice of fuel flow rate interpolation at maximum throttle. Black points are the engine table data points.

where  $P_{\text{turb}}$  is the power required by the unified propulsion system, and  $\bar{P}_{\text{turb}}$  is the maximum power available that varies as a function of the flight conditions and is obtained by evaluating the engine deck at the maximum throttle. The scaling factor  $s_{\text{turb}}$  should be close to unity because linear scaling is only a valid approximation near the original maximum power of the engine deck. If the aircraft power requires a much larger or smaller scaling factor, it should use a different engine deck with a closer power rating.

### 3.2.3 Thermodynamic Cycle Model (pyCycle)

The highest-fidelity option for the turboshaft component is a thermodynamic cycle model using pyCycle [25]. The thermodynamic cycle modeling library pyCycle is heavily inspired by Numerical Propulsion System Simulation (NPSS) [26], but it is implemented using OpenMDAO and provides analytical derivatives. This enables seamless integration with the rest of our OpenMDAO-based models. The details of our engine modeling approach in pyCycle will be presented in an upcoming publication led by our collaborators at MIT’s Gas Turbine Lab.

To incorporate the pyCycle model into mission analysis, our framework provides two coupling strategies [27] as illustrated in Fig. 7:

1. A **direct coupling approach** uses pyCycle’s on-design analysis within Aviary’s pre-mission group, and Aviary’s mission analysis directly calls pyCycle in off-design mode at all mission discretization points. The off-design analyses can be parallelized.
2. A **dynamically-trained surrogate approach** calls pyCycle in on-design mode and a series of off-design analyses within Aviary’s pre-mission group to generate an engine deck data set. We dynamically re-train the engine deck data every time the optimizer changes the engine design variables. The deck’s data set generation can be parallelized. When the mission analysis runs, the engine deck functions as a surrogate model and is used in place of directly calling pyCycle’s off-design analyses. OpenMDAO refers to this approach as *dynamic training*, and it propagates the derivatives through the dynamic surrogate process.

In a theoretical limit with zero mission discretization error and zero surrogate prediction error, the two approaches yield identical results. In practice, there is a trade between computational cost and accuracy, with the cost of each approach scaling differently. Because a series of off-design analyses dominates the computational cost, the cost of direct coupling is roughly proportional to the number of missions and the level of mission discretization. On the other hand, the cost of using a dynamically-trained surrogate scales with the number of deck data points, which is determined by the nonlinearity of the operational envelope and the desired accuracy of the surrogate. The cost of surrogate coupling does not scale with the number of missions, so we recommend it for multi-mission optimization.

We wrap pyCycle on-design and off-design analyses in OpenMDAO subproblems, *i.e.*, an Aviary-level *ExplicitComponent* wraps a lower-level OpenMDAO problem that hosts pyCycle. This accelerates the total derivative computation for the direct coupling approach by leveraging a nested-hierarchical linear solver strategy [28]. Furthermore, the subproblem significantly simplifies the Aviary-level problem structure by hiding thousands of pyCycle’s intermediate state variables from the aircraft-level layer to only expose the relevant inputs and outputs.

For sizing, we impose the power margin constraint Eq. (6) but without the scaling factor (*i.e.*,  $s_{\text{turb}} = 1$ ) because the optimizer directly varies the engine design and sizing variables. The maximum

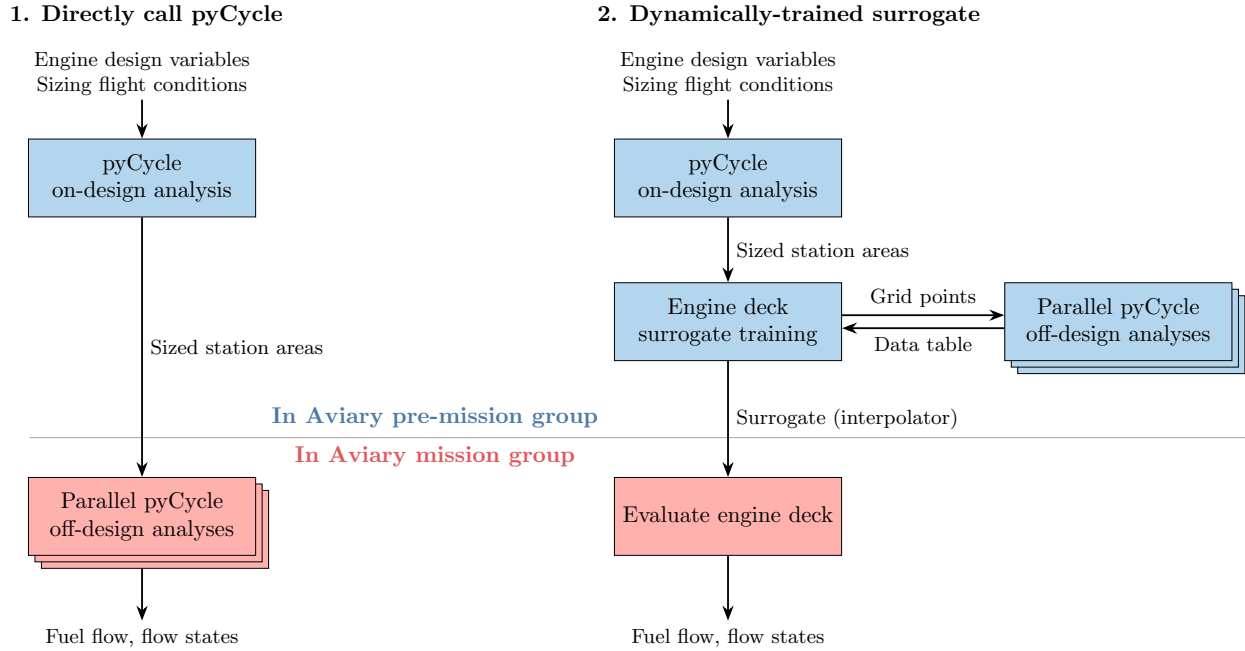


Figure 7: Two approaches to calling pyCycle analysis from Aviary.

turboshaft power available at each flight condition is primarily determined by the turbine inlet temperature limit.

### 3.2.4 Turboshaft/Core Weight Estimation

One option to estimate the turboshaft weight is Raymer’s statistical model [29, Sec. 10.4], which computes the turboshaft mass in kilograms as

$$m_{\text{turb}} = 0.96 P_{\text{rate, turb}}^{0.803}, \quad (7)$$

from the turboshaft sea-level static power rating  $P_{\text{rate, turb}}$  specified in kilowatts.

Equation (7) is valid for engines rated up to 3,728 kW, corresponding to an engine weight of 708 kg [29, Sec. 10.4]. We supplement it with another weight model for higher-power engines in order to support much larger engines for the study of single- and twin-aisle transport aircraft. For this, we developed the weight regression analysis detailed in Appendix A that shows that a specific power of 4.54 kW/kg is appropriate for weight estimations. The framework allows the user to specify a different specific power.

## 3.3 Propulsors

The unified propulsion system defines a *propulsor* as a component that converts shaft power into thrust. This section explains our models for power-to-thrust conversion performance and their weight estimation.

### 3.3.1 Constant Efficiency Model

The lowest fidelity model converts the shaft power to thrust via an overall efficiency  $\eta_{\text{overall}}$  as

$$T = \frac{\eta_{\text{overall}} P_{\text{shaft}}}{V_{\infty}}, \quad (8)$$

where  $T$  is the thrust,  $P_{\text{shaft}}$  the total shaft power input to the propulsor(s), and  $V_{\infty}$  the flight speed. This model is only valid when  $V_{\infty}$  is sufficiently high and cannot be used in, for example, a sea-level static condition where  $V_{\infty} = 0$ . We add a power margin constraint similar to Eq. (5) to size the propulsor power rating.

### 3.3.2 Thrust Coefficient Propeller Model

For propellers, we use a mid-fidelity model with a non-constant efficiency following Kruger and Uranga [9]. The efficiency becomes a function of the thrust coefficient to account for losses due to viscous profile drag and propwash, and thus varies with flight speed and thrust level.

### 3.3.3 Ducted Fan Deck

The ducted fan deck interpolates user-provided performance data to compute the net thrust and fan shaft power as a function of altitude, freestream Mach number, and throttle. A nonlinear solver then varies the throttle to match the deck's shaft power to the power provided by the propulsion system, in a manner similar to how a turboshaft engine deck is used. We also implement linear scaling to rubberize the ducted fan deck, and impose a power margin constraint similar to Eq. (6).

The fan performance data can be generated by the user from any model or experimental data. Our framework also provides ducted fan data tables generated by pyCycle.

### 3.3.4 pyCycle Ducted Fan Model

To provide a higher fidelity fan model, our framework supports directly calling pyCycle's ducted fan model. This then internally solves the implicit equations to compute fan thrust for a given shaft power.

When the fan and core components operate together as a turbofan engine, we connect the flow properties behind the fan to the engine core inlet. A Newton solver then iterates the fan and core components to ensure consistency in the mass flow rates, the bypass ratio, and the shaft speed (with a given gear ratio).

In the same way as for the turboshaft pyCycle model, we provide two coupling options (direct or dynamically-trained surrogate) to integrate pyCycle into mission analysis. We also use the same subproblem approach to accelerate derivative computations.

### 3.3.5 Propulsor Weight Estimation

A propeller's weight is estimated using the textbook model [30, Sec. 8.4]

$$m_{\text{propeller}} = 0.108 \left( d_{\text{prop}} P_{\text{rate, prop}} \sqrt{N_{\text{blade}}} \right)^{0.782}, \quad (9)$$

where  $d_{\text{prop}}$  is the propeller diameter,  $P_{\text{rate, prop}}$  is the propeller power rating at sea level, and  $N_{\text{blade}}$  is the number of blades.

For ducted fans, we estimate the weight based on a user-provided specific power, for which we use a default value of 14.43 kW/kg based on our regression analysis of Appendix A.

### 3.4 Electrical Machines and Related Components

#### 3.4.1 Motors and Generators

As a low-fidelity model, we assume a constant motor efficiency  $\eta_{\text{mot}}$  regardless of the power level, and thus relate the motor's electrical input power  $P_{\text{mot}}$  to the shaft power  $P_{\text{prop, E}}$  and heat  $\dot{Q}_{\text{mot}}$  it produces via

$$P_{\text{prop, E}} = \eta_{\text{mot}} P_{\text{mot}}, \quad \dot{Q}_{\text{mot}} = (1 - \eta_{\text{mot}}) P_{\text{mot}}. \quad (10)$$

Similarly, for a generator receiving electrical power from a converter,  $P_{\text{gen}} = \eta_{\text{conv}} P_{\text{conv}}$  and  $\dot{Q}_{\text{gen}} = (1 - \eta_{\text{gen}}) P_{\text{conv}}$ .

In this model, we estimate the motor and generator weight based on the specific power and the motor power rating (*i.e.*, maximum continuous power). The specific power value is a user input and represents the technology level of the machine. The power rating is sized by a power margin constraint similar to Eq. (5), imposing that the power required throughout the mission and at all sizing points be less than or equal to the power rating of the electrical machine.

#### 3.4.2 Converters

We use a low-fidelity constant efficiency model for inverters and rectifiers, similar to the model used for electrical machines, such that

$$P_{\text{out}} = \eta_{\text{conv}} P_{\text{in}}, \quad \dot{Q} = (1 - \eta_{\text{conv}}) P_{\text{in}}. \quad (11)$$

where  $P_{\text{in}}$  is the power input and  $P_{\text{out}}$  the power output, respectively, to the power electronics component. We estimate converter weight based on its power rating and a user-provided specific power.

#### 3.4.3 System Voltage Constraint

The system voltage is constrained to not exceed the safe operating voltage (SOV) to avoid the risk of arcing and discharge through the air. We calculate the safe operating voltage using the model by Byahut [31, Sec. 4.1], which is based on the breakdown voltage by Paschen's Law [32] and a basic wire insulation model.

#### 3.4.4 Cables

Electrified aircraft require high-voltage cables to transmit power from source to load. As power increases, the mass of cables increases and the power dissipated in the cables increases. We use models based on Byahut and Uranga [33] to calculate cable lengths and the resulting weights and power dissipation.

Assuming that there are  $N_{\text{prop}}$  electrically-driven propulsors equally spaced along the wing's span length  $b$ , the length of cables throughout the aircraft is calculated as

$$L_{\text{cable}} = n_{\text{wires}} \left( N_{\text{prop}} C_{\text{fuse}} + 2 \sum_{k=1}^{\frac{1}{2} N_{\text{prop}}} \frac{kb}{N_{\text{prop}}} \right), \quad (12)$$

where the factor  $n_{\text{wires}}$  is set to 4 for a DC system and the fuselage circumference  $C_{\text{fuse}}$  is calculated assuming an elliptical cross-section. If the propulsors are on the fuselage rather than the wing, the fuselage length is used in place of the second term in the parenthesis of the above equation.

This length is then used to calculate the mass of the cable given some insulator and conductor (taken to be copper and  $t_i = 1$  mm thick polyethylene, respectively) as

$$m_{\text{cable}} = 1.2 \times \frac{1}{4} \pi \left( \rho_{\text{cond}} d_{\text{cond}}^2 + \rho_{\text{ins}} \left[ (2t_i + d_{\text{cond}})^2 - d_{\text{cond}}^2 \right] \right) L_{\text{cable}} . \quad (13)$$

via the conductor's and insulator's densities,  $\rho_{\text{cond}}$  and  $\rho_{\text{ins}}$ , and diameters,  $d_{\text{cond}}$  and  $d_{\text{ins}}$ . The factor 1.2 accounts for cable attachments and other fixtures.

With the electrical propulsion system's current  $I$  calculated as the ratio of maximum required to safe operating (SOV), the American Wire Gauge standard can be used to determine the conductor wire diameter. As the current changes from iteration to iteration, we use the data-fitting relation

$$d_{\text{cond}} = 6.0172 \times 10^{-5} I + 5.6664 \times 10^{-4} . \quad (14)$$

Finally, the cable length  $L_{\text{cable}}$  and area  $A_{\text{cable}}$  set the resistance  $R_{\text{cable}}$  of the cable, which is used to calculate the power dissipated in the cables  $P_{\text{cable}}$ , namely

$$R_{\text{cable}} = \frac{L_{\text{cable}}}{\sigma A_{\text{cable}}} , \quad P_{\text{cable}} = I^2 R_{\text{cable}} . \quad (15)$$

The unified propulsion system accounts for this power loss as part of the other power offtakes (see Sec. 3.8.1).

### 3.4.5 Thermal Management System (TMS)

**Lumped Model.** A low-fidelity TMS model sizes its weight based on the maximum instantaneous heat load across all mission flight conditions and other sizing conditions. The overall heat load  $\dot{Q}$  is the summation of all the component's heat generation, which is computed by the motor, generator, converters, and battery models. The optimizer determines the TMS's overall heat load capacity,  $\dot{Q}_{\text{capacity}}$ , to satisfy the following constraint

$$\dot{Q} = \sum_{\text{comp}} \dot{Q}_{\text{comp}} \leq \dot{Q}_{\text{capacity}} \quad \text{for all flight conditions.} \quad (16)$$

The TMS weight is then calculated based on a user-provided constant specific heat load capacity (in kW/kg), which is analogous to the use of a constant specific power for electrical components. For example, Kruger and Uranga [9] suggested the value range of 10–25 kW/kg for technology that is expected to be mature by 2035. The power and duct drag penalties of TMS are not considered in this model.

**Component Temperature Simulation Model.** We also implement a detailed, component-based TMS model originally developed in OpenConcept [7, 34]. This section provides a brief summary of this TMS model as used in our framework, and a more detailed description can be found in Adler et al. [34].

This model assumes separate motor-cooling and battery-cooling loops. The motor cooling loop consists of a ducted heat exchanger, motors as a heat source, a coolant hose, and a coolant pump. The

battery cooling loop is similar, but adds a variable-nozzle ducted heat exchanger and a refrigerator for active cooling. The refrigerator can be bypassed when the heat load is low, saving power.

The TMS models simulate the motor and battery temperatures across the mission. The original implementation in OpenConcept used its own time integrator to solve the temperature ODE, while in this work, we solve the ODE using the Radau collocation method in Dymos. Path inequality constraints are to maintain component temperatures within specified limits. These temperature constraints size the ducted heat exchangers, coolant pumps, coolant mass flow, coolant pump power rating, and battery refrigerator power rating. We then estimate the TMS weight by component build-up. The TMS model draws power required in coolant pumps and refrigerators from the propulsion system, and the cooling duct drag (or thrust) is added to the aircraft-level drag build-up of the aerodynamics subsystem.

## 3.5 Batteries

### 3.5.1 Constant Efficiency Model

The lowest fidelity battery model assumes a constant battery efficiency value  $\eta_{\text{bat}}$  to compute the rate of change of the battery's stored energy,  $\dot{E}_{\text{bat}}$ , and the heat rate,  $\dot{Q}_{\text{bat}}$  it generates as

$$\dot{E}_{\text{bat}} = \frac{P_{\text{bat}}}{\eta_{\text{bat}}}, \quad \dot{Q}_{\text{bat}} = (1 - \eta_{\text{bat}})P_{\text{bat}}, \quad (17)$$

We integrate  $\dot{E}_{\text{bat}}$  across the mission to compute the mission-level electrical energy consumption.

Once the total energy and power profile are computed by mission analysis, we size the battery mass  $m_{\text{bat}}$  based on a user-provided pack-level battery specific energy (BSE), pack-level battery specific power (PSE), minimum state of charge (SOC), and maximum discharge C-rate. The optimizer determines the battery weight that satisfies both energy-sizing and power-sizing constraints:

$$\text{SOC}(t) = \frac{E_{\text{bat}}(t)}{\text{BSE} \cdot m_{\text{bat}}} \geq \text{Min. SOC} \quad (\text{energy sizing}) \quad (18)$$

$$\text{Discharge C-rate}(t) = \frac{P_{\text{bat}}(t)}{\text{PSE} \cdot m_{\text{bat}}} \leq \text{Max. discharge C-rate} \quad (\text{power sizing}) \quad (19)$$

Equation (19) assumes a constant nominal voltage. Equations (18) and (19) are path constraints and imposed throughout the flight mission and at any other non-mission sizing conditions. This is because the power profile is a function of time, and the SOC is not necessarily minimum at the end of the mission when considering in-flight charging. For charging, we also impose a charging C-rate limit similar to Eq. (19).

### 3.5.2 Equivalent Circuit Model

Our mid-fidelity model uses an equivalent circuit model (ECM), which is an electric circuit composed of a cell, resistors, and capacitors. We use the Thevenin model with two parallel resistor-capacitor (RC) pairs connected in series.

Given the power profile from the propulsion system, the ECM module solves a set of ODEs to simulate the battery's dynamic response, which are solved using the Radau collocation method [35] provided by Dymos and Aviary. The ECM module outputs the time histories of SOC, voltage, C-rate, and heat produced. The optimizer then sizes the number of cells in both parallel and series to satisfy

path constraints on SOC and C-rate limits, similar to Eqs. (18) and (19). We also impose minimum and maximum voltage constraints to ensure that the cells operate within their safe operational limits.

The battery pack mass is given by

$$m_{\text{bat}} = \frac{m_{\text{cell}} N_p N_s}{B.F.}, \quad (20)$$

where  $m_{\text{cell}}$  is the mass on an individual cell,  $N_p$  and  $N_s$  are the number of cells in parallel and series, respectively, and  $B.F.$  stands for burden factor, a user-provided factor to account for packaging efficiency. We apply continuous approximation to  $N_p$  and  $N_s$  to avoid integer variables.

### 3.5.3 Doyle-Fueller-Newman Model via PyBaMM

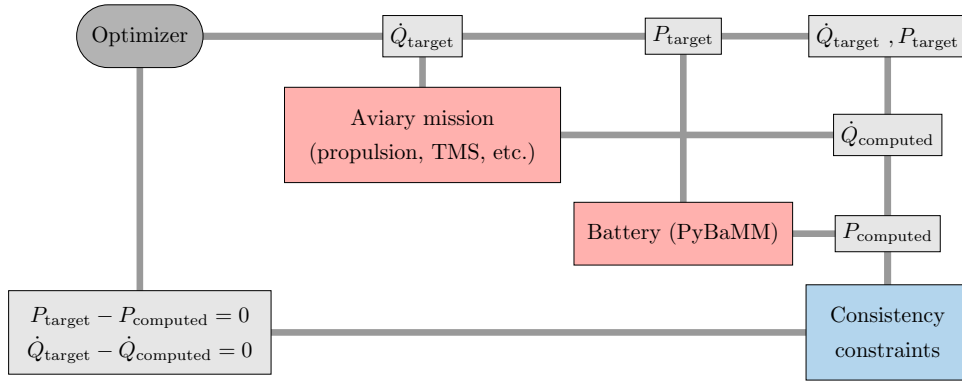
The high-fidelity battery model available in our framework is the Doyle-Fueller-Newman (DFN) model [36]. We use the implementation in Python Battery Mathematical Modelling (PyBaMM) [37], an open-source tool for battery cell simulations. We wrap PyBaMM in an OpenMDAO component so it works with the rest of the framework. The derivatives of PyBaMM’s outputs with respect to inputs are computed using the forward model sensitivities provided by the SUNDIALS IDAS solver [38, 39]. The details of the battery modeling in PyBaMM will be presented in upcoming publications by our collaborators.

The inputs and outputs to the battery module are the same between DFN and ECM, but DFN replaces ECM’s ODEs with its higher-fidelity simulation. The battery sizing variables and constraints, as well as the pack weight model, are also the same.

PyBaMM uses its own differential algebraic equations (DAE) solver, unlike the ECM that integrates the ODEs using Aviary and Dymos. Because of this, we place the PyBaMM module outside of the Aviary’s mission analysis and couple the aircraft-level dynamics and DFN battery dynamics using *individual discipline feasible* (IDF) techniques [14, Sec. 13.4] via the implementation illustrated in Fig. 8. This is achieved by introducing *target variables* for the power profile  $P_{\text{target}}$  and battery heat profile  $\dot{Q}_{\text{target}}$ , which are vector variables that consist of power and heat values at each mission discretization node. The optimizer directly controls these target variables as optimization variables. We then provide  $P_{\text{target}}$  to the PyBaMM module and  $\dot{Q}_{\text{target}}$  to the Aviary’s mission group, which is connected to the TMS model internally. Finally, the following consistency constraints are imposed

$$\begin{aligned} h_{\text{power}}^c &= P_{\text{target}} - P_{\text{computed}} = 0, \\ h_{\text{heat}}^c &= \dot{Q}_{\text{target}} - \dot{Q}_{\text{computed}} = 0, \end{aligned} \quad (21)$$

where  $P_{\text{computed}}$  is the power profile computed by the propulsion system, and  $\dot{Q}_{\text{computed}}$  is the battery heat production profile computed by the PyBaMM module. Once the optimizer finds a solution that satisfies these constraints, the power and heat are consistent across propulsion, TMS, and battery components. A converged solution is equivalent to the one that would be obtained by directly coupling these subsystems using the *multidisciplinary feasible* approach [14, Sec. 13.4].



**Figure 8:** IDF coupling between Aviary mission analysis and PyBaMM’s battery DAE solver using the target variables and consistency constraints.

## 3.6 Empty Weight and Takeoff Weight Sizing

### 3.6.1 Weight breakdown

We break down the operational empty weight (OEW) as follows:

$$W_{\text{OEW}} = \underbrace{W_{\text{airframe}} + W_{\text{systems}} + W_{\text{operational}}}_{\text{Component build-up or regression}} + \underbrace{W_{\text{propulsion}} + W_{\text{battery}}}_{\text{Component build-up}} \quad (22)$$

For the airframe, systems, and operational weights, we implement component build-up methods and a regression model as detailed in the following subsection. The propulsion system and battery weights are the summation of the component weights discussed in the corresponding sections.

### 3.6.2 Aviary Weight Buildup

Our framework can directly use Aviary’s FLOPS weight build-up. For this, the conventional engine weight from FLOPS is excluded and replaced by our own propulsion system weight.

### 3.6.3 Textbook weight buildups

We also implement custom weight build-up models. One option is to use models imported from OpenConcept [7, 40]. This implements the component weight models of the wing, horizontal tail, vertical tail, fuselage, main landing gear, and nose landing gear from Raymer [29, Sec. 15.3]. The Raymer models only work for a tube-and-wing configuration, so we also use OpenConcept’s blended-wing body (BWB) weight model. The BWB model combines the cabin and aft-body weight models from Bradley [41] with the Raymer’s tube-and-wing models for the wing and landing gear.

The other method we implemented is the weight buildup from the appendices in Torenbeek [30, Sec. 8, App. C, App. D]. This consists of wing, fuselage, vertical tail, and horizontal tail models. We use the same landing gear weight model as the previous method.

### 3.6.4 Regression Model for Airframe, Systems, and Operational Items Weight

Another option is a simple, system-level regression model. The approach estimates the aircraft, systems, and operational weights as a function of MTOW, payload weight, propulsion system weight,

range  $R$ , and maximum sea-level static thrust  $T_{SLS}$ . The weight regression model is adopted from the Future Aircraft Sizing Tool [42], an open-source aircraft sizing tool. The regression methodology is detailed in Arnson et al. [43]. With this regression model, the OEW is calculated from the regression function  $f_{\text{regression}}$  as

$$W_{\text{OEW}} = f_{\text{regression}}(W_{\text{MTO}}, W_{\text{burden}}, R, T_{\text{SLS}}) + W_{\text{propulsion}} + W_{\text{battery}}, \quad (23)$$

where  $W_{\text{burden}}$  is defined as the sum of payload and propulsion system weights, which consists of the electrified powerplant and conventional mechanical powerplant weights.

### 3.6.5 Weight Consistency Constraints

The OEW models and MTOW are linked by an implicit relation that needs to be solved iteratively. In Aviary (hence in our framework), this is done by adding the MTOW and actual takeoff weight  $W_{\text{TO}}$  for each mission as design variables and imposing the weight consistency constraints

$$W_{\text{TO}}^i = (W_{\text{OEW}} + W_{\text{fuel}}^i + W_{\text{payload}}^i) \quad \forall i, \quad (24)$$

$$W_{\text{TO}}^i \leq W_{\text{MTO}} \quad \forall i, \quad (25)$$

where  $i$  is a mission index for multi-mission optimization.

## 3.7 Aerodynamics

We implemented or brought into Aviary multiple drag models. The appropriate model selection depends on the aircraft configurations and flight phases.

### 3.7.1 Aviary Build-up

The first drag model is Aviary's FLOPS-based drag build-up model [11, 19]. This uses the Delta Method, an empirical drag build-up [44]. The implementation in Aviary is modular, so the lift-dependent component is separated out for use with either of the zero-lift drag models in Sec. 3.7.2.

### 3.7.2 Zero-Lift Drag Build-up

We have two options for zero-lift drag in addition to the FLOPS buildup. The first is the drag buildup from Raymer [29, Sec. 12.5]. This calculates individual form factors and then drag coefficients for components, including the landing gear and flap drag in takeoff configuration.

The other option is the empirical drag model in OpenConcept [7, 40]. This uses form factors from Torenbeek [30], flap drag from Roskam [45, Part VI, Sec. 4.7], and the skin friction coefficient used in OpenVSP's aerodynamics tool [46]. This option can also compute the drag for a takeoff configuration.

### 3.7.3 High-Lift Model

We use the maximum lift coefficient components from OpenConcept [7, 40], which calculates the maximum lift coefficient without flap from Raymer [29] and the maximum lift coefficient with flap down by fitting the data in Roskam [45, Part VI]. The stall speed is adapted from Aviary's GASP implementation [20].

### 3.7.4 Boundary Layer Ingestion (BLI)

We implement the analytical BLI model developed by Uranga et al. [47], which captures both drag and power reduction benefits of BLI. It was originally developed using the power balance method [48], but we repurposed it for a conventional force-based method to incorporate it into Aviary’s mission analysis. We compute the “net” thrust  $T$  for a given flow power  $P_k$  using the following equations:

$$T = \dot{m}_{\text{prop}}(V_{\text{jet}} - V_{\infty}) + f_{\text{BLI}}D'_p + \frac{1}{2}\rho_{\infty}V_{\infty}^2S_{\text{ref}}\Delta C_{\Phi,\text{surf}} , \quad (26)$$

$$\dot{m} = \rho_{\infty}V_{\text{jet}} \frac{A_{\text{jet}}}{A_{\text{nozzle}}} \frac{A_{\text{nozzle}}}{A_{\text{prop, in}}} A_{\text{prop, in}} , \quad (27)$$

$$V_{\text{jet}} = \sqrt{\frac{2 \left( P_k - (1 - f_{\text{wake}}) f_{\text{BLI}} D'_p V_{\infty} \right)}{\dot{m}}} + V_{\infty}^2 , \quad (28)$$

where the net thrust  $T$  is defined to be the sum of the conventional thrust (first term) and the BLI drag reduction (second and third terms),  $\dot{m}_{\text{prop}}$  is the mass flow rate through the propulsor,  $V_{\text{jet}}$  is the propulsor’s jet velocity,  $D'_p$  is the parasite drag in the absence of BLI,  $\rho_{\infty}$  and  $V_{\infty}$  are the freestream density and velocity, respectively, and  $A_{(\cdot)}$  designates the areas for the jet, nozzle, and propulsor inlet. The model parameters consist of  $f_{\text{BLI}}$ ,  $f_{\text{wake}}$ ,  $\Delta C_{\Phi,\text{surf}}$ , and  $A_{\text{jet}}/A_{\text{nozzle}}$ , for which we follow the definitions by Uranga et al. [47]. These parameter values are dependent on the aircraft and BLI configuration. The parameter values for an aft-fuselage BLI configuration are adopted from Uranga et al. [47].

In our implementation, the BLI component is included as a part of the propulsor blocks in the unified propulsion system model. Equations (26)–(28) are implicit with respect to  $\dot{m}_{\text{prop}}$  and  $V_{\text{jet}}$ , so a Newton solver is used to iteratively solve for these variables. This BLI-level solver is nested under the Aviary-level and propulsion system-level solvers discussed in Sec. 3.1.2.

## 3.8 Other Models

### 3.8.1 Non-Propulsive Power Offtake

Power offtakes are power loads that an aircraft’s powerplant must supply in addition to the propulsive power. These offtakes represent amount to power draws from such systems as ice protection, landing gear, control surface actuation, environmental control, avionics, fuel systems, passenger service, or cabin lighting. In the conceptual design stage, exact offtakes are infeasible to calculate as it is too early in the design process to make assumptions about secondary subsystems. Therefore, we use data-driven regressions to estimate power offtakes to ensure that the aircraft’s powerplant and energy source(s) are not undersized.

We implement the installed and continuous electric power draw models from Arnson et al. [49], namely

$$P_{\text{offtake, inst}} = 0.78391 W_{\text{MTO}}^{0.5429} , \quad (29)$$

$$P_{\text{offtake, cont}} = 0.58794 W_{\text{MTO}}^{0.5429} , \quad (30)$$

where  $W_{\text{MTO}}$  is the aircraft’s MTOW in kilograms, and the powers are in kilowatts. *Installed* power assumes that all subsystems are actively drawing their full power load, which is the sizing condition. *Continuous* power represents the subsystem power draw during nominal operation, where some subsystems, such as ice protection or landing gear actuation, are inactive [49].

### 3.8.2 Geometry

**Wing Area.** We provide two options for wing area sizing. First, a user can declare it as a design variable, and the optimizer sizes it subject to various constraints, including cruise lift coefficient, takeoff balanced field length, and wing loading. Second, it is also possible to specify the wing loading value and compute the wing area as a function of MTOW.

**Tail Areas.** To size the horizontal and vertical tails, the components in OpenConcept [40, 50] are used, which implement the tail volume calculations in Raymer [29, Sec. 6.5]. Given the horizontal tail volume coefficient  $C_{HT}$ , the horizontal tail area is computed as

$$S_{\text{ref,HT}} = \frac{C_{HT}}{L_{HT}} \text{MAC}_{\text{wing}} S_{\text{ref,wing}}, \quad (31)$$

where  $S_{\text{ref,wing}}$  is the wing area,  $L_{HT}$  the tail moment arm, and  $\text{MAC}_{\text{wing}}$  the wing's mean aerodynamic chord (MAC), calculated from another OpenConcept component. Similarly, the vertical tail area is computed based on its volume coefficient  $C_{VT}$ , moment arm  $L_{VT}$ , and the wing span  $b_{\text{wing}}$  as

$$S_{\text{ref,VT}} = \frac{C_{VT}}{L_{VT}} b_{\text{wing}} S_{\text{ref,wing}}. \quad (32)$$

The tail moment arms  $L_{HT}$  and  $L_{VT}$  are approximated as 52.5% of the fuselage length for wing-mounted engines and 47.5% of the fuselage length for fuselage-mounted engines following Raymer [29, Sec. 6.5].

## 4 Demonstrations of Electrified Aircraft Design Optimization

### 4.1 Hybrid-Electric Single-Aisle Aircraft: NASA SUSAN Concept

#### 4.1.1 Model Setup

SUSAN (subsonic single aft-engine electrofan) is a hybrid-electric<sup>1</sup> aircraft concept proposed by NASA [4, 51] that has one aft-fuselage BLI engine that generates thrust and drives a generator, along with distributed electric propulsors under the wing. It also carries a rechargeable battery that assists nominal flight power and a single-use battery for emergency all-electric flight in case of engine failure. Figure 9 shows the overview of SUSAN's propulsion system.

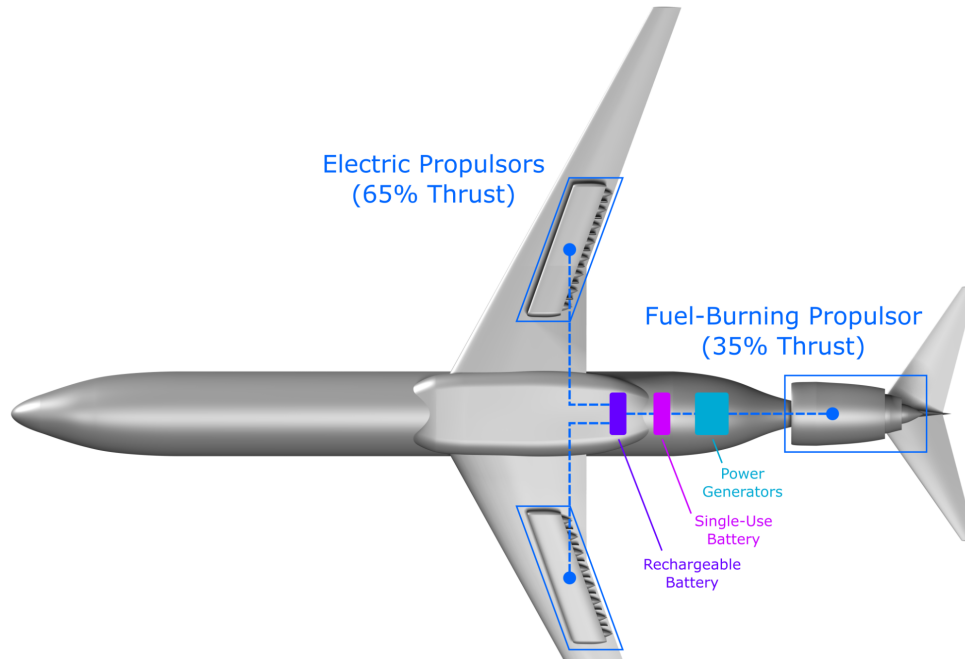
We optimize the SUSAN configuration for a design range of 2,500 nmi and a design payload of 39,600 lb [51]. For consistency, our mission setup and technology-level assumptions follow the NASA literature [51, 52]. We only model the climb-cruise-descent phases, and add a constant fuel weight to account for taxi, takeoff, landing, and reserve following Chau and Duensing [51].

The SUSAN model in this work assembles the following subsystem models that are discussed in the previous sections:

- Scaled engine core deck with specific-power-based weight estimation
- Constant-efficiency motors, generators, and converters with specific-power-based weight estimation
- Constant-efficiency battery model with pack-level power and energy sizing
- Component-buildup TMS model

---

<sup>1</sup>A hybrid-electric propulsion architecture is a particular instance of electrified propulsion in which batteries are used



**Figure 9:** Overview of SUSAN's electrified propulsion system [51]

- Power offtake regression model and TMS power draw
- Airframe-systems-operational items weight surrogate model
- BLI propulsor model with specific-power-based weight estimation
- OpenConcept  $C_{D_0}$  build-up and Aviary's FLOPS induced drag model
- User-provided polar model for low-speed aerodynamics
- Wing area sizing by a constant wing loading of  $129.8 \text{ lb/ft}^2$  [51]
- Tail area sizing by the volume coefficient methods

We calibrated the parasite drag, fuel flow, as well as airframe, systems, and operational weight to closely match the reference values [51]. This calibration is necessary because our models are either statistical or physics-based models of existing aircraft and engines, whereas the NASA SUSAN study assumes advanced technologies that are outside the scope of the present work but that result in lower drag, a more efficient and lighter engine, and a lighter airframe.

#### 4.1.2 Optimization problem

Table 3 summarizes the SUSAN aircraft design optimizations we considered, defined as three problems with a different set of design variables and constraints each. Case 1 solves the sizing problem with a fixed load electrification factor that is calibrated to match the 65/35 thrust split proposed by the previous study [53]. Case 2 includes the load and source electrification factors as dynamic design variables to optimize the degree of electrification and climb battery-boost strategy. The electrification factors are then time-dependent variables and parametrized by a quadratic polynomial for each of the climb, cruise, or descent phases. Finally, Case 3 adds the rechargeable battery weight as an additional design variable. For all cases, we assume that the rechargeable battery is only used during climb, and fix the emergency single-use battery weight to that specified in the reference study [51]. All

**Table 3:** SUSAN conceptual design optimization problems

			Case 1	Case 2	Case 3
<b>Objective function</b>	Minimize	Fuel + battery total energy consumption	✓	✓	✓
<b>Design variables</b>	By varying	MTOW	✓	✓	✓
		Engine core sizing scaling factor	✓	✓	✓
		Aft-fuselage propulsor power rating	✓	✓	✓
		Motor power rating	✓	✓	✓
		Generator power rating	✓	✓	✓
		Rechargeable battery weight			✓
		Load elec. factor (takeoff, climb, cruise, descent)		✓	✓
		Source elec. factor (climb only)		✓	✓
		TMS design variables (see [34] for details)	✓	✓	✓
<b>Constraints</b>	Subject to	Sea-level static $T/W \geq 0.298$	✓	✓	✓
		Component power margins in takeoff and climb	✓	✓	✓
		Battery SOC $\geq 0.2$	✓	✓	✓
		Battery discharge C-rate $\leq 5.0$		✓	✓
		Motor and battery temperature limits	✓	✓	✓
		Takeoff BFL $\leq 8,000$ ft	✓	✓	✓

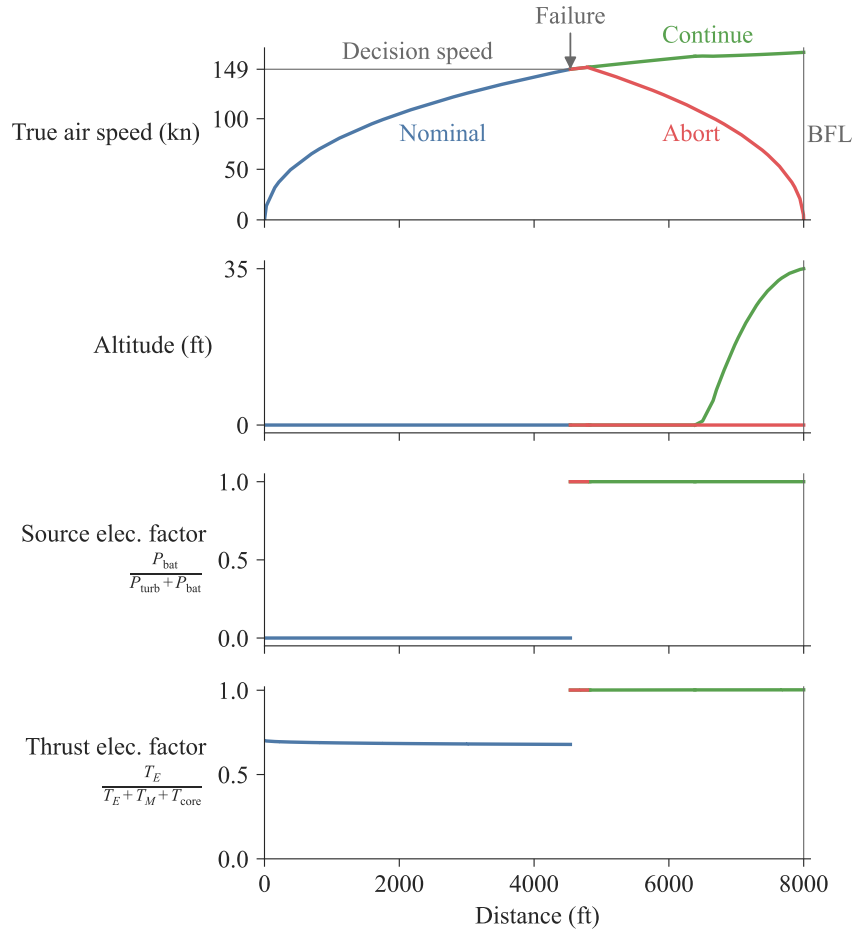
three optimization problems minimize the total energy consumption, which is the sum of the battery (electrical) energy consumption and fuel energy usage, the latter being computed by multiplying the fuel burn by the jet-A specific energy of 43.0 MJ/kg.

The component power margin constraints determine the propulsion component power ratings or the scaling factor. We impose these constraints throughout the takeoff and climb phases, as well as at the sea-level static maximum thrust condition for which we set the required thrust based on MTOW and a thrust to weight ratio of  $T/W = 0.298$  [51]. The battery SOC and discharge C-rate constraints size the rechargeable battery, and the TMS is sized by the motor and battery temperature limit.

We also include the takeoff balanced field length (BFL) limit as another sizing constraint. For this, we assume failure of the turbofan and 2 out of 16 electric propulsors at the decision speed, and that the aircraft would have to abort or continue takeoff in all-electric mode with the remaining electric propulsors and battery power. A field length of 8,000 ft is chosen since it is the Boeing 737 MAX 8’s runway length requirement for a standard day, sea level, and MTOW takeoff [54]. Figure 10 shows an example of the takeoff BFL analysis results.

### 4.1.3 Results

Table 4 compares the key sizing and performance results of the three optimization problems. Comparing Cases 1 and 2, the inclusion of dynamic electrification factors in Case 2 contribute to reducing the total fuel and battery energy consumption by 5.3%. As Fig. 11 shows, the optimized dynamic electrification allocates battery power during climb, with maximum source electrical power draw occurring at the top-of-climb point for which the engine is sized. This battery power assist reduces the engine core power requirement and thus downsizes the core power rating by 20.0%. This in turn results in a lower engine core weight, while also allowing the core to operate at a higher throttle level during cruise and thus at an improved cruise specific fuel consumption (SFC). Case 2 also allocates 69% of the cruise thrust to the electric propulsors compared to 65% in the baseline. This is because we assume that the electric propulsors are more efficient than the turbofan because of the higher total disk area. Hence, more load electrification improves overall efficiency. As a result of both



**Figure 10:** SUSAN takeoff BFL analysis results for the Case 1 optimization. We assume failure of the turbofan and 2 out of 14 electric propulsors at the decision speed, such that the aircraft would have to abort or continue the takeoff with only the remaining electric propulsors and battery power, which results in the source and thrust electrification factors being one at the decision speed.

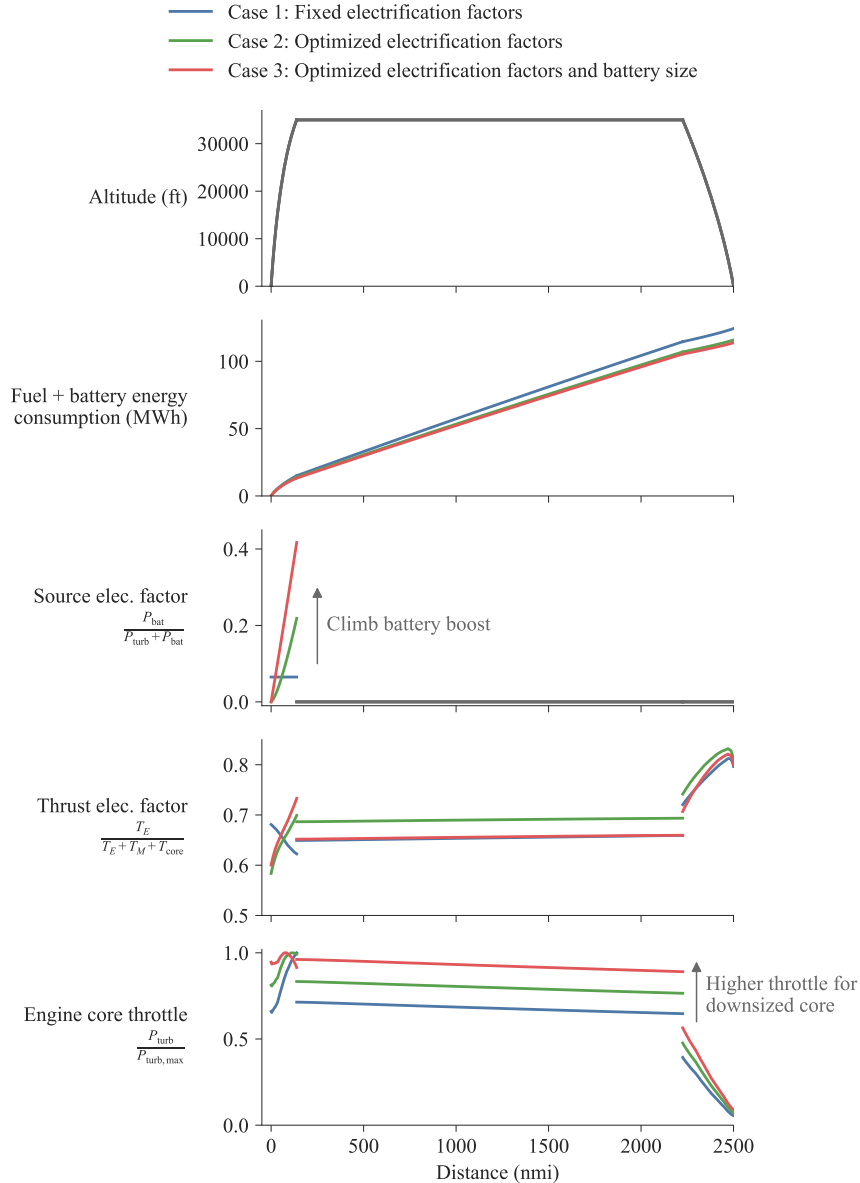
these effects, the aircraft-level thrust-specific fuel consumption (TSFC) at the start of cruise improves from 0.434 lb/lbf/hr in Case 1 to 0.419 lb/lbf/hr in Case 2.

Case 3 further downsizes the engine core by increasing the rechargeable battery size to more than double the baseline one. The MTOW is higher in Case 3 than in Case 2 because the battery weight increase is greater than the engine weight decrease, but the higher cruise efficiency of the optimally-sized core (with a start-of-cruise TSFC of 0.408 lb/lbf/hr) contributes to a mission fuel saving relative to both Cases 1 and 2.

Optimal conceptual design may vary depending on the modeling approaches, technology level assumptions, and additional sizing requirements and considerations. Therefore, these results should be viewed as preliminary, and further investigation is needed before drawing conclusions about the optimal design of the SUSAN configuration or similar concepts. Nevertheless, the example results presented here demonstrate the ability of our MDAO framework to effectively explore the conceptual design space of electrified aircraft.

**Table 4:** Comparison of SUSAN conceptual design optimization cases.

	Total energy (MWh)	MTOW (lb)	Fuel burn (lb)	Battery weight (lb)	Engine core power rating (hp)
<b>Case 1</b>	159.74	187,085	29,393	2,940	38,786
<b>Case 2</b>	151.21 (-5.3%)	178,693 (-4.5%)	27,812 (-5.4%)	2,940	31,015 (-20.0%)
<b>Case 3</b>	149.12 (-6.6%)	183,655 (-1.8%)	27,298 (-7.1%)	6,926 (+135.6%)	26,196 (-32.5%)



**Figure 11:** Comparison of the SUSAN mission profiles for the three conceptual design optimizations defined in Table 3. With time-variable power management, it uses battery power toward the end of the climb to shave off the top-of-climb power. This downsizes the engine core, allowing it to operate at a higher throttle setting in cruise. This figure does not include takeoff and landing phases.

## 4.2 Hybrid-Electric Commuter Aircraft: Dornier 228

### 4.2.1 Model Setup

The Dornier 228 [55] is a widely used 19-passenger twin-turboprop commuter aircraft. At an MTOW of 6,400 kg with a payload of 547 kg, it has a range of 1276 nmi, and is powered by two TPE331 engines with 5-bladed propellers. The popularity of this aircraft and the amount of data available have made it a frequent reference configuration for electrification studies of small aircraft [9, 56].

To demonstrate the capabilities of our framework, we optimize two versions of this aircraft for a payload of 547 kg and a range of 500 nmi: a conventional jet-fuel-power one and a hybrid-electric version. Our hybrid-electric configuration follows the aircraft studied by Kruger and Uranga [9]. This adds a battery with a pack-level specific energy of 575 Wh/kg, a motor and generator both with specific power of 12 kW/kg, and an inverter and rectifier both with specific power of 12 kW/kg. The battery specific energy and power estimates are based on what is called the intermediate technology level estimates for the 2035 time frame in that study, and we only include climb, cruise, and descent phases in our mission setup with no BFL analysis to match the referenced work [9].

The Dornier 228 cases in this work use the following models discussed in previous sections:

- Scaled turboshaft engine deck with specific-power-based weight estimation
- Thrust coefficient propeller model
- Power offtake regression model
- Torenbeek airframe weight model
- OpenConcept  $C_{D_0}$  build-up and Aviary's FLOPS induced drag model
- Tail area sizing by the volume coefficient methods

The hybrid-electric case further adds: constant-efficiency motors, generators, and converters with specific-power-based weight estimation; constant-efficiency battery model with pack-level power and energy sizing; thrust coefficient propeller model; lumped TMS model

### 4.2.2 Optimization problem

Table 5 summarizes the aircraft design optimization problems, with both cases optimized to minimize the total mission energy consumption. The conventional case optimizes a configuration close to the actual aircraft. The hybrid-electric case adds electrification by including battery energy in addition to jet fuel, and the electrification factors are allowed to vary throughout the mission as was done for Cases 2 and 3 of the SUSAN example.

Power margin constraints determine the power ratings or scaling factors for the propulsion system components. These ensure that the required power does not exceed the power rating, which has an upper bound. In the conventional case, this is only necessary in climb because the turboshaft power is highest in climb. In the hybrid-electric case, all phases must be constrained, as this assumption no longer holds with the addition of a battery boost. The propulsion system is primarily sized by the sea-level static (SLS) thrust-to-weight ratio. In the hybrid-electric case, the optimizer also controls the load and source electrification factors to best meet this constraint. The battery is subject to a minimum SOC of 20% and a maximum discharge C-rate of 5.0.

**Table 5:** Dornier 228 conceptual design optimization problems

			Conventional	Hybrid-electric
<b>Objective function</b>	Minimize	Fuel + battery total energy consumption	✓	✓
<b>Design variables</b>	By varying	MTOW	✓	✓
		Turboshaft scaling factor	✓	✓
		Propeller power rating	✓	✓
		Motor power rating		✓
		Generator power rating		✓
		Battery weight		✓
		Load elec. factor (all mission phases)		✓
		Source elec. factor (all mission phases)		✓
		Load elec. factor, pre-mission SLS condition		✓
		Source elec. factor, pre-mission SLS condition		✓
<b>Constraints</b>	Subject to	Sea-level static $T/W \geq 0.33$	✓	✓
		Component power margins in climb	✓	✓
		Component power margins in cruise and descent		✓
		Battery SoC $\geq 0.2$	✓	✓
		Battery discharge C-rate $\leq 5.0$		✓

### 4.2.3 Results

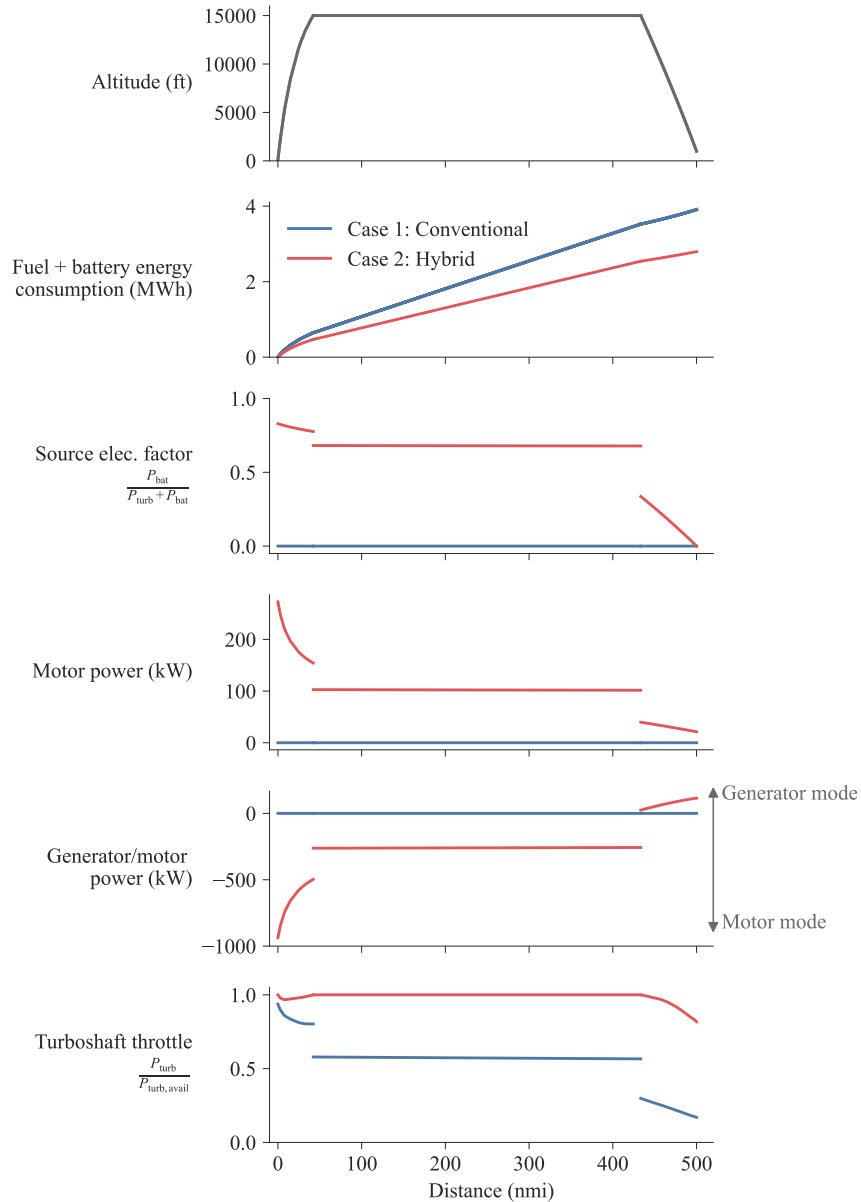
We first tune the conventional case to calibrate the lift-to-drag ratio  $L/D$  to a value of 16 per the CFD analysis of the Dornier 228 [56] by adding a calibration factor to the drag, and maintain this calibration for both optimization cases. With the weight model from Torenbeek described in Section 3.6.3, the OEW is already close to the manufacturer’s value of 14,110 lb [55], so no weight calibration is needed.

Table 6 compares the key sizing and performance parameters of the two optimization problems, and Fig. 12 shows the variation along the mission of selected properties. Both aircraft fly the same mission, but the electrified vehicle uses 28.5% less total mission energy: 3.91 MWh of energy for the conventional vehicle versus 2.79 MWh for the electrified one. This is explained by the effect of the source electrification on the turboshaft size and operation (throttle profile). By pulling the majority of the required power from the battery during climb and then about half of it during cruise, the turboshaft engine required by the electrified aircraft can be significantly smaller (power- and thus weight-wise) than the one on the conventional vehicle. Additionally, in the conventional case, the turboshaft is sized for top of climb, so the throttle is at its maximum at the top of climb but at a much lower level at other flight conditions, which causes the engine to be operated at a lower efficiency point for most of the mission. Thanks to the availability of battery power during climb, the electrified propulsion system can be smaller and also operate at higher efficiency (maximum throttle) throughout climb and cruise, both effects contributing to reducing the overall in-flight energy consumption.

The motor power of the electrified case shows where a significant amount of the propulsive power previously produced by the turboshaft in the conventional case is now coming from. As Fig. 12 shows, the electrical motors draw the most power during climb, the segment for which the turboshaft would otherwise do the most work and over which the source electrification is highest. The other component related to downsizing the turboshaft is the generator. In the unified propulsion system, the generator links power between mechanical and electrical sub-branches but can let power flow in either direction, acting as either a generator or a motor (Figure 4). Positive link power denotes electrical power generation, or power flowing from mechanical to electrical branches. Negative link

**Table 6:** Comparison of Dornier 228 conceptual design optimizations for Case 1 (conventional) and Case 2 (hybrid-electric).

	Total energy (MWh)	MTOW (lb)	Fuel burn (lb)	Battery weight (lb)	Engine core power rating (hp)
<b>Case 1</b>	3.91	14,838	721.33	0.0	670.51
<b>Case 2</b>	2.79 (-28%)	20,929 (+41%)	286.93 (-60%)	6,000.0	171.16 (-74%)



**Figure 12:** Comparison of the Dornier 228 mission profiles for the conventional and hybrid-electric configuration optimizations. By varying the electrification factor, the battery is used for power in climb and cruise, allowing the turboshaft to downsize.

power denotes electrical power consumed, or going from electrical to mechanical, in which case the generator component acts as a motor. The propulsion system as currently implemented does not capture the details of component location, but in the negative-link-power case, the system might benefit from the use of an integrated turbogenerator in which a generator is embedded within the thermal engine. For optimal power flow usage, the link functions as a motor during climb and cruise, sending power from the battery to the mechanical branch. Conversely, the link component acts as a generator during descent, sending power to the electrical system. This extra power allows the turbine core to be sized even smaller, reducing weight and increasing efficiency.

These results demonstrate our framework’s capabilities and show the promise of designing an electrified aircraft and its operation together, in particular, how the battery-engine combination is best exploited during operation.

## 5 Conclusions

This work incorporates existing and novel component models for advanced aircraft design into an MDAO framework, with the aim of being able to design vehicles that leverage propulsion system electrification and other novel integration technologies. This framework offers a range of aircraft subsystem models, with multiple model fidelity options for engines, motors, generators, batteries, TMS, empty weight estimation, aerodynamics, and aero-propulsive integration models.

Our framework is built on NASA’s OpenMDAO and Aviary, enabling efficient gradient-based optimization with coupled analytic derivatives. Additionally, we adopted Aviary’s modular philosophy, allowing “plug-and-play” assembly of the subsystem models to represent any aircraft concepts. Computational efficiency is crucial to being able to explore a wide design space, and is ensured by the use of gradient-based optimization and a multi-fidelity model tool-set to better balance model fidelity against computational runtime.

The capabilities of the framework are demonstrated via electrified aircraft conceptual design optimization for the NASA SUSAN single-aisle configuration and for a 19-passenger commuter aircraft. The results show that our MDAO framework effectively optimizes the propulsion component sizing and the dynamic electrification strategy to minimize mission-level energy consumption. This demonstrates the capability of our framework to model and optimize a complex electrified aircraft configuration.

The framework is being actively developed, and a future publication will present further implementation of additional higher-fidelity models. Upcoming publications by the authors and project collaborators will discuss the subsystem modeling approaches in greater detail and present the aircraft-level trade studies to show subsystem effects on system level performance. These research efforts ultimately contribute to identifying the advanced technologies that have the greatest potential to provide radical improvements in aircraft energy efficiency, thereby helping to enable sustainable air transportation.

## Appendix

### A Engine Mass Regression Model

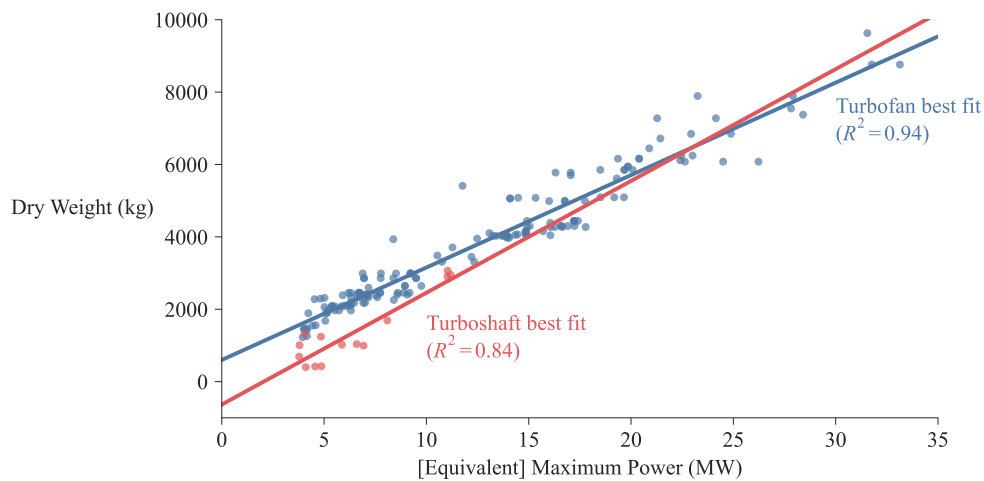
Our MDAO framework is intended to design a wide variety of aircraft concepts and propulsion system architectures, and thus requires a weight model for engines whose power is outside of the

applicable range of Raymer’s Eq. (7). Other existing engine weight models are not publicly available and/or require detailed knowledge of the internal engine architecture, information that may not be available during conceptual design.

We have therefore developed a regression for weight estimation of turboshafts using a data-driven approach under the following definitions and assumptions:

- “Dry”, or “uninstalled”, refers to a gas turbine engine weight (or mass) as reported by the manufacturer. This excludes mounting, nacelle, and accessory weights.
- All gas turbine engines that are not turbofans are referred to as “turboshaft engines”, the distinction coming from the fact that these engine weights are reported *without* the propulsive element (*e.g.*, propeller or fan and casing).
- Turboshaft data used for the mass regression comes exclusively from all-axial turboshafts, to ensure that the only difference in mass between a turboshaft and a turbofan engine is indeed from the turbofan’s built-in propulsor.
- To directly compare turboshafts and turbofans, the latter of which often report static thrust production instead of power, we use a conversion factor of 15.51 N static thrust per 1 kW of power (or 2.6 lbf  $\approx$  1 hp), following the FAA’s *Aircraft Technician Maintenance* handbook, a conversion used since 1979 [57, 58].
- Data for both turbofans and turboshafts is sourced from the FAST Aerobase [59], a public repository of aircraft and gas turbine engine data.

Applying the thrust-to-power conversion to turbofan data, and excluding all engines of both turbofan and turboshaft classes that are rated lower than 3,728 kW yields the data shown in Fig. 13. It is clear that while large turboshaft engines are comparable in power class to some turbofans, the data for the latter is much less sparse. Both curves exhibit linear trends in their mass response, and we therefore assume a linear relationship between mass and power. Because of the sparsity of the available turboshaft data, a higher-order curve would be overfit, as would a mass model that were to include additional parameters such as overall pressure ratio or turbine entry temperature.



**Figure 13:** Data points and the best linear fit of dry mass versus maximum power (or its equivalent for turbofans) for both turbofan and turboshaft engines. This figure includes only engines that are rated higher than 3,728 kW, the upper limit of the Raymer model of Eq. (7).

Fitting linear relationships to the data that minimizes the least squared residuals yields high coefficients of determination (a goodness-of-fit metric) of  $R^2 = 0.94$  and  $R^2 = 0.84$  for turbofans and turboshafts, respectively. The fits are shown in Fig. 13. While these curves are appropriate in a purely statistical sense, they do not yield physical results, such as engines that can produce power while having zero mass. Additionally, the best-fit curves suggest that very large turboshafts would weigh more than their turbofan counterparts at the same power sizing, which contradicts the fact that a turbofan is assumed identical to its turboshaft core with the addition of an integrated propulsor (and therefore its weight).

Two changes are made to the curve fits to counter these deficiencies. First, we enforce that both curves intersect the origin to impose zero weight for a zero-power engine. The second modification corresponds to setting the difference between the slopes of the turbofan and turboshaft fit lines to be equal to the weight of a fan and its casing at a given power. While weight data of fans and fan casings is not readily available, isolated values are reported by Tong and Naylor [60] for the purpose of validating the WATE++ code on a GE90-class engine: they use a dry mass for the notional GE90 of 7,452 kg, with the fan and fan casing accounting for 1,826 kg. Since this reported notional dry mass is lighter than actual GE90 variant data, which ranges from 7,892 to 8,761 kg [59], and since the WATE++ validation does not specify a thrust value for the notional engine, we assume that the lightest variant, the GE90-94B, is the closest one to the notional GE90, and therefore use its maximum power of 27,906 kW as the power value at which to calibrate the fan and fan casing mass needed to set the regression curve slopes. Assuming a linear scaling for the fan and casing weight, the difference between the actual and notional GE90 mass can be expressed as

$$m_{\text{turbofan}} - m_{\text{turboshaft}} = 1826 \text{ kg} \times \frac{7892 \text{ kg}}{7452 \text{ kg}} \text{ at } 27,906 \text{ kW power} . \quad (33)$$

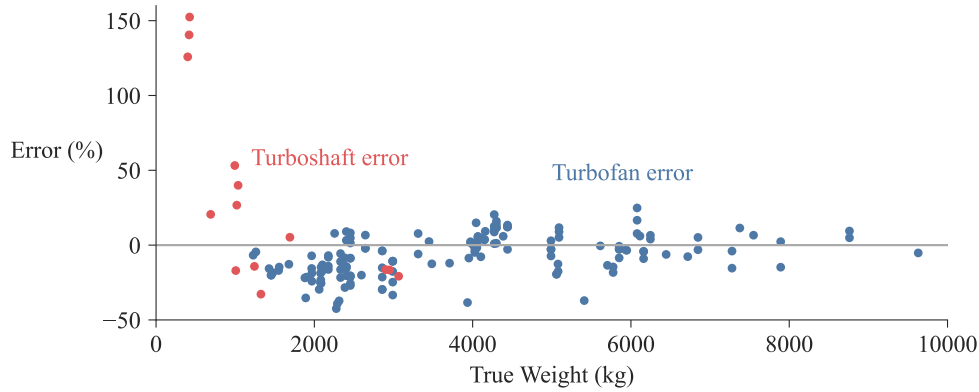
This therefore prescribes the difference between the slopes of the turbofan and turboshaft fit lines to be equal to 0.0693 kg/kW, which we use as calibration for the ducted fan mass model.

With these two constraints imposed on the linear fits, we sweep through the slopes of the models to calculate the root-mean-squared error (RMSE) relative to the true data, and minimize the sum of the RMSE of turbofans and turboshafts with respect to the turboshaft slope, to find the best value for the turboshaft slope to be 0.22 kg/kW. Together with the ducted-fan mass calibration, we can write the required turboshaft and ducted fan mass models as, respectively,

$$m_{\text{turboshaft}} = 0.22 P_{\text{rating, turb}} , \quad (34)$$

$$m_{\text{ducted fan}} = 0.0693 P_{\text{rating, fan}} , \quad (35)$$

in which masses are in kilograms, powers in kilowatts, and the power rating  $P_{\text{rating}}$  is defined at sea level static conditions. Equivalently, these results can be expressed in terms of specific power as 4.54 kW/kg for a turboshaft and 14.43 kW/kg for a ducted fan. Since we are constrained to a low-order fitting due to the limited amount of data available, the accuracy of the model is limited. The modeling error can be seen Fig. 14, and the fitting itself has  $R^2$  values of 0.915 and 0.767 for turbofans and turboshafts, respectively. The accuracy of this engine mass model is nevertheless assumed to be sufficient for the purposes of this work, and will be improved in future work. Our framework allows the user to easily change the engine mass model if a more accurate one is available.



**Figure 14:** Linear fitting error versus actual weight comparisons for turbofan and turboshaft data.

## B IDF Decoupling Approach to Improve Robustness

Numerical convergence in Aviary and our framework is generally sensitive to the initial guess, especially because it uses a collocation method for mission analysis. When the initial guess is poor, optimization has to traverse a highly infeasible region, often with non-physical mission and power profiles. Lower-fidelity models can usually handle these poor inputs, and the optimizer would eventually find a physical, well-defined design space. However, most higher-fidelity models in our framework do not work well for poor inputs, resulting in solver divergence and optimization failure.

To alleviate this issue, we implement an optional IDF approach [14, Sec. 13.4] for the pyCycle engine model, the battery ECM, and the OpenConcept TMS model. This is similar to the PyBaMM battery model coupling approach discussed in Sec. 3.5.3. IDF formulation delegates the component coupling to the optimization level and decouples model evaluations, thereby avoiding the propagation of poor coupling variables across components. Once converged, the solution is equivalent to the one of the original, non-IDF problem.

Here, we explain the IDF decoupling approach for the pyCycle turboshaft model as an example. By default, the shaft power input to the pyCycle turboshaft component is connected from the upstream propulsion system, which determines the power flow given the mission profile and thrust requirement from Aviary. A poor initial guess may result in requiring negative shaft power from the turboshaft model, which causes the pyCycle solver to fail. To avoid this, we introduce additional optimization variables, called *target variables*, to represent the turboshaft power profile. Instead of connecting the power from the propulsion system to the turboshaft component, the optimization process directly controls the power input to the shaft power. A series of equality constraints impose consistency between the power target variables and the actual turboshaft power requested by the propulsion system.

This approach contributes to improving solver robustness because we set bounds on the power target variables so they never become negative (or below idle power), protecting the pyCycle turboshaft component against a non-physical shaft power input. The variable bound constraints are generally easier for optimizers to satisfy than nonlinear constraints, and certain optimizers (*e.g.*, SNOPT) never violate bound constraints during search iterations. However, the downside is the larger optimization problem size, which generally increases the computational cost. Thus, we trade computational efficiency for robustness when using higher-fidelity models.

Our framework also provides an option to use the same decoupling strategy for the battery ECM

power inputs and the TMS heat inputs. These may improve the convergence when a good initial guess is not available.

## Acknowledgments

This work was partly supported by the National Aeronautics and Space Administration (NASA) under contract number 80GRC024CA036 as part of the Advanced Aircraft Concepts for Environmental Sustainability (AACES) 2050 program, and carried out under Electra’s leadership. The authors would like to acknowledge contributions from the rest of the team, including partners at Electra, the University of Michigan, Massachusetts Institute of Technology, the University of California Irvine, Hinetics, and Honeywell. We would like to thank, in particular, Ara Mahseredjian, Amal Sebastian, Aditeya Shukla, Anushka Tahiliani, and Parker Vascik for their intellectual and material contributions to the development of this paper. We acknowledge the technical support by the OpenMDAO and Aviary teams at NASA, in particular, Eliot Aretskin-Hariton, Robert Falck, Jason Kirk, and Kenneth Moore.

## References

- [1] Brelje, B., and Martins, J. R. R. A., “Electric, Hybrid, and Turboelectric Fixed-Wing Aircraft: A Review of Concepts, Models, and Design Approaches,” *Progress in Aerospace Sciences*, Vol. 104, 2019, pp. 1–19. doi:10.1016/j.paerosci.2018.06.004.
- [2] Welstead, J., and Felder, J. L., “Conceptual Design of a Single-Aisle Turboelectric Commercial Transport with Fuselage Boundary Layer Ingestion,” *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-1027.
- [3] Yildirim, A., Gray, J. S., Mader, C. A., and Martins, J. R. R. A., “Boundary Layer Ingestion Benefit for the STARC-ABL Concept,” *Journal of Aircraft*, Vol. 59, No. 4, 2022, pp. 896–911. doi:10.2514/1.C036103.
- [4] Jansen, R. H., Kiris, C. C., Chau, T., Kenway, G. K. W., Machado, L. G., Duensing, J. C., Mirhashemi, A., Haglage, J. M., Dever, T. P., Chapman, J. W., French, B. D., Goodnight, T. W., Miller, L. R., Litt, J. S., Denham, C. L., Lynde, M., Campbell, R., Hiller, B., and Heersema, N., “Subsonic Single Aft Engine (SUSAN) Transport Aircraft: Concept and Trade Space Exploration,” *AIAA SCITECH 2022 Forum*, AIAA, San Diego, CA & Virtual, 2022. doi:10.2514/6.2022-2179.
- [5] Cinar, G., “A Methodology for Dynamic Sizing of Electric Power Generation and Distribution Architectures,” Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, 2018. URL <http://hdl.handle.net/1853/60754>.
- [6] Cinar, G., Garcia, E., and Mavris, D. N., “A Framework for Electrified Propulsion Architecture and Operation Analysis,” *Aircraft Engineering and Aerospace Technology*, Vol. 92, No. 5, 2019, pp. 675–684. doi:10.1108/AEAT-06-2019-0118.
- [7] Brelje, B. J., and Martins, J. R. R. A., “Development of a Conceptual Design Model for Aircraft Electric Propulsion with Efficient Gradients,” *Proceedings of the AIAA/IEEE Electric Aircraft Technologies Symposium*, Cincinnati, OH, 2018. doi:10.2514/6.2018-4979.

- [8] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. doi:[10.1007/s00158-019-02211-z](https://doi.org/10.1007/s00158-019-02211-z).
- [9] Kruger, M., and Uranga, A., "Benefits and Challenges of Electric Propulsion for Commuter Aircraft," *Journal of Aircraft*, Vol. 61, No. 1, 2024. doi:[10.2514/1.C037284](https://doi.org/10.2514/1.C037284).
- [10] Aretskin-Hariton, E., Gratz, J., Kirk, J., Lyons, K., Jasa, J., Moore, K., Falck, R., Caldwell, D., Kuhnle, C., Recine, C., Hendricks, E., and Olson, E., "Multidisciplinary Optimization of a Transonic Truss-Braced Wing Aircraft using the Aviary Framework," *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024. doi:[10.2514/6.2024-1084](https://doi.org/10.2514/6.2024-1084).
- [11] Gratz, J., Kirk, J., Recine, C., Jasa, J., Aretskin-Hariton, E., Moore, K., and Marfatia, K., "Aviary: An Open-Source Multidisciplinary Design, Analysis, and Optimization Tool for Modeling Aircraft With Analytic Gradients," *Proceedings of the Aviation Forum*, Las Vegas, NV, 2024. doi:[10.2514/6.2024-4219](https://doi.org/10.2514/6.2024-4219).
- [12] Falck, R., Gray, J. S., Ponnappalli, K., and Wright, T., "dymos: A Python package for optimal control of multidisciplinary systems," *Journal of Open Source Software*, Vol. 6, No. 59, 2021, p. 2809. doi:[10.21105/joss.02809](https://doi.org/10.21105/joss.02809).
- [13] Hwang, J. T., and Martins, J. R. R. A., "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives," *ACM Transactions on Mathematical Software*, Vol. 44, No. 4, 2018, p. Article 37. doi:[10.1145/3182393](https://doi.org/10.1145/3182393).
- [14] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, Cambridge, UK, 2022. doi:[10.1017/9781108980647](https://doi.org/10.1017/9781108980647), URL <https://mdobook.github.io>.
- [15] Martins, J. R. R. A., and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. doi:[10.2514/1.J052184](https://doi.org/10.2514/1.J052184).
- [16] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. doi:[10.1137/S0036144504446096](https://doi.org/10.1137/S0036144504446096).
- [17] Wächter, A., and Biegler, L. T., "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. doi:[10.1007/s10107-004-0559-y](https://doi.org/10.1007/s10107-004-0559-y).
- [18] Wu, E., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems," *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. doi:[10.21105/joss.02564](https://doi.org/10.21105/joss.02564).
- [19] McCullers, A., *FLOPS User's Guide*, , Release 8.12, NASA Langley Research Center, Hampton, VA, June 2010.
- [20] Hague, D., "GASP - General Aviation Synthesis Program, Vol. 1," Tech. rep., 1978. URL <https://ntrs.nasa.gov/api/citations/19810010562/downloads/19810010562.pdf>, NASA Technical Report. Accessed Nov. 25, 2025.

- [21] Falck, R. D., and Gray, J. S., "Optimal control within the context of multidisciplinary design, analysis, and optimization," *AIAA Scitech 2019 Forum*, 2019. doi:10.2514/6.2019-0976.
- [22] Capristan, F. M., and Welstead, J. R., "An Energy-Based Low-Order Approach for Mission Analysis of Air Vehicles in LEAPS," *AIAA SciTech 2018 Forum*, 2018. doi:10.2514/6.2018-1755.
- [23] Kruger, M., Byahut, S., Uranga, A., Dowdle, A., Gonzalez, J., and Hall, D. K., "Electrified Aircraft Trade-Space Exploration," *2018 Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-4227.
- [24] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.
- [25] Hendricks, E. S., and Gray, J. S., "pyCycle: A Tool for Efficient Optimization of Gas Turbine Engine Cycles," *Aerospace*, Vol. 6, No. 87, 2019. doi:10.3390/aerospace6080087.
- [26] Claus, R. W., Evans, A. L., Lylte, J. K., and Nichols, L. D., "Numerical Propulsion System Simulation," *Computing Systems in Engineering*, Vol. 2, No. 4, 1991, pp. 357–364. doi:10.1016/0956-0521(91)90003-N.
- [27] Kaneko, S., and Martins, J. R. R. A., "Simultaneous Design and Trajectory Optimization Strategies for Computationally Expensive Models," *AIAA Journal*, Vol. 63, No. 2, 2025, pp. 420–438. doi:10.2514/1.J063976.
- [28] Kaneko, S., and Martins, J. R. R. A., "Accelerating coupled derivative computations with nested-hierarchical linear solver strategies," *Optimization and Engineering*, 2025. doi:10.1007/s11081-025-09965-0.
- [29] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, 7<sup>th</sup> ed., AIAA, Reston, VA, 2024.
- [30] Torenbeek, E., *Synthesis of Subsonic Airplane Design*, 6<sup>th</sup> ed., Delft University Press and Kluwer Academic Publishers, Delft, NL, 1990.
- [31] Byahut, S., "Modeling and Analysis of Propulsion Systems and Components for Electrified Commercial Aircraft," Ph.D. dissertation, University of Southern California, Los Angeles, CA, May 2021.
- [32] Husain, E., and Nema, R. S., "Analysis of Paschen Curves for Air, N<sub>2</sub> and SF<sub>6</sub> Using the Townsend Breakdown Equation," *IEEE Transactions on Electrical Insulation*, Vol. EI-17, No. 4, 1982, pp. 350–353. doi:10.1109/TEI.1982.298506.
- [33] Byahut, S., and Uranga, A., "Power Distribution and Thermal Management Modeling for Electrified Aircraft," *2020 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, 2020. doi:10.2514/6.2020-3578.
- [34] Adler, E. J., Brelje, B. J., and Martins, J. R. R. A., "Thermal Management System Optimization for a Parallel Hybrid Aircraft Considering Mission Fuel Burn," *Aerospace*, Vol. 9, No. 5, 2022. doi:10.3390/aerospace9050243.

- [35] Garg, D., Patterson, M. A., Francolin, C., Darby, C. L., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method," *Computational Optimization and Applications*, Vol. 49, 2011, pp. 335–358. doi:10.1007/s10589-009-9291-0.
- [36] Doyle, M., Fuller, T. F., and Newman, J., "Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell," *Journal of The Electrochemical Society*, Vol. 140, No. 6, 1993, p. 1526. doi:10.1149/1.2221597.
- [37] Sulzer, V., Marquis, S. G., Timms, R., Robinson, M., and Chapman, S. J., "Python Battery Mathematical Modelling (PyBaMM)," *Journal of Open Research Software*, Vol. 9, No. 1, 2021, p. 14. doi:10.5334/jors.309.
- [38] Gardner, D. J., Reynolds, D. R., Woodward, C. S., and Balos, C. J., "Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 48, No. 3, 2022, pp. 1–24. doi:10.1145/3539801.
- [39] Serban, R., Petra, C., Hindmarsh, A. C., Balos, C. J., Gardner, D. J., Reynolds, D. R., and Woodward, C. S., "User Documentation for IDAS, v6.5.0," , 2025. URL <https://sundials.readthedocs.io/en/latest/idas>, accessed Nov. 25, 2025.
- [40] Adler, E. J., and Martins, J. R. R. A., "Blended wing body configuration for hydrogen-powered aviation," *Journal of Aircraft*, Vol. 61, No. 3, 2024. doi:10.2514/1.C037582.
- [41] Bradley, K. R., "A Sizing Methodology for the Conceptual Design of Blended-Wing-Body Transports," Tech. Rep. ASA/CR-2004-213016, NASA Langley Research Center, Hampton, VA, Sept 2004.
- [42] Mokotoff, P. R., Arnson, M., Wang, Y.-C., and Cinar, G., "FAST: A Future Aircraft Sizing Tool for Conventional and Electrified Aircraft Design," *Journal of Aircraft*, 2025. doi:10.2514/1.c038452.
- [43] Arnson, M. G., Aljaber, R., and Cinar, G., "Predicting Aircraft Design Parameters Using Gaussian Process Regressions on Historical Data," *AIAA SciTech 2025 Forum*, 2025, p. 1287. doi:10.2514/6.2025-1287.
- [44] Feagin, R. C., and Morrison, W. D., "Delta method, an empirical drag buildup technique," Tech. Rep. NASA-CR-151971, NASA Ames Research Center, Dec. 1978. URL <https://ntrs.nasa.gov/citations/19790009630>.
- [45] Roskam, J., *Airplane Design, Volumes 1-8*, Roskam Aviation and Engineering Corporation, 1989.
- [46] McDonald, R. A., and Gloudemans, J. R., "Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design," *AIAA Scitech*, San Diego, CA, 2022. doi:10.2514/6.2022-0004.
- [47] Uranga, A., Drela, M., Hall, D. K., and Greitzer, E. M., "Analysis of the Aerodynamic Benefit from Boundary Layer Ingestion for Transport Aircraft," *AIAA Journal*, Vol. 0, No. 0, 2018, pp. 1–11. doi:10.2514/1.J056781.

- [48] Drela, M., "Power Balance in Aerodynamic Flows," *AIAA Journal*, Vol. 47, No. 7, 2009, pp. 1761–1771. doi:10.2514/1.42409.
- [49] Arnson, M. G., Cinar, G., Waddington, E., Ansell, P. J., Clarke, M. A., de Vries, R., Salucci, F., Prabhakar, N., Gladin, J. C., Shi, M., Lovelace, E., and de Bock, P., "System-Level Energy Pack Requirements for Sustainable Commercial Aviation," *AIAA AVIATION FORUM AND ASCEND 2024*, American Institute of Aeronautics and Astronautics, 2024. doi:10.2514/6.2024-3828.
- [50] Brejle, B. J., Anibal, J., Yildirim, A., Mader, C. A., and Martins, J. R. R. A., "Flexible Formulation of Spatial Integration Constraints in Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2571–2580. doi:10.2514/1.J058366.
- [51] Chau, T., and Duensing, J. C., "Conceptual Design of the Hybrid-Electric Subsonic Single Aft Engine (SUSAN) Electrofan Transport Aircraft," *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2024. doi:10.2514/6.2024-1326.
- [52] Wang, Y.-C., Stockhausen, M., Mokotoff, P. R., Arnson, M., and Cinar, G., "SUBsonic Single Aft eNGine (SUSAN) System Integration Analysis With the Future Aircraft Sizing Tool (FAST)," *AIAA SCITECH 2025 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2025. doi:10.2514/6.2025-2376.
- [53] Chapman, J. W., Kratz, J. L., Dever, T. P., Mirhashemi, A., Stalcup, E. J., Sixel, W. R., Woodworth, A. A., and Jansen, R. H., "Update on SUSAN Concept Vehicle Power and Propulsion System," *AIAA SciTech 2023 Forum*, National Harbor, MD & Online, 2023. doi:10.2514/6.2023-1749.
- [54] Boeing Commercial Airplanes, "737 MAX Airplane Characteristics for Airport Planning," Tech. Rep. D6-38A004, November 2018.
- [55] RUAG Aerospace Services GmbH, "Dornier 228 Advanced Commuter (AC) Facts & Figures," , 2009. URL <https://www.ga-ats.com/en/>, accessed Nov. 25, 2025.
- [56] Finger, D. F., Vries, R. D., Vos, R., Braun, C., and Bil, C., "Cross-Validation of Hybrid-Electric Aircraft Sizing Methods," *Journal of Aircraft*, Vol. 59, No. 3, 2022, pp. 742–760. doi:10.2514/1.C035907.
- [57] Federal Aviation Administration, *Aviation Maintenance Technician Handbook – Powerplant*, faa-h-8083-32b ed., U.S. Department of Transportation, Federal Aviation Administration, Flight Standards Service, Washington, D.C., 2023. URL [https://www.faa.gov/regulations\\_policies/handbooks\\_manuals/aviation/amt\\_handbook\\_powerplant](https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/amt_handbook_powerplant).
- [58] Federal Aviation Administration, *Airframe and Powerplant Mechanics Powerplant Handbook*, first revision ed., Advisory Circular AC 65-12A, U.S. Department of Transportation, Federal Aviation Administration, Flight Standards Service, Washington, D.C., 1976. URL [https://www.faa.gov/regulations\\_policies/handbooks\\_manuals/aircraft/media/faa-h-8083-32-ch1.pdf](https://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/media/faa-h-8083-32-ch1.pdf).
- [59] Acar, H., Arnson, M., Tsai, M., and Cinar, G., "Historical Trends and Future Projections of Key Performance Parameters in Aircraft Design," *Journal of Aircraft*, 2025. doi:10.2514/1.c038340.
- [60] Tong, M. T., and Naylor, B. A., "An Object-Oriented Computer Code for Aircraft Engine Weight Estimation," Technical Memorandum NASA/TM-2009-215656, GT2008-50062, NASA Glenn Research Center, Cleveland, Ohio, January 2009. URL <https://ntrs.nasa.gov/citations/20090022277>.