NASA/CR-20250007059

NASA Model-Based Systems Analysis and Engineering Final Report

Jason Corman, Jimmy Tai, Evan Harrison, Jai Ahuja, Christian Perron,
Bogdan-Paul Dorca, Josef D'cruz, and Samuel Moore
Georgia Institute of Technology, Atlanta, Georgia

August 2025

NASA/CR-20250007059

NASA Model-Based Systems Analysis and Engineering Final Report

Jason Corman, Jimmy Tai, Evan Harrison, Jai Ahuja, Christian Perron,
Bogdan-Paul Dorca, Josef D'cruz, and Samuel Moore
Georgia Institute of Technology, Atlanta, Georgia

Prepared under Contract 80GRC024CA002

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

August 2025

Acknowledgments

Salil Sodhi, an undergraduate research assistant at the time he was on the project, helped debug the conventional vehicle mode in its infancy in the kinetic mission analysis approach. Chloé Pothérat, a former graduate research assistant working for ASDL, worked with Aviary in initial exploratory capacity. Last but certainly not least, Dr. Chung Lee was on the team which drafted the initial proposal for this project and attended the initial project presentation.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by expert reviewer(s).

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

NASA Model-Based Systems Analysis and Engineering Final Report

Principal Investigator

Prof. Dimitri Mavris

Aerospace Systems Design Laboratory

Georgia Institute of Technology, Atlanta, GA, 30332, USA

1 May 2025

Authors

Dr. Jason Corman
Dr. Jimmy Tai
Dr. Evan Harrison
Dr. Jai Ahuja
Dr. Christian Perron

Research Assistants

Bogdan-Paul Dorca
Josef D'cruz
Samuel Moore

Contents

Nomenclature

Variable	Meaning
$\vec{a}$	acceleration vector
a_x	acceleration component in x direction
γ	flight path angle as defined by velocity vector
$\dot{\gamma}$	Time rate of change of flight path angle
C_D	aircraft three-dimensional drag coefficient
C_{Fx}	Coefficient of net aeropropulsive force in streamwise direction
D	Drag force on aircraft (streamwise by definition)
h	Altitude
M_∞	Freestream Mach number
P	Power setting for propulsion system
q	Dynamic pressure
Q_V	Variability characteristic of parameter V in input domain model
Q_{Fh}	Variability characteristic of net horizontal force in input domain model
$Q_{Fh}[3]$	Third block within characteristic Q_{Fh} in input domain model
S	Reference wing planform area
s	Horizontal distance measured from mission point of reference
s_{phase}	Horizontal distance measured from beginning of current mission phase
R	Range for entire mission
T_{SL}	Sea level, static thrust of all engines on aircraft
$\vec{V}$	velocity vector
V	magnitude of velocity relative to static atmosphere
$\dot{V}$	Time rate of change of velocity magnitude
W_f	Weight of fuel on aircaft
$W_{f,design}$	Weight of *burned* fuel for design mission (not reserve mission)
W_b	Weight of fuel burned over the mission
W_0	Ramp weight: weight before engine start on ramp
W_{TO}	Takeoff weight: weight immediately prior to takeoff
x	Aircraft configuration design variables
x_{min}	Aircraft configuration design variables' lower bounds
x_{max}	Aircraft configuration design variables' upper bounds

Chapter 1

Introduction

Many novel aircraft concepts exploit complex phenomena or coupled physics, such as aero-propulsion interactions that can only be captured in MDA with higher fidelity analyses. The results of these physics analyses are used in mission and sizing analyses to yield top-level system metrics such as fuel burn and take-off field length. Yet, preliminary investigations suggest that traditional mission analysis codes can mask the key physics, such that there are serious errors in the top-level metrics. One recent example is that of coupled aero-propulsion for cases such as boundary layer ingestion, over-wing nacelles, or flow control devices. In such cases, the traditional decomposition of forces may not directly capture complex effects, i.e., where the engine affects the airframe aerodynamics, the airframe flowfield influences the engine cycle, and vice versa.

This report presents the development of a higher-fidelity mission sizing and synthesis code that uses generalized dynamics description of air vehicle motion, the use of which reduces the errors produced by the traditional decomposition of forces. The developed capability was used to analyze integrated models of aircraft relevant to the Sustainable Flight National Partnership (SFNP), such as the Sustainable Flight Demonstrator (SFD) transonic truss-braced wing (TTBW) aircraft [1].

The present work aims to augment the existing NASA mission analysis tools, including methods from the General Aviation Sizing and synthesis Program (GASP) [2] and the Flight OPtimization System (FLOPS) [3], now re-implemented in the Aviary framework [4]. It builds on a heritage of aircraft system-level design and analysis tools, particularly those integrated into the Environmental Design Space (EDS)[5] and its FLOPS mission analysis module. Specifically, it attempts to strip away thrust and drag book-keeping assumptions — traditionally useful for early conceptual design for their simplicity — to make mission analysis and design more flexible and vehicle-agnostic. The intent is not to replace current NASA efforts, such as AVIARY and GASPy, but rather to augment mission and sizing when needed and adequately capture higher fidelity physics for SFNP vehicles. For example, classic drag and thrust-based sizing methods may be used for early design, while the proposed higher fidelity mission and sizing analysis can be used as high-fidelity MDA information becomes available.

The primary contribution is a novel dynamics formulation that abstracts away potentially aircraft-specific configuration details such as the propulsor type. Additionally, the GT-ASDL team developed a framework for optimizing and simulating design and off-design missions consisting of typical mission phases with predefined assumptions or constraints. This framework's capabilities were demonstrated and compared to the legacy FLOPS tool using two civil transport aircraft types. Results from these analyses demonstrate that in some cases, fuel burn estimates from the newly developed framework differ from those obtained by the traditional, FLOPS mission-analysis tool.

The content of this report is presented as follows. Chapter 2 presents the technical approach that is the backbone of the developed mission sizing and synthesis code. This includes a description of the general equations of motion used, the implementation of mission sizing and synthesis framework, and the integration of the various disciplinary models. Chapter 3 presents the case studies that demonstrate the capabilities of the developed framework. Specifically, two case studies are considered: a conventional tube and wing vehicle and an unconventional strut-braced wing aircraft with an open rotor propulsion system. The results from these case studies are presented in Chapter 4, where the newly developed framework is compared against a traditional mission analysis tool.

Chapter 2

Technical Approach

2.1 General Equations of Motion

2.1.1 Traditional Equations of Motion: Forces of Flight

The traditional approach to aircraft mission analysis is formulated as an extension of the rigid fixed-wing aircraft performance problem. First, a tailored formulation of the equations of motion is posed for an aircraft in accelerating, climbing, and turning flight, in terms of forces T, D, W on the aircraft of mass m, a velocity magnitude V relative to the static atmosphere, a flight path angle γ (angle of velocity vector relative to the flat earth), and thrust angle of attack ϵ, as given in Eqs. (2.1) - (2.3) [6].

$$T \cos\epsilon - D - W \sin\gamma = m\frac{\mathrm{d}V}{\mathrm{dt}} \tag{2.1}$$

$$L \cos\phi + T \sin\epsilon \cos\phi - W \cos\gamma = m\frac{V^2}{r_1} \tag{2.2}$$

$$L \sin\phi + T \sin\epsilon \sin\phi = m\frac{\left(V \cos\gamma\right)^2}{r_2} \tag{2.3}$$

Within conceptual design studies, it is common to assume that the vehicle conducts a series of steady-state mission phases, wherein the vehicle is that no turning flight is performed ($\phi = 0$), yielding the simplified set of equations given in Eqs. (2.4) and (2.5).

$$T \cos\epsilon - D - W \sin\gamma = m\frac{\mathrm{d}V}{\mathrm{dt}} \tag{2.4}$$

$$L + T \sin\epsilon - W \cos\gamma = m\frac{V^2}{r_1} \tag{2.5}$$

These equations present two challenges which are considered within the present research. First, while these equations are proven to provide accurate estimates of aircraft point performance, their formulation is not amenable to integration within more general trajectory optimization methods. Most importantly, the time derivatives of the states of motion are not explicitly defined for all states within the present formulation. Secondly, the present formulation utilizes a forces-of-flight formulation, presuming that the external forces exerted on the vehicle can be decomposed completely into propulsive forces (i.e., T) and aerodynamic forces (i.e., L and D). While reasonably true for many aircraft configurations, this assumption is poorly suited for concepts exhibiting a high degree of aeropropulsive coupling. Furthermore, these equations exclude the consideration of other external forces that may be applied to the vehicle. To remedy these concerns, a reformulation of the aircraft equations of motion is needed that emphasizes a more generalized form that is amenable to a more diverse expression of aircraft external forces and compatible with modern trajectory optimization approaches.

2.1.2 General Dynamics Equations

Consider the representation given in Figure 2.1, which pictures a three degree-of-freedom system representing the aircraft with its mass concentrated at a point. This system is posed within the aircraft wind axis frame, translating within the $x-z$ plane and rotating about the y-axis. External, non-gravitational (aeropropulsive) forces applied to the system are resolved along the x-axis and z-axis as $F_{x,w,ext}$ and $F_{z,w,ext}$, respectively, and external moments about the y-axis as M_Y.

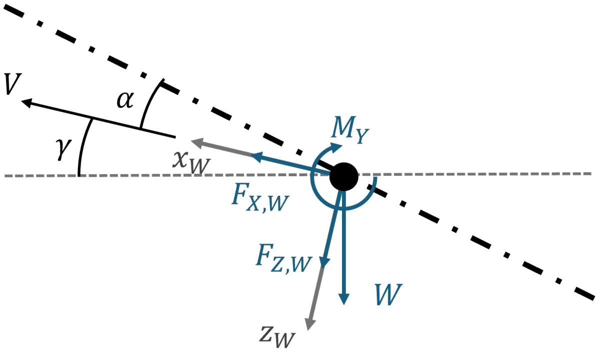

Figure 2.1: Point-mass representation free-body diagram.

The general equations of motion are formulated using Newton's second law, assuming constant mass for each time step. Vectors in the wind axis frame can be expressed with respect to the inertial reference frame using the vector derivative transport theorem. The acceleration within this frame is expressed as Eq. (2.6).

$$\vec{a} = \dot{\vec{V}} + \tilde{\vec{\omega}}\vec{V} \tag{2.6}$$

In Eq. (2.6), the vectors are posed as translational velocity expressed in the wind axes $\vec{V} = [V, 0, 0]^T$ and angular velocity $\vec{\omega} = [p, q, r]^T$. Here and throughout this analysis, the aircraft is assumed to be moving through a static atmosphere, so the relative velocity of the aircraft to the freestream is equal to the velocity of the aircraft relative to the ground. The accelerations in the x and z directions are given as scalar Eqs. (2.7) and (2.8), time derivatives of scalar velocity magnitude V and angular rate q.

$$a_x = \frac{\mathrm{d}V}{\mathrm{d}t} \tag{2.7}$$

$$a_z = -Vq \tag{2.8}$$

Taking the sum of forces along axes parallel to the freestream (*streamwise*) and perpendicular to the freestream (*stream-normal*) results in the equations of motion provided as Eqs. (2.9) and (2.10), noting the relationship between q and γ as $q = \frac{\mathrm{d}\gamma}{\mathrm{d}t}$. This set of translational governing equations — also presented in [7, pp. 23] — relate net aeropropulsive streamwise force $F_{x,w,ext}$ and stream-normal force $F_{z,w,ext}$ to time derivatives of V and γ.

$$F_{x,w,ext} - W \sin\gamma = m\frac{\mathrm{d}V}{\mathrm{d}t} \tag{2.9}$$

$$F_{z,w,ext} + W \cos\gamma = -mV\frac{\mathrm{d}\gamma}{\mathrm{d}t} \tag{2.10}$$

Isolating the time derivatives on one side of each equation, yields the form of these equations as actually coded into the `RigidBody2DEOM` class in its `compute` method:

$$g\left(\frac{F_{x,w,ext}}{W} - \sin(\gamma)\right) = \dot{V} \tag{2.11}$$

$$-\frac{g}{V}\left(\frac{F_{z,w,ext}}{W} + \cos(\gamma)\right) = \dot{\gamma} \tag{2.12}$$

These equations are complemented with the resolution of the moment about the y-axis, taken within the body axis frame, as Eq. (2.13) to form the full three degree-of-freedom equations of motion.

$$M_Y = I_Y \frac{\mathrm{d}q}{\mathrm{d}t} \tag{2.13}$$

Within conceptual design studies, the rotational motion about the y-axis is commonly neglected, assuming that sufficient control authority is available to trim the vehicle in all flight conditions. In such instances, only Eqs. (2.9) and (2.10) are needed to estimate translational motion throughout the mission.

Comparison of Eqs. (2.9) and (2.10) with Eqs. (2.4) and (2.5) reflect a large degree of similarity. Indeed, in instances where external forces acting on the aircraft arise from separable aerodynamic and propulsive effects, then the relationships:

$$F_{x,w,aero} = -D \tag{2.14}$$

$$F_{z,w,aero} = -L \tag{2.15}$$

$$F_{x,w,prop} = T\cos\epsilon \tag{2.16}$$

$$F_{z,w,prop} = -T\sin\epsilon \tag{2.17}$$

provide a mapping of the traditional forces of flight into the more general expression of the equations of motion.

The general equations (2.9) and (2.10), however, allow for more general expressions of external forces, particularly those which may be strongly coupled and difficult to accurately separate between various disciplinary sources. Furthermore, taking the states of motion as V and γ, these equations provide explicit time derivatives for the vehicle motion. These, then, are second-order ODEs in terms of position states for altitude h and horizontal distance s, in terms of general aeropropulsive forces resolved in a wind axes system.

2.1.3 Kinematic Equations

The first time derivative of position is given by the inertial velocity vector in an earth-fixed reference frame $\vec{V}$, which can be expressed in terms of its magnitude V and direction, which is by definition the flight path angle γ. Thus, as Hull [7] also poses, the scalar equations relating the velocity variables V and γ to the first time derivatives of position states h and s are as given in Eqs. (2.18) and (2.19):

$$\frac{ds}{dt} = V\cos\gamma \tag{2.18}$$

$$\frac{dh}{dt} = V\sin\gamma \tag{2.19}$$

Integrating the Eqs. (2.7) and (2.8) yields information on the velocity states, which in turn may be integrated in Eqs. (2.18) and (2.19) to yield the position state information. Thus, Eqs. (2.7), (2.8), (2.18) and (2.19) together provide the governing equations of the dynamics and kinematics of the aircraft model used in the general, physics-based mission analysis methodology presented here.

2.1.4 Weight Equation

The final equation needed to define the system dynamics is one relating the mass of the vehicle to its stored energy state, as represented by the fuel weight of the vehicle. Simply equating the vehicle's weight time derivative to the fuel burn rate suffices here.

$$\dot{W}_F = \frac{dW}{dt} \tag{2.20}$$

Hull [7] goes one step further to explicitly include the thrust dependence by means of a thrust-specific fuel consumption variable. However, in the methodology applied here, the fuel flow rate was directly extracted from the propulsion system model, so thrust-specific fuel consumption was not used. Note that this would not apply to electric-propulsion aircraft. In the case of a non-fuel-burning vehicle, both the weight timeseries state and the fuel burn rate could be omitted from the system of governing ODEs.

2.1.5 Disciplinary Relations

Some variables in the ODEs above depend on other variables, at least implicitly. The aeropropulsive forces $F_{x,w,ext}$ and $F_{z,w,ext}$, for example, most likely depend on V and atmosphere conditions, which also depend on h. They also depend on the aircraft angle of attack α and the engine power setting P, which so far have not been directly included in the governing equations. Therefore, some of these variables may be expressed as functions:

$$F_{x,w,ext} = F_{x,w,ext}(h, V, \alpha, P) \tag{2.21}$$

$$F_{z,w,ext} = F_{z,w,ext}(h, V, \alpha, P) \tag{2.22}$$

$$\dot{W}_F = \dot{W}_F(h, V, \alpha) \tag{2.23}$$

$$\dot{W}_F = \dot{W}_F(h, V, \alpha) \tag{2.24}$$

$$\tag{2.25}$$

Given the full set of ODEs presented above, the *state variables* of position, velocity, and weight are s, h, V, γ, and W. There are then five state variables and five governing ODEs. The two remaining variables α and P are *control variables*, which must be somehow constrained or specified to solve the system [7, pp. 24–25].

In summary, the aircraft's position, velocity, and weight timeseries may be obtained by integrating the equations of motion (and weight) over time across the mission, given timeseries for the control variables and appropriate boundary conditions. This formulation also allows a trajectory optimizer to choose controls to minimize an objective such as mission fuel burn, the motivator for this effort.

2.1.6 Equations of Motion Testing Methodology

The system of equations discussed previously is central to the solution of the trajectory optimization problem. If the accelerations (or state time derivatives) are not correctly related to the state variables, the vehicle dynamics may be incorrect, resulting in very suboptimal or even infeasible trajectories. Therefore, it is worthwhile to invest some thought and effort into testing the implementation of the equations of motion component.

Generally, software testing involves the following components [8]:

1. software entity under test
2. selected inputs to exercise the capability under test
3. results to compare the output with: an "oracle"

In this effort, the software under test is the equations of motion mission analysis component. The inputs to test with are, generally speaking, a trajectory timeseries (vectors of state variables). Outputs of interest from the equations of motion are primarily the accelerations given by the left-hand side of equations (2.7) and (2.8). The method of obtaining "oracle" outputs requires some consideration here.

The naive approach would be to execute, perhaps "by hand" or in a computational environment outside the component under test, the same mathematical relations given in the acceleration equations. Industry best practice holds [9] that this is not a suitable source for test oracles. Rather, a significantly different mechanism or process is preferred to obtain oracles that are unlikely to be tainted by any faults present in the software under test.

Considering this goal, the researcher chose to use straight differentiation on an arbitrarily chosen kinematic trajectory mathematical model to obtain closed-form expressions for the accelerations for that specific trajectory. The following workflow describes this process in general:

1. Identify a kinematic trajectory description of motion
   - This may be a general class of analytical functions, such as a line, a parabola, etc.
2. Determine expressions for the position states as functions of time.

3. Differentiate these expressions to obtain the velocity components as functions of time
4. Differentiate again to obtain acceleration equations ($\dot{V}$ and $\dot{\gamma}$ in this context)
    - This provides oracles for the test case
5. Multiply the accelerations obtained previously by vehicle mass to obtain net applied forces required to cause those accelerations
    - These forces include *all* applied forces, including gravitational weight
6. Use vector arithmetic to determine the force contributions from non-gravitational forces
    - These non-gravitational forces are assumed to be the aeropropulsive forces

Some steps may be skipped or implicitly carried out on certain trajectories. For example, in an unaccelerated trajectory, it is obvious that $\dot{V} = \dot{\gamma} = 0$.

2.1.7 Review of Criteria-Based Testing Theory

This section exists merely to explore theory-based criteria which could have been applied. In actuality, the issue of how to choose trajectories for Step (1) above was resolved without a formal method. This issue may be thought of as one of selecting test inputs, since the trajectory determines the values of all velocity and force inputs to the equations of motion under test. The test designer chose arbitrary trajectories conveniently expressed in closed form and readily differentiated.

Ammann and Offutt [8] labels this approach to choosing test inputs the "human-based" approach, as opposed to a formal, systematic, "criteria-based" approach that defines rigorous criteria from which the tester derives requirements for test cases to write based on some model of the software being tested. Precise criteria and requirements enable precise "coverage" measurements indicating what percentage of the test requirements are covered by the test suite. Ammann and Offutt [8] acknowledges that this human-based approach is generally more manual and requires more domain knowledge of the software application. In this case, the test designer is familiar with the mathematical model and the application domain, from experience maintaining and using the software, which helps address the domain knowledge requirement. The drawback is that there is no transparent, traceable process showing where each test case originated from or why it is required. Rather, intuition from the perspective

At one point in the project, the test designer began sketching out a systematic approach to developing test requirements for this equations of motion component. Several criteria-based software modeling approaches came under consideration, based on prior knowledge from a software testing course based on [8].

- Input space partitioning
- Graph modeling
- Logic modeling
- Textual modeling

Input space partitioning [8] drew particular attention, because of the relative simplicity of the code under test (few logic branches), and therefore a simple execution control flow graph. Textual modeling may have been an alternate candidate choice. Lack of experience in mutation testing — the only criteria-based testing strategy with which the test designer was familiar — in the software language used here rendered textual model-based test design unattractive. Input space partitioning frankly bears the most resemblance to the intuitive human-based testing approach already implemented at this point in the project

2.1.8 Theoretical Example of Input Space Partitioning

The authors of [8] lay out the process of input-space partitioning. For this EoM component, which implements the relations of Eqs˙ (2.7) and (2.8), the this process is enumerated as:

1. Declare the EoM component's `compute` method as the software unit under test
2. Define characteristics of relevant function input parameters V, γ, $F_{x,w,ext}$, $F_{z,w,ext}$, and W
   - Characteristics Q_V, Q_γ, Q_{Fx}, Q_{Fz}, and Q_W: degree of variation of values
   - Characteristics Q_{Fh} and Q_{Fv} of the sum of $F_{x,w,ext}$, $F_{z,w,ext}$, and W: variation of net load
3. Partition entire input space into "blocks" using characteristics from Step (2) above
   - Define "blocks" — (1), (2), and (3) listed below — for each of the seven characteristics above
4. Select a criterion for combining blocks to derive test requirements
5. Define actual values to fill blocks in test requirements
   - Determine trajectories yielding appropriate V, γ, $F_{x,w,ext}$, $F_{z,w,ext}$, and W for test requirements

Note that one could also hypothetically choose `compute_partials` as the code under test in Step (1). Also note that for the test cases implemented here, gravitational acceleration g is held constant. This disqualifies it from the set of variables to be characterized and partitioned with blocks.

The first group of parameter characteristics (Q_V, Q_γ, Q_{Fx}, Q_{Fz}, and Q_W) described in Step 2 comes from considering that the vectorized arithmetic operations in EoM component functionally may exhibit distinct behavior in the cases where inputs are zero versus nonzero (multiplication and division especially). The vectorized nature of the computation motivates cases where some inputs are constant throughout the timeseries and others in which inputs vary across time. All of these cases can be summed up into the three mutually exclusive (pairwise disjoint) blocks for each variable timeseries vector's characteristic of variation relative to zero:

1. all elements of timeseries vector equal zero
2. all elements equal a fixed nonzero value
3. varying across the timeseries

Note that such blocks form a valid partition [8] of the input domain for each of the timeseries inputs V, γ, $F_{x,w,ext}$, $F_{z,w,ext}$, and W, since all possible vectors of real values for each of these falls into exactly one of these three blocks.

Another parameter characteristic ideated for this work is the dimensionality of the acceleration. Such characteristics include the degree of acceleration in one or more dimensions, which is equivalent to the degree of net force including weight in two independent directions. The dimensions are the earth-fixed horizontal and vertical directions, hence characteristics Q_{Fh} and Q_{Fv}. These characteristics do not map onto parameters in a one-to-one manner. However, every selection for input values maps to only one of these blocks. In other words, which blocks are covered is a function of the test inputs, but not vice-versa.

As an example, consider a hypothetical test case that specifies a constant, nonzero velocity; zero flight path angle; linearly decreasing weight; linearly decreasing stream-normal force; and quadratically decreasing streamwise force — how one may model an aircraft in cruise. In this case, the net vertical force including weight is constantly zero, and the net horizontal is varying. Therefore, this case would cover blocks Q_V[2], Q_γ[1], Q_{Fx}[3], Q_{Fz}[3], Q_W[3], Q_{Fh}[2], and Q_{Fv}[1].

For each of the characteristics Q_V, Q_γ, Q_{Fx}, Q_{Fz}, Q_W, Q_{Fh}, and Q_{Fv}, the three blocks partition that dimension of the input space. Since these blocks are pairwise disjoint, a given test case's inputs must fall into exactly one block for each characteristic. In contrast to certain varieties of graph-model-based testing approaches, there must be as many test input cases as test requirements to achieve complete (100%) coverage based on this input space partition.

Given this input space partition, the test designer must next choose a coverage criterion to measure a test suite's progress toward satisfying test requirements. One fairly simple criterion for the ISP method is the All Combinations Criterion (ACoC). As its name suggests, this requires testing all possible combinations of blocks from the various input characteristics. For the EoM component input space partition described previously, the ACoC requires tests indicated in Table 2.1 Note that testing all possible combinations of blocks

Table 2.1: Test Requirements (TRs) for All Combinations Criterion for ISP (2187 total)

	Test Requirements (TRs)																						
Characteristic																							
Q_V	1	1	1	1	1	1	1	1	1	1	1	1	1	...	3	3	3	3	3	3	3	3	3
Q_γ	1	1	1	1	1	1	1	1	1	1	1	1	1	...	3	3	3	3	3	3	3	3	3
Q_{Fx}	1	1	1	1	1	1	1	1	1	1	1	1	1	...	3	3	3	3	3	3	3	3	3
Q_{Fz}	1	1	1	1	1	1	1	1	1	1	1	1	1	...	3	3	3	3	3	3	3	3	3
Q_W	1	1	1	1	1	1	1	1	1	2	2	2	2	...	3	3	3	3	3	3	3	3	3
Q_{Fh}	1	1	1	2	2	2	3	3	3	1	1	1	2	...	1	1	1	2	2	2	3	3	3
Q_{Fv}	1	2	3	1	2	3	1	2	3	1	2	3	1	...	1	2	3	1	2	3	1	2	3

with only three blocks per input parameter, in a fairly straightforward component with only 5 parameters considered here requires $3^7 = 2187$ tests because of the combinatorial nature of this criterion. Also note, not all these blocks represent physically achievable trajectories. Consider those columns shown in gray. These are physically inconsistent. For example, the second column shows zero velocity, weight, and applied forces, yet requires a nonzero, constant net applied force. Though inputs could be supplied that satisfy any arbitrary set of conditions on the inputs independently, the requirement for net force is coupled to the force inputs.

A different criterion leading to a more manageable number of test requirements is the Base Choice Coverage (BCC) criterion [8]. This criterion is best explained by example. Suppose the block (3) is taken as the base case, since it represents the largest share of input values encountered in actual modeling and simulation. The other characteristics are then varied one at a time, holding all but that one at the base case. The result of this process carried out for this ISP appears in Table 2.2.

Table 2.2: Test Requirements (TRs) for Base Case Criterion for ISP (15 total)

	Test Requirements														
Characteristic															
Q_V	**3**	1	2	3	3	3	3	3	3	3	3	3	3	3	3
Q_γ	**3**	3	3	1	2	3	3	3	3	3	3	3	3	3	3
Q_{Fx}	**3**	3	3	3	3	1	2	3	3	3	3	3	3	3	3
Q_{Fz}	**3**	3	3	3	3	3	3	1	2	3	3	3	3	3	3
Q_W	**3**	3	3	3	3	3	3	3	3	1	2	3	3	3	3
Q_{Fh}	**3**	3	3	3	3	3	3	3	3	3	3	1	2	3	3
Q_{Fv}	**3**	3	3	3	3	3	3	3	3	3	3	3	3	1	2

As can be seen from the table, there are only fifteen possible BCC test requirements using this base case, a dramatically smaller number than for the All Combinations Criterion. Again, there are some cases that represent physically invalid trajectories, shown in gray. The one input case shown in bold-face type in Table 2.2 is the "realistic" test case mentioned below. None of the other test requirements shown have corresponding tests implemented.

2.1.9 Equations of Motion Test Cases as Implemented

Distinct from abstractly designing test requirements is the task of implementing automated test cases to exercise the EoM code with concrete inputs. Even though from the standpoint of the EoM component itself the inputs are independent, they should describe a physically valid trajectory. Non-physical trajectory inputs could satisfy test requirements from an ISP criterion described in Section 2.1.8, but computing "oracle" acceleration values $\dot{V}$ and $\dot{\gamma}$ without using the EoM relations themselves would be an issue, since the process outlined above in Section 2.1.6 assumes the trajectory is kinematically and dynamically consistent.

The testing suite implemented for this work includes the following six EoM test trajectory cases:

1. Static, $V = \gamma = F_{x,w,ext} = 0$, $F_{z,w,ext} = -W$, zero-acceleration state: `_NullAcceleration`
2. Constant-velocity, horizontal cruise, $\gamma = 0$, $F_{z,w,ext} = -W$: `_UnacceleratedAcceleration` (a)
3. Constant velocity, steady descent (constant, negative γ): `_UnacceleratedAcceleration` (b)
4. Parabolic, ballistic motion with constant horizontal velocity: `_ParabolicAcceleration`
5. Purely horizontal damping force defined by $F_h = -CV\cos(\gamma)$: `_1dDragAcceleration`
6. Sample of data from a solved trajectory: `_RealisticAcceleration`

These trajectories seemed reasonably straightforward to model, but were intuitively interesting cases to test. In each case, the test suite compares outputs from `compute` to corresponding outputs from the prescribed test-case trajectory and checks the results from `compute_partials` against finite-difference approximations at each point in all trajectories except (1). Each consists of a set of vectors, each vector a series of input or output values. Mapping each test case listed above into the input space partitioning results in Table 2.3.

Table 2.3: Input Space Partitioning of Implemented Test Cases

	Blocks in Test Cases by Number					
Characteristic	Case #1	Case #2	Case #3	Case #4	Case #5	Case #6
Q_V	1	2	2	3	3	**3**
Q_γ	1	1	2	3	3	**3**
Q_{Fx}	1	1	2	1	3	**3**
Q_{Fz}	2	2	2	1	3	**3**
Q_W	2	2	2	2	2	**3**
Q_{Fh}	1	1	1	1	3	**3**
Q_{Fv}	1	1	1	2	2	**3**

The one column in bold is the one also appearing in the ISP BCC test requirements (Table 2.2). The others, of course, appear in the ACoC requirements (Table 2.1), in the large part of the table omitted for brevity.

Coverage of EoM Tests

The ISP method (Section 2.1.8) was not the driver for selecting these cases, but the test suite as implemented can still be evaluated against a coverage criterion. If the ACoC is the chosen criterion, then the coverage metric for this six-case test suite is $6/2178 \approx 0.3\%$, seemingly quite low. This number would be greater if non-physical trajectories were excluded from the total number of test requirements. Generally speaking, very rarely, unless for extremely simple functions, can 100% of possible combinations of blocks be tested, this example suggests. Using the BCC criterion described above with a base case of (3), the test suite scores $1/15 \approx 7\%$. One might choose another base case, such as (1) — all-zero — and see a larger coverage value for the same test suite. Indeed, one may note from the structure of the tests in Table 2.3 that the de-facto base case may be (1).

In applications where sufficiently thorough testing is important and the cost of test design is not negligible, as in this case, such a criterion provides a definite stopping point to avoid unnecessary or redundant testing effort [8]. Still, there is clearly room for improvement to provide a thorough test campaign for this EoM component. Furthermore ISP method suggests which test cases would be worth implementing to improve the test suite's quality.

2.2 Sizing and Synthesis

The Olympus tool largely follows a similar paradigm for sizing as the FLOPS tool. It has been demonstrated in use cases that includes pure off-design mission analysis and sizing. Traditional sizing and synthesis (S&S) implies finding a vehicle takeoff gross weight W_{TO}, wing planform area S, and total sea-level static thrust available T_{SL} to carry enough fuel to complete a given mission subject to point performance requirements, such as climb rate or landing field length [10]. Such performance constraints may be analyzed to determine values for thrust-to-weight ratio T_{SL}/W_{TO} and wing loading W_{TO}/S parameters [10, 7], leaving W_{TO} to be determined. FLOPS and the present approach both deviate slightly from this approach by defining wing loading in terms of the *ramp weight* W_0 rather than takeoff gross weight. Therefore, in this work the wing loading is defined as W_0/S, and similarly, thrust loading as T_{SL}/W_0. Also note the definition here of T_{SL}. Thrust loading is defined in terms of this total thrust (not thrust per engine), at nominal, rated operating conditions. Wells, Horvath, and McCullers [11] refer to "rated" thrust per engine, however, which is used in weight estimation relationships.

The weight is found through the process of mission analysis, estimating the fuel required for an aircraft of given (or guessed) takeoff weight to fly a given range and land with perhaps a specified amount of fuel in reserve. A design mission profile is often defined in terms of mission phases such as climb, cruise, descent, and hold, as illustrated by the generic example in Figure 2.2. Key flight condition requirements are defined throughout the mission. In Figure 2.2 the mission is subdivided into a portion termed the *design mission*, which the aircraft would be intended to actually fly, and a *reserve mission* portion so named because its primary purpose is to provide a margin of safety for a contingency scenario where the aircraft would not be allowed to land at its original destination and would need fuel in reserve to fly to an alternate airport. For purposes of the discussion in this paper, the combination of the design and reserve missions is termed the *sizing mission*.

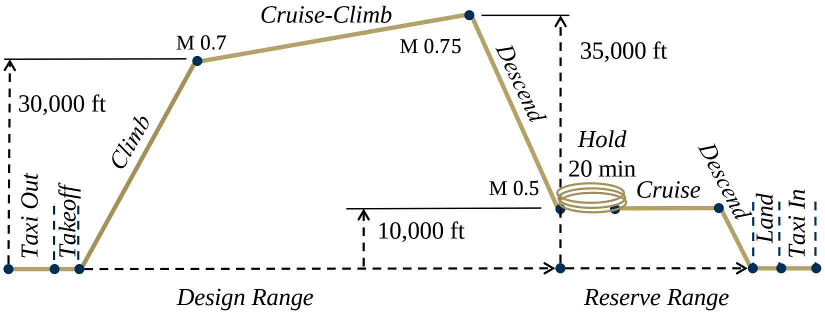

Figure 2.2: Example of a generic mission profile with a design portion and a reserve portion

Mission analysis involves guessing certain characteristics of the vehicle design and preliminary estimates of performance-related characteristics used to evaluate its ability to complete a given design mission, ensuring W_{TO} allows sufficient fuel to be carried. Typically an iterative process, a mission analysis updates its guess for W_{TO} until it converges to a value consistent between the weights disciplinary analysis and the value used for mission analysis, as summarized in [7, pp. 13–14]. Estimating the fuel burn for the mission phases requires a model of the aircraft's performance, so the thrust and therefore fuel flow required to fly the mission can be computed. The equations of motion describing the physics of the aircraft in flight can provide such a model. To compute fuel burn from these equations, one must integrate them over the mission. The exact algorithm and mathematical implementation can differ between tools and depend on the level of detail of the analysis. The following sections discuss this study's approach to accomplishing that.

2.2.1 Legacy Tool Sizing Formulation

The FLight Optimization System (FLOPS) software has a range of available modes for optimizing an aircraft design in tandem with determining an optimal mission trajectory [3]. In FLOPS, W_0 is an implicit output from an iterative mission analysis mode where W_0 is varied to achieve the target range or endurance [11]. This iterative process is distinct from the aircraft configuration optimization. FLOPS optimizes the vehicle configuration design variables, such as W_0 or S, using constraints on the wing loading and thrust-to-weight ratios. This study did not include sizing and synthesis in this sense when using FLOPS. Researchers used FLOPS in this study to optimize a composite mission analysis objective with respect to mission trajectory variables and aircraft gross weight, subject to mission constraints such as cruise Mach number. However, FLOPS was not permitted to vary the wing area or thrust, which were held at values previously determined to be optimal for the respective aircraft configuration.

2.3 Trajectory Optimization

As Hull [7, pp. 10,24–26] discusses, the equations of motion constitute a mathematical system of equations with a certain number of state and control variables. The equations of motion must be integrated to obtain the mission state timeseries. This integration may be done explicitly, such as in a shooting method, or implicitly, such as using a collocation method. The Dymos optimal control library [12] is able to do both, but collocation was chosen for this problem. The number of time-dependent state variables (e.g., speed, altitude, etc.) minus the number of governing equations gives the number of control variables [7]. So, a system defined by seven timeseries states with five governing equations has two independent control variables, or degrees of freedom which can be constrained by defining kinematic trajectory relations (fixed altitude, climb rate, or velocity for example) or performing an optimization with respect to those controls, to minimize the trajectory's total fuel burn for instance. The latter option is suitable for this study, since the goal is to analyze next-generation, fuel-efficient aircraft.

2.3.1 Legacy Tool Trajectory Optimization

As discussed in [13], the FLOPS mission analysis method uses a sub-optimization routine to find an optimal mission trajectory for climb and descent phases. It discretizes the mission trajectory design space for a vehicle in terms of energy height and weight, computing the corresponding objective function value at each of these points, with respect to the altitude, propulsion system control variables, and storing this grid of sub-optimized points as a lookup table to be queried as it steps through the trajectory. To find the best cruise conditions, it assumes steady flight with equilibrium of thrust, lift, drag, and weight, then steps through discrete vehicle weights as fuel is burned to update the optimal cruise condition. Depending on the constraints, the best altitude or velocity may be chosen.

FLOPS uses the DFP or BFGS optimization algorithm [3]. Minimum fuel burn is selected as the optimization objective for all climb mission phases and specific range is the objective for cruise for this study. Maximum lift-to-drag ratio is chosen as the descent objective. FLOPS provides other options for each mission phase as shown in Table 2.4, based on [3].

Table 2.4: FLOPS Mission Optimization Objective Options [3]

Phase	Objective Options
Climb	min fuel, min time, min time over distance, min fuel over distance
Cruise	max endurance, max specific range, max Mach
Descent	max lift-to-drag ratio L/D

.

2.3.2 Kinetic Trajectory Optimization

In the approach proposed here, two of the seven total mathematical degrees of freedom are available as control variables for the equations of motion [7]. Prescribing timeseries for these controls and imposing boundary conditions is sufficient (theoretically) to numerically solve the system of ODEs and obtain timeseries for the state variables throughout the mission. The method here formulates an optimization problem to choose these controls to minimize an objective such as total mission fuel burn. It uses path constraints c_p and boundary constraints c_{bc} in some cases to enforce limits on the solution variables. There may be aircraft configuration design variables x, with respective bounds x_{min} and x_{max}. The implicit collocation integration scheme implemented in the Dymos optimal control library [12] enables simultaneously optimizing the trajectory for fuel burn and solving the model to obtain physically consistent state timeseries, using special constraints called *defects* c_d to enforce the governing equations of the physics. This architecture formulates a large nonlinear programming problem, provides gradient information, and presents it to be solved by an optimizer such as the IPOPT tool used in this effort [14].

One may formally state the optimization problem as follows:

$$\begin{aligned}
\text{Minimize: } & W_b \\
\text{w.r.t.: } & \vec{x}, \vec{u}, \vec{y} \\
\text{s.t.: } & \vec{c}_d = 0, \\
& \vec{c}_p \leq 0, \\
& \vec{h}_{bc} = 0, \\
& \vec{x}_{min} - \vec{x} \leq 0,\ \vec{x} - \vec{x}_{max} \leq 0 \\
& \vec{u}_{min} - \vec{u} \leq 0,\ \vec{u} - \vec{u}_{max} \leq 0 \\
& \vec{y}_{min} - \vec{y} \leq 0,\ \vec{y} - \vec{y}_{max} \leq 0
\end{aligned}$$

However, unlike may be surmised from this formal statement, most design variables x_i, u_i, and y_i have inherent lower and upper bounds directly enforced by preventing the optimizer from choosing values outside those ranges.

2.4 Disciplinary Analyses

To perform a mission analysis, vehicle data on weights, aerodynamics, and propulsion performance are required. Distinct analysis components, embedded within the ODE, provide these data at discrete time steps as the optimizer integrates the ODE over the mission. There are several options for several disciplines, especially weights and aerodynamics. Also, lines between disciplines may be blurred by components that provide data traditionally provided by separate components, namely aerodynamics and propulsion. In coupled aeropropulsive analysis, distinct thrust and drag outputs are replaced by mixed aeropropulsive forces from an integrated analysis component. Generally speaking, all disciplinary analyses within the ODE take vehicle design parameters, time-varying state variables, and certain control variables as inputs, returning other time-dependent variables (forces for example) required for the equations of motion. An illustration of this for a conventional vehicle model using propulsion table lookup (Section 2.4.5), FLOPS-based "external" aerodynamics modeling (Section 2.4.4), and a separable aeropropulsive force interface component (Section 2.4.6 is given in Figure 2.3.

2.4.1 Weights

To size a vehicle, the designer chooses a gross takeoff weight or ramp weight that satisfies the mission range requirement as determined via mission analysis. The adequately sized vehicle has sufficient fuel capacity (represented as a fuel weight allowance) while also carrying the required payload (passengers, their baggage, or any cargo) and other miscellaneous items for standard operation (e.g., safety equipment). To evaluate the weight allowance for fuel in order to complete the mission analysis phase of design, the analyst must estimate the empty aircraft weight [6]. This may take the form of a rudimentary empirical correlation for overall empty weight, based on historical trends [6]. Or, for a more detailed (still preliminary design), it may be based on a component-wise weight build-up, such as the area-based one provided by Raymer [15] as cited in [8]. Note that such an empty weight estimation method requires a guess of the overall gross weight. In the present work, this gross weight guess is supplied by the optimizer for the aircraft/trajectory optimization problem (2.2).

The FLOPS tool includes detailed weight estimation relations derived from empirical data on a representative sampling of mid-to-late 20th century aircraft [3, 11]. These equations provide relations for bare

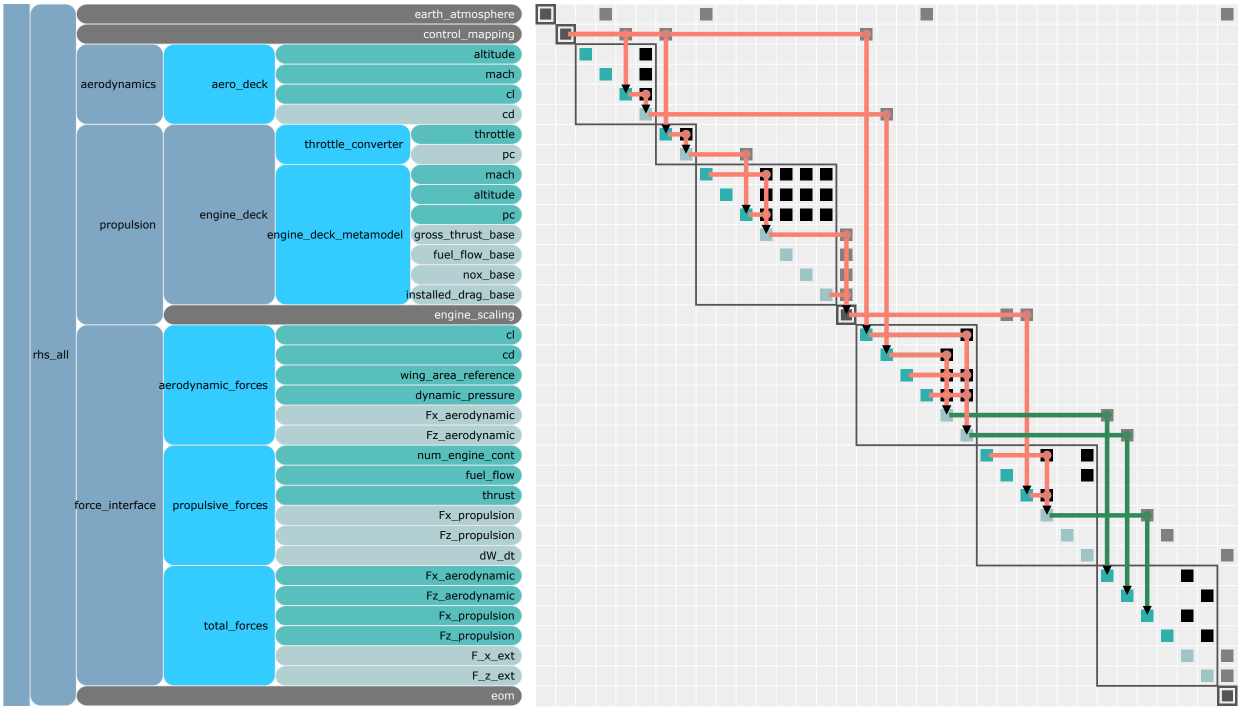

Figure 2.3: N-Squared Diagram of Disciplinary Analyses for Conventional Vehicle

structural, propulsion, crew, baggage, and equipment weight categories. Researchers in this present effort relied on (re-implementations) of these equations. For the most part, these re-implementations were validated against the FLOPS implementations by straightforward comparison of output results for common inputs. Exceptions to the general agreement of these results included wing structural weight in the detailed wing weight estimation method, horizontal and vertical tail weights, and the landing gear weight. The outcome of the structural weight discrepancies between these implementations is that the re-implemented weight module estimates the vehicle's zero fuel weight at 160 lb higher than FLOPS in the case of a notional 150-passenger transport model used for validation testing.

2.4.2 Geometry

For purposes of weights estimation, some key vehicle geometry parameters are required. Some of these parameters, such as fuselage planform area or wingspan, have relations based on empirical [11] or fundamental design relationships. Others are used for aerodynamic drag estimation. In particular, the empirical aerodynamic drag estimation method originally implemented in the kinetic mission analysis framework relied on wetted areas computed by formulae based on the FLOPS routines.

2.4.3 Earth Atmosphere

In order to evaluate the time-dependent aerodynamic and propulsive variables, intermediate variables such as vehicle Mach number M_∞ and dynamic pressure q, the ODE included a standard atmosphere model based on publicly available data on the NASA 1976 standard atmosphere [16], as made available for retrieval via a third-party web application [17] in standard Imperial units. The implementation used an OpenMDAO `MetaModelStructuredComp` trained on data retrieved from the web application's table feature in altitude increments of 1000 ft ASL from −1000 ft until 100,000 ft, then in 5,000 ft increments up to 200,000 ft.

2.4.4 Aerodynamics

For estimating the aerodynamic forces on the vehicle in flight, mission analysis requires an aerodynamics model integrated into the ODE being integrated over time. Initially, in the early stages of this framework's development, designers selected the Empirical Drag Estimation Technique [18] as adapted and augmented in FLOPS [3]. However, this method proved cumbersome and never demonstrated acceptable performance in a mission analysis optimization. Therefore, the framework developers implemented alternative drag estimation routines, such as the table interpolation-based method discussed in more detail in Section 2.4.4.

Empirical Drag Estimation Technique

The Empirical Drag Estimation Technique, or EDET as it was known, is based on a theoretical work from 1978 [18], which led to a software implementation by the same name. However, the researchers could find no publicly extant copy of the standalone software, or its detailed documentation. Since the recent releases of FLOPS have prohibitions on public disclosure of technical information not otherwise published, this report does not go into detail on the EDET routine.

In any case, the researchers chose not to use this method for the analyses presented here. The relatively recent release of the Aviary aircraft design and analysis tool [4] includes its own re-implementations of EDET. The interested reader may refer to its source code and documentation for details.

Externally-defined (FLOPS-based) Aerodynamics Table

The aerodynamics module class named `FlopsExternalAero` reads aerodynamic data from an external text file and builds an OpenMDAO `SemiStructuredMetaModel` component to model the drag coefficient as a function of flight condition. This model takes inputs such as Mach number, altitude, and lift coefficient to compute a drag coefficient for the aircraft. The aerodynamic data used to train the surrogate model must be organized in a semi-structured format, meaning that Mach, altitude, and lift coefficient domains are discretized to form a grid, the only irregularity being the non-uniform spacing in the Mach dimension. A third-order Lagrange polynomial is used to interpolate between data points.

2.4.5 Propulsion

The detailed mission analysis presented here involves propulsion system performance evaluation, so it requires a model of an aircraft propulsion system. The propulsion model for this analysis follows the approach of FLOPS [3] and Aviary [4]: interpolating on an externally-supplied propulsion data table, called an "engine deck," of steady-state engine performance obtained from a thermodynamic engine cycle analysis using a tool such as NASA's NPSS [19]. The interpolant primarily used for this was linear, though a Lagrange interpolation polynomial [20, (3.3.1)] is another option. As can be seen from Figure 2.3, the propulsion block takes as inputs the flight condition (Mach number M_∞, altitude h), and the engine power setting parameter (power code or `pc`) herein represented as P. The outputs from the basic engine deck model (`engine_deck_metamodel` in the diagram) are unscaled thrust, fuel flow, ram drag, and nitrous oxide (NOx) emissions, as in Aviary [4, 21]. The Mach number and altitude determine operational flight regimes, while the throttle setting dimension gives a control degree of freedom. Given these inputs, the propulsion interpolating model returns the engine performance parameters at that point in the mission.

The mission analysis methodology thus abstracts away the thermodynamics via table of performance data called an "engine deck" — as Aviary [4] and FLOPS in its original conception [3] at least did as well. Such a data-driven approach is suitable at the level of conceptual or early preliminary design. However, this type of detailed performance data must be generated by computational simulation or thermodynamic analysis, since engine manufacturers typically do not freely disclose detailed engine operating schedules such as this. Also, if the engine design is to be scaled up or down to size the engine for some air mass flow to meet a thrust requirement such as for takeoff [22], this properly requires cycle analysis to get accurate results. The present method includes a scaling capability, mimicking FLOPS' engine scaling [23], which scales the engine deck outputs such as thrust and ram drag by a ratio of the scaled rated thrust to the baseline or reference thrust [24]. Still, at least a single cycle analysis is needed to generate a performance data table for mission analysis.

One innovation from ASDL is a process to synthesize an engine thermodynamic cycle design according to a set of design points [25] rather than a single point design. Prior to this project, such a process was used to design and analyze an engine cycle for appropriate performance requirements. The NPSS [19] and CMPGEN [26] tools provided environments for engine cycle analysis and fan/compressor maps, respectively. The engine weight was estimated using the WATE++ tool [27], the latter of which ASDL has previously used as an improvised two-dimensional flowpath geometrical definition tool [28]. To calibrate the engine model to manufacturer data, engine performance is estimated by cycle analyses at prescribed design points, tuning certain cycle parameters until the model matches published performance data[5].

This project simply took the results from that propulsion cycle analysis and calibration process and modeled them across the operating domain of the vehicle using OpenMDAO `SemiStructuredMetaModel` components. The domain of the Mach-altitude space sampled for this project resembled something like that shown in Figure 2.4. Note that for each (Mach, altitude) point on the visualization, there are several throttle settings (as if a third dimension out of the page).

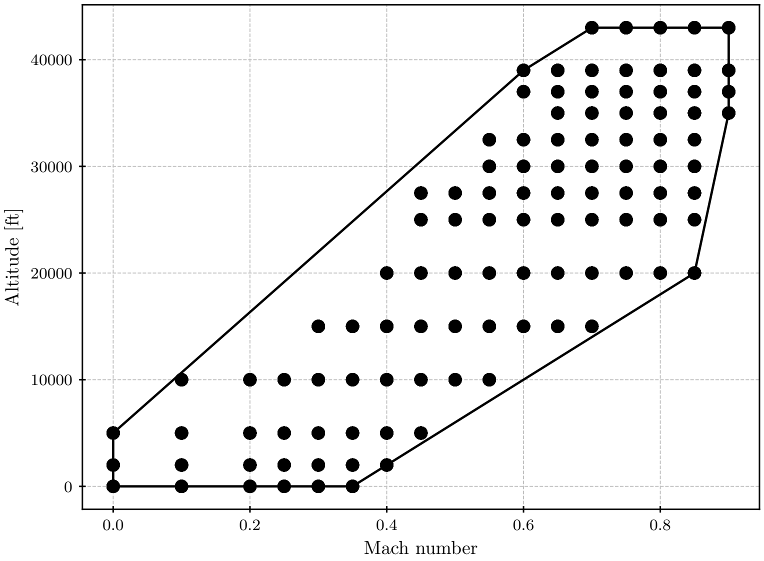

Figure 2.4: SBW engine deck ConvexHull visualization

2.4.6 Aero-Propulsive Modeling

The kinetic framework presented here has at its core the force-agnostic general equations of motion (2.9) and (2.10) presented in Section 2.1. These equations take as inputs the streamwise $F_{x,w,ext}$ and stream-normal $F_{z,w,ext}$ aeropropulsive forces. To compute these forces at arbitrary flight conditions, the mission analysis requires aerodynamics and propulsion models. It also requires an interface to map the aerodynamic and propulsive forces to discipline-agnostic forces $F_{x,w,ext}$ and $F_{z,w,ext}$. Different vehicle models can use different interfaces to handle different disciplinary forces. The following sections discuss such interfaces.

Separable and Uncoupled Forces Interface

Aerodynamics is provided by the external FLOPS-based data table interpolation method (2.4.4) or by an aerodynamic drag coefficient surrogate model based on computational fluid dynamics simulation data. In the traditional, uncoupled aerodynamics case, the output from the aerodynamics code is the drag coefficient for the aircraft C_D, computed as a function of variables such as the lift coefficient C_L. Given this coefficient and the reference planform area S, the code can quickly compute the drag force at any flight condition and corresponding dynamic pressure q using the relation $D = qC_DS$ [6]. In the case of the simple forces of flight bookkeeping scheme, this drag force can be directly related to the streamwise aero-propulsive force as in Eq. (2.14). Since drag by definition is parallel to the velocity vector of the aircraft, it does not have a stream-normal component.

Lift, computed as $L = qC_LS$, is defined in the stream-normal direction and therefore parallel to the wind axes' z direction, hence the straightforward relation in (2.15) which has a negative sign to account for the fact that the lift is traditionally defined in the upward direction (if the airplane is oriented right-side-up) — in the direction corresponding to the negative z-direction in the wind axes.

The "forces of flight" case includes a thrust component in the streamwise direction (2.16). This becomes a simple relation

$$F_{z,w,prop} = T \tag{2.26}$$

If the angle between the thrust and the freestream is assumed to be zero ($\epsilon = 0$), as it is for the analysis presented later in Section 4.1. Though in general the vehicle can have propulsive force contributions in the stream-normal direction due to a nonzero angle between the thrust centerline and the airstream (Eq. (2.17)), assuming $\epsilon = 0$ eliminates this component, yielding

$$F_{z,w,prop} = 0$$

which implies the total aero-propulsive force from these contributions can be seen to be the simple algebraic sum of the respective force components from these disciplines, as shown in Eqs. (2.27) and (2.28).

$$F_{x,w,ext} = F_{x,w,prop} + F_{x,w,aero} \tag{2.27}$$

$$F_{z,w,ext} = F_{z,w,prop} + F_{z,w,aero} \tag{2.28}$$

In terms of the specific aero and propulsive forces in this formulation, the aero-propulsive force components are

$$F_{x,w,ext} = T - D \tag{2.29}$$

$$F_{z,w,ext} = 0 - L = -L \tag{2.30}$$

assuming the thrust angle of attack $\epsilon = 0$. This force interface formulation is the translation layer between the conventional, tube-and-wing vehicle's (Section 3.2) aerodynamics and propulsive models and the equations of motion.

Coupled Aeropropulsive Force Interface

In the more general case of a vehicle that experiences aerodynamic and propulsion-related forces in both the streamwise and stream-normal directions, the thrust angle of attack ϵ may be nonzero, perhaps even time-varying in the case of a vectored-thrust configuration. This would imply $F_{z,w,prop} \neq 0$. In such a case, this contribution of the thrust to net stream-normal aeropropulsive force $F_{x,w,ext}$ may be separable. However, if the aerodynamic forces on a lifting surface of the aircraft are influenced by flow turning as a consequence of, for example, installing engines above instead of below the wing of an aircraft [29], then it is less clear exactly how much the propulsion system or the wing separately contribute toward the stream-normal forces, and the distinction between $F_{z,w,aero}$ and $F_{z,w,prop}$, becomes less meaningful. In such a case, it may not be possible to straightforwardly apply Eqs. (2.27) and (2.28) or compute the force components (Eqs. (2.15) and (2.17)) separately, since the only accurate modeling of such complex aeropropulsive interaction may be a CFD simulation of the integrated airframe-propulsion system.

In such a case, the only accurate model of the aero-propulsive loads on the vehicle may be obtained at the same level of abstraction as Eqs. (2.9) and (2.10). This was the case in the over-wing nacelle study [29]. A similar approach was useful for modeling the aeropropulsive forces of the strut-braced-wing novel aircraft configuration (See Figure 3.2), anticipating interactions between the propwash from the propulsion system and the lifting surface(s). In these cases, the aero-propulsive loads may be expressed directly in terms of similar variables as traditionally used for drag or lift, but this time accounting for the forces resulting from the propulsion system and not purely "aerodynamic" interactions with the airframe. Forces may be obtained by CFD analysis, non-dimensionalized by freestream dynamic pressure and reference area, and that data used to train machine learning models giving these non-dimensional force coefficients as functions of convenient flight condition variables such as Mach number, Reynolds number, and angle of attack, plus an engine power setting variable, as shown in Eqs. (2.31) and (2.32).

$$F_{x,w,ext} = qSC_{Fx}(\alpha, M_\infty, Re, P) \tag{2.31}$$

$$F_{z,w,ext} = qSC_{Fz}(\alpha, M_\infty, Re, P) \tag{2.32}$$

This force formulation only requires a thin interface between the non-dimensionalized and dimensional aeropropulsive forces. The data flow through the pertinent components of such a model is illustrated the N-squared diagram in Figure 2.5.

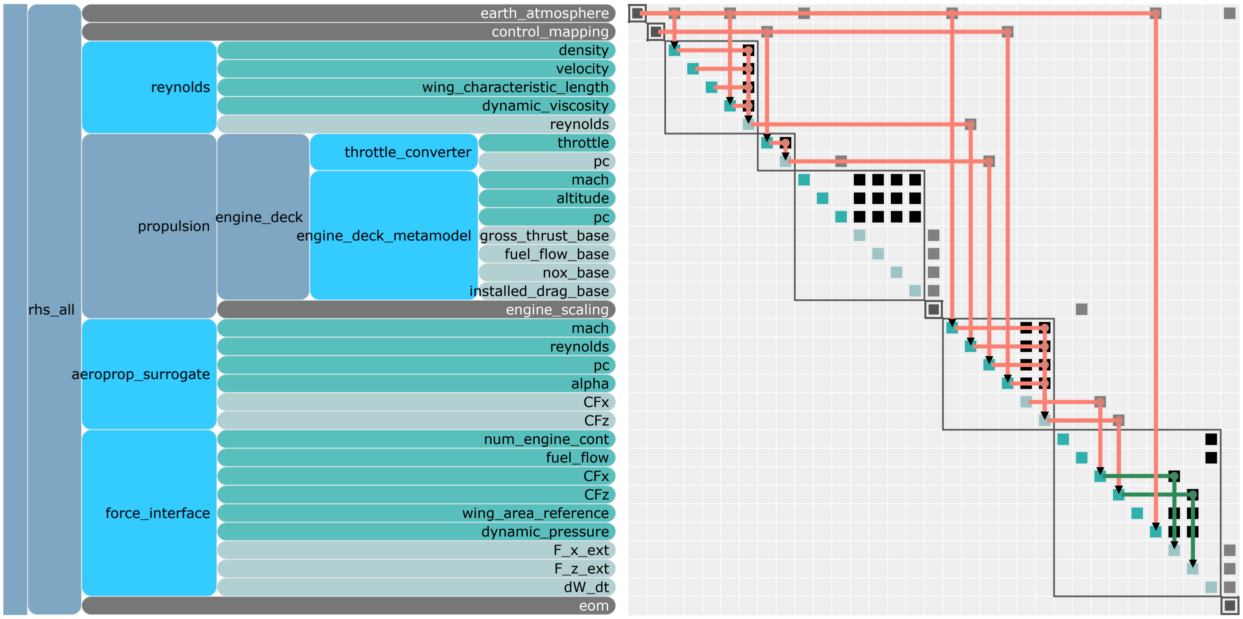

Figure 2.5: Fully-coupled aeropropulsive force interface as integrated with ODE of strut-braced-wing model

Chapter 3

Case Studies

3.1 Overview of Cases

The purpose of this effort was to develop a sizing and synthesis approach with a mission analysis capability more applicable to novel vehicle models and aeropropulsive coupling than legacy analysis tools. To that end, the following presents two case studies, each aiming to demonstrate slightly different aspects of the method. First, a conventional tube-and-wing aircraft model analysis benchmarks the proposed kinetic analysis against the legacy tool. Second, a novel vehicle of interest to the Sustainable Flight National Partnership [30] is analyzed as a case study in how the mission analysis tool compares to its legacy counterpart for a concept which may involve more coupled aerodynamics and propulsion.

The reader should note that the aerodynamics and propulsion data used for this analysis were obtained from disciplinary models (the FLOPS adaptation of the Empirical Drag Estimation Technique capability in the case of aerodynamics for the conventional vehicle) depend on the vehicle configuration parameters and therefore do not provide the most accurate performance models as the vehicle size changes as it would in a full sizing exercise.

3.2 Conventional Vehicle

In order to compare the sizing capability of the kinetic method against the legacy benchmark, this section presents a sizing experiment on an existing production vehicle modeled in each of the tools. This vehicle represents a configuration which the legacy code is trusted and calibrated to model accurately. It explores the capability of the kinetic approach on a conventional vehicle that has nominal aeropropulsive coupling. For the experiment, the tools chosen to compare against the kinetic formulation were FLOPS [3] and the recently released Aviary tool [4].

The candidate vehicle chosen for this experiment is Airbus SAS' A320neo mid-size (150-passenger) commercial air transport. In addition to the access to vehicle data, this aircraft was a logical choice as it represents a large portion of typical mid-haul commercial aircraft operations in the United States. The following section discusses how the model for this aircraft is developed.

3.2.1 Calibration against Data Sources

To accurately capture the conventional vehicle's mission performance and sizing behavior, this aircraft model is based as much as practicable on actual aircraft specifications, including overall parameters such as wing span, rated thrust, cruise Mach number, maximum takeoff weight (MTOW), range-versus-payload, plus details such as empennage geometrical characteristics, wing sweep angle, wing planform area, landing gear height, number of passengers typically carried, and fuel tank capacity. In-house processes exist in ASDL to calibrate legacy airframe and engine models in the Environmental Design Space (EDS) [5], matching results from propulsion cycle design to key performance requirements [25] in conjunction with matching aircraft performance constraints and mission range. Much of the data used to develop the models used for benchmarking is based on the data collected by the lab from Airbus's Airport Planning Manual (APM) [31], CAD drawings, the ICAO Aircraft Emissions databank [32], and calculations based on the data obtained. The specifications of the A320neo vehicle chosen for this research are included in the appendices in Table 1. The geometric data obtained from measurements taken from CAD drawings and from the APM are recorded in the appendix in Tables 2– 9. A prior research team used these data and modeled the vehicle in FLOPS/EDS. They calibrated this airframe model in terms of payload, range, and gross weight using aircraft specifications such as payload-range diagrams and the typical passenger cabin arrangement.

Referring to this FLOPS/EDS model as "Stage 0", the team initially constructed a "Stage 1" model in the kinetic analysis framework based almost entirely on data in the FLOPS analysis input/output files. These FLOPS files were the starting point for developing the Aviary analysis input files as well. In "Stage 1," the aim was to demonstrate the kinetic tool's ability to closely replicate FLOPS mission analysis results using similar assumptions, a fixed mission profile following the trajectory FLOPS selected, using the same aircraft configuration data.

Propulsion Modeling

To accurately capture the aircraft performance for mission analysis, propulsion system performance, especially relating thrust and fuel flow rate, is necessary. Previous research teams had conducted propulsion cycle analysis and engine model calibration for the Pratt & Whitney PW1127G-JM engine powering the A320neo aircraft. Data for this engine is in Table 10. This calibration process entails specifying a set of five mission point-performance requirements which the designed propulsion cycle must satisfy [5]. CMPGEN [26], NPSS [19], and WATE++ [27] codes together form an engine design loop [5] converging engine cycle design parameters to match engine performance as published in the ICAO Emissions Databank [32], subject to manufacturer-published engine parameters like fan diameter and number of compressor and turbine stages. Resulting modeled performance data is in Table 11. The final outputs from this engine design/calibration loop are engine characteristics which result in an engine design matching published performance data. Steady-state engine performance is captured in the form of an "engine deck" data table as previously described. These outputs then feed into aircraft-level weight and mission analysis.

3.2.2 Mission Requirements

This vehicle's mission profile, shown in Figure 3.1 is a 3,420 nmi design mission, consisting of a climb, step-cruise, and descent, followed by a 200 nmi reserve mission to ensure the design carries sufficient fuel to reach an alternate airport. The FLOPS mission definition also included a short cruise segment at 10,000 ft between initial climb and climb to cruise altitude above 30,000 ft. To avoid perturbing the calibrated FLOPS vehicle model, the researchers left this segment in the profile for that tool alone. Also of note, the FLOPS mission permitted a sequence of "steps" in the cruise phase, so a sequence of short climbs interspersed with steady, level cruise phases.

Assumptions

The following general assumptions applied to the analysis of the conventional transport aircraft modeled here:

- Seating configuration: 18 first-class and 132 economy-class (Table 1)
- Initial weight in climb = 99.4105% of ramp weight
- Based on FLOPS climbout weight ratio, skipping taxi-out
- FAA speed limits enforced on FLOPS mission
- not exceeding 250 kn below 10,000 ft., at
- This speed limit was only enforced for the initial climb and descent in Olympus
- for FLOPS, 31 steps in climb segments

Table 3.1: FLOPS Mission Definition for Conventional Vehicle

Phase	Objective	Constraints
Climb 1	Fuel W_f (minimize)	$h \leq 10{,}000\,\text{ft}$, $0.3 \leq M_\infty$, $V_{IAS} \leq 250\,\text{kn}$
Cruise 1	Fuel W_f (minimize)	$1{,}000\,\text{ft} \leq h \leq 10{,}000{,}\text{ft}$, $0.3 \leq M_\infty \leq 0.4$, $s_{phase} = 10\,\text{nmi}$
Climb 2 (Reserve)	Fuel W_f (minimize)	$h \leq 33{,}000\,\text{ft}$, $0.3 \leq M_\infty$
Step Cruise (Reserve)	Fuel W_f (minimize)	$1{,}000 \leq h \leq 39{,}000$, $0.78 \leq M_\infty \leq 0.78$
Descent (Reserve)	L/D (maximize)	$0.3 \leq M_\infty$, $h \leq 10{,}000\,\text{ft}$, $C_{L,des} = 0.8$, $V_{IAS} \leq 250\,\text{kn}$ below 10,000 ft
Climb 3 (Reserve)	Fuel W_f (minimize)	$h \leq 25{,}000\,\text{ft}$, $0.3 \leq M_\infty$, $V_{IAS} \leq 250\,\text{kn}$ below 10,000 ft
Cruise 3 (Reserve)	Fuel W_f (minimize)	$1{,}000\,\text{ft} \leq h \leq 15{,}000\,\text{ft}$, $0.4 \leq M_\infty \leq 0.78$
Hold	Fuel (Reserve) W_f (minimize)	$1{,}000\,\text{ft} \leq h \leq 1{,}500\,\text{ft}$, $0.4 \leq M_\infty \leq 0.5$
End		$h = 0\,\text{ft}$, $M_\infty = 0.3$
Overall	Mixed objective	$R = 3420\,\text{nmi} + 200\,\text{nmi}$ (design + reserve)

- For FLOPS analysis, performance is constrained as follows
  - Service ceiling: positive climb at $h \geq 35{,}000\,\text{ft}$ at Mach $= 0.8$
  - Rate of climb: $\dot{h} \geq 250\,\text{ft/min}$ at $h = 35{,}000\,\text{ft}$, Mach $= 0.8$
- Thrust angle relative to freestream is negligible: $\epsilon = 0$ for Eqs. (2.1)–(2.3)

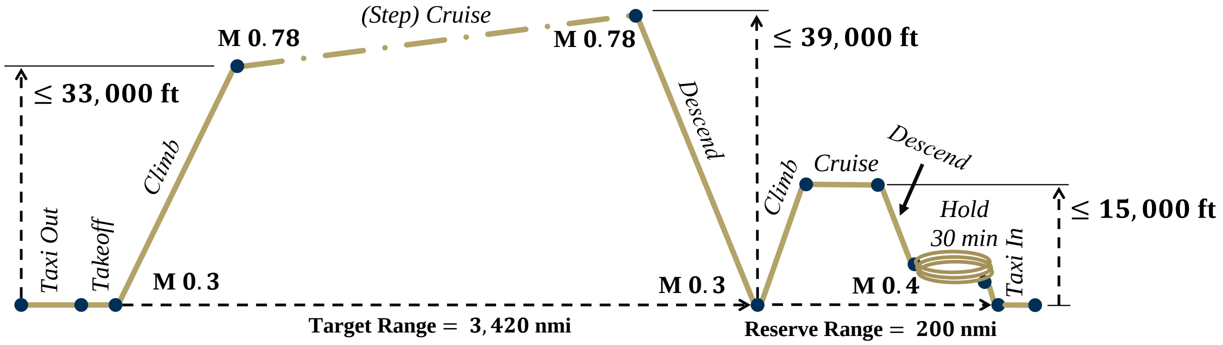

Figure 3.1: Design and reserve nominal mission profiles of the conventional tube-and-wing transport

The FLOPS trajectory optimization objectives used here are listed in Table 3.13.1. FLOPS mission analysis here uses a step cruise at discrete altitudes in the range shown, similar to that implementation in [33]. It also includes constraints applied to the trajectory optimization of each mission phase.

The kinetic mission formulation presented here assumes that whatever thrust is output by the propulsion module is aligned with the velocity vector of the aircraft, so $\epsilon = 0$ in Equations (2.16) and (2.17). Further, phase-specific assumptions appear in Table 3.2. Note the objective for all phases of the kinetic analysis missions identically to minimize overall mission fuel burn. Unlike FLOPS, the kinetic analysis does not support phase-specific objectives. In addition to the phase-wise constraints in Table 3.2, all phases up to and including "FAA Descent 1" had distance constraints: $s \leq 3{,}420$ nmi. Also, FAA Descent 1 itself also had a minimum: $1\,\text{nmi} \leq s$. The reserve mission phases had upper distance bounds equal to the total sizing range: $s \leq 3620\,\text{nmi}$, except the hold segment which does not track distance. The reserve cruise phase also lower-bounded distance to the design range:$s \geq 3420\,\text{nmi}$.

[10] These Mach constraints were actually implemented as velocity constraints, with Mach number converted to speed in ft/s at the respective min/max altitudes on the mission phases.

Table 3.2: Kinetic Analysis Mission Definition for Conventional Vehicle

Phase	Objective	Constraints
FAA Climb 1	Fuel W_f (minimize)	$1\,\text{min} \le t \le 30\,\text{min}$, $h \le 10{,}000\,\text{ft}$, $0.3 \le M_\infty \le 0.8$[1], $V_{IAS} \le 250\,\text{kn}$, $0 \le \dot{h}$, $0 \le \dot{V}$, $0 \le \gamma \le 4^\circ$, $s \le 3{,}420\,\text{nmi}$
Climb 1	Fuel W_f (minimize)	$1\,\text{min} \le t \le 45\,\text{min}$, $10{,}000\,\text{ft} \le h \le 33{,}500\,\text{ft}$, $0\,\text{ft/s} \le V \le 1{,}000\,\text{ft/s}$, $0.3 \le M_\infty \le 0.85$[2], $0 \le \dot{h}$, $0 \le \dot{V}$, $-5^\circ \le \gamma \le 5^\circ$, $s \le 3{,}420\,\text{nmi}$
Cruise 1	Fuel W_f (minimize)	$100\,\text{min} \le t \le 1{,}000\,\text{min}$, $32{,}500\,\text{ft} \le h \le 35{,}000\,\text{ft}$, $0.65 \le M_\infty \le 0.78$[3], $-1^\circ \le \gamma \le 1^\circ$, $-1 \le \text{ ft/s } \dot{h} \le 1\,\text{ft/s}$, $-0.1\,\text{ft/s}^2 \le \dot{V} \le 0.1\,\text{ft/s}^2$, $s \le 3{,}420\,\text{nmi}$
Descent 1	Fuel W_f (minimize)	$1\,\text{min} \le t \le 30\,\text{min}$, $h \le 35{,}000\,\text{ft}$, $0.3 \le M_\infty \le 0.78$[4], $-10^\circ \le \gamma \le 0^\circ$, $s \le 3{,}420\,\text{nmi}$
FAA Descent 1	Fuel W_f (minimize)	$1\,\text{min} \le t \le 1{,}000\,\text{min}$, $h \le 10{,}000\,\text{ft}$, $0.3 \le M_\infty$[5], $V \le 421\,\text{ft/s}$, $V_{IAS} \le 250\,\text{kn}$, $-10^\circ \le \gamma \le 0^\circ$, $s \le 3{,}420\,\text{nmi}$
Climb 2 (Reserve)	Fuel W_f (minimize)	$3{,}420\,\text{nmi} \le s \le 3{,}620\,\text{nmi}$, $1\,\text{min} \le t \le 1{,}000\,\text{min}$, $h \le 1{,}500\,\text{ft}$, $0\,\text{ft/s} \le V \le 1{,}000\,\text{ft/s}$, $0.3 \le M_\infty \le 0.8$[6], $0 \le \dot{h}$, $0 \le \dot{V}$, $-5^\circ \le \gamma \le 5^\circ$, $3{,}420\,\text{nmi} \le s \le 3{,}620\,\text{nmi}$
Cruise 2 (Reserve)	Fuel W_f (minimize)	$1\,\text{min} \le t \le 1{,}000$, $s \le 3{,}620\,\text{nmi}$, $14{,}950\,\text{ft} \le h \le 15{,}000\,\text{ft}$, $0.4 \le M_\infty \le 0.78$[7], $-1^\circ \le \gamma \le 1^\circ$, $-1 \le \text{ ft/s } \dot{h} \le 1\,\text{ft/s}$, $-0.1\,\text{ft/s}^2 \le \dot{V} \le 0.1\,\text{ft/s}^2$
Descent 2 (Reserve)	Fuel W_f (minimize)	$1\,\text{min} \le t \le 1{,}000\,\text{min}$, $h \le 35{,}000\,\text{ft}$, $0.4 \le M_\infty \le 0.78$[8], $-10^\circ \le \gamma \le 0^\circ$, $s \le 3{,}420\,\text{nmi}$
Hold (Reserve)	Fuel W_f (minimize)	$t = 30\,\text{min}$, $1{,}450\,\text{ft} \le h \le 1{,}500\,\text{ft}$, $0.4 \le M_\infty \le 0.5$[9], $-1^\circ \le \gamma \le 1^\circ$, $-1 \le \text{ ft/s } \dot{h} \le 1\,\text{ft/s}$
Descent 3 (Reserve)	Fuel W_f (minimize)	$1\,\text{min} \le t \le 1{,}000\,\text{min}$, $1{,}000\,\text{ft} \le h \le 15{,}000\,\text{ft}$, $0.3 \le M_\infty \le 0.5$[10], $-10^\circ \le \gamma \le 0^\circ$, $s \le 3{,}620\,\text{nmi}$, $\dot{h} \le 0\,\text{ft/s}$, $\dot{V} \le 0\,\text{ft/s}^2$
Overall	Fuel W_f (minimize)	$R = 3420\,\text{nmi} + 200\,\text{nmi}$ (design + reserve)

3.3 Novel SFNP Vehicle: SBW SROR

The novel vehicle considered for this study is the strut-braced wing (SBW) concept powered by a single rotor open rotor (SROR) novel propulsion system, as shown in Fig. 3.2. This configuration was developed by GT-ASDL as part of the NASA-sponsored Zero Emissions project. The SBW is intended to be a notional and non-restricted notional common research model inspired by the Transonic Truss Braced Wing (TTBW) concept developed by NASA and now being developed into the X-66A Sustainable Flight Demonstrator by The Boeing Company [30, 1]. The planform of the SBW is derived from the three views presented in the SUGAR Phase IV report [34]. Engineering Sketch Pad is the parametric geometry modeling tool used to develop the geometry model for the SBW. The goal of the Zero Emissions effort was to develop a system-level model of the SBW. For this to be possible, aerodynamics, structures, and propulsion models were developed. A notional mission profile was also defined for this configuration based on a typical single-aisle aircraft mission.

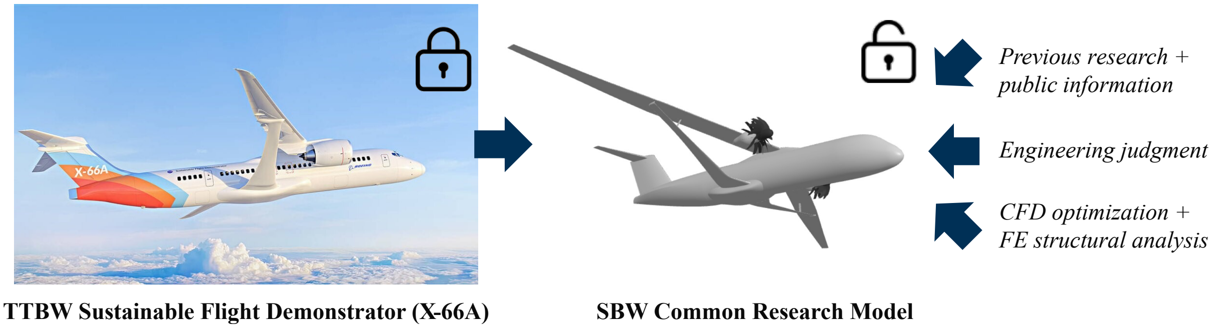

Figure 3.2: GT-SBW SROR, inspired by the NASA Sustainable Flight Demonstrator X-66A TTBW. Left image credit: NASA, public domain, via Wikimedia Commons

3.3.1 Aerodynamics Model Generation by CFD Analysis

Mission analysis for the SBW required a set of drag polars spanning the entire operating envelope. This was accomplished by first optimizing the wing and strut camber and twist distribution using fine-grid Reynolds-Averaged Navier Stokes (RANS) CFD with an actuator disc representation of the SROR in the model. The empennage, pylon, and nacelles were not part of this optimization problem, but their impacts on the aerodynamic performance were included for drag modeling later. Additionally, all CFD cases assumed fully turbulent flow, and the optimization did not focus on promoting natural laminar flow (NLF) over the geometry. This was a major departure from the work documented in the SUGAR Phase IV report, which included NLF benefits in the aerodynamics model. After the optimized geometry was obtained, the mission aerodynamics models were generated using a multi-fidelity approach.

The Zero-Emissions Project work resulted in a steady-state aerodynamic drag model implicitly including the effects of the SROR propulsion system on the airflow over the vehicle airframe, referred to here as the "weakly-coupled" model. Although the mission analysis for the Zero Emissions effort used the "weakly-coupled" steady-state drag model discussed below, two additional aerodynamics data sets were developed for this MBSA&E study, one model of the aerodynamics uncoupled from the propulsor model, and another set of CFD cases including the propulsor to model throttle-dependent drag, referred to as the "strongly-coupled" model.

Partial Coupling of Aerodynamics and Propulsion (Weakly-Coupled)

In the "weakly-coupled" strategy, the operating envelope was sampled extensively using coarse grid RANS simulations, which were significantly cheaper to run than the finer grid cases used for the outer mold line optimization. Following this, a smaller sample of fine grid RANS cases was run at key operating points as well as for a random selection of points. A hierarchical Kriging [35] model was then used to correct the

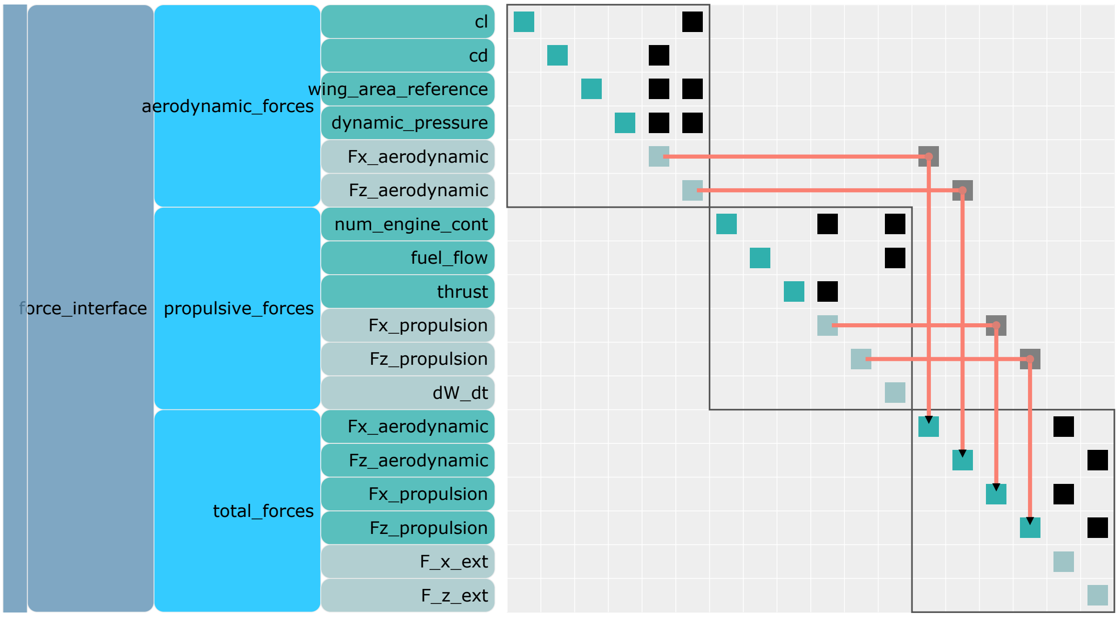

Figure 3.3: Weakly-coupled aeropropulsive force interface in strut-braced wing aircraft model

low-fidelity data with the high-fidelity results to generate a combined aerodynamics model that was sampled to produce a table predicting C_D as a function of C_L for different Mach and altitude combinations. For both the coarse and fine grid RANS cases, a horizontal tail was included in the CFD model. The incidence angle of this surface was adjusted to achieve a zero pitching moment at the center of gravity for each sample. Additionally, the SROR thrust was adjusted iteratively within a given CFD simulation to ensure that thrust and drag balanced each other for each sample. In short, equilibrium was enforced for each CFD sample.

The pylon drag was assumed to be 14 counts (a coefficient of 0.0014) at cruise conditions based on the SUGAR Phase IV report, but Mach and Reynolds number corrections were included to account for changes in flight conditions relative to the reference operating point. Similarly, the vertical tail was assumed to produce 14 counts of drag at this reference cruise condition, estimated from a CFD simulation that included the geometry for the vertical tail. Like with the pylon, the drag estimate was corrected for variations in Mach and Reynolds number

Note that since the SROR was present in the CFD model used to generate the polars, the aero-propulsion interactions were captured in the drag polar data, even though the lift and drag coefficients were computed by only integrating the pressure and shear contributions over the airframe. The mission analysis phase This coupling was not explicitly captured in however, since the models from this CFD study did not include an input for engine power setting, implying that drag was independent of throttle setting, dependent only on flight condition variables M_∞, altitude h, and lift coefficient C_L. The propulsive force model independently computed thrust as a function of M_∞, h, and P. The force interface described in Section 2.4.6 and shown in Figure 3.3 separately-bookkeeps these aerodynamic and propulsive forces before summing them in Eqs. (2.27) and (2.28). Hence, this model is referred to as "weakly coupled," since interactions of the propulsion system with flight forces were indirectly and not completely consistently accounted for.

Aerodynamics Model Uncoupled from Propulsion (Uncoupled)

The "uncoupled" approach generated the drag polars for the SBW in the same manner as above, but this time without the SROR in the CFD model. Thus, no aero-propulsion interactions were accounted for in this data set, representing a complete uncoupling of the propulsion system from the aircraft aerodynamics. The C_D and C_L estimates from this model only capture the SBW geometry impact on the airflow.

Aerodynamics Model Fully Coupled With Propulsion (Strongly Coupled)

The third data set involved calculating streamwise and stream-normal force coefficients C_{F_x} and C_{F_z} respectively. The difference between these coefficients and the traditional drag and lift coefficients is the inclusion of the propulsor contributions to the forces. In other words, the SROR thrust contribution in the x-direction in the wind axes is also included with the airframe pressure and shear contributions in the same coordinate direction. Similarly, the SROR thrust contribution in the z-direction of the wind axes is included with the contributions that would have counted towards the lift coefficient only. In this method, the contributions from the airframe and propulsor on the net streamwise and stream-normal forces are considered together, making it a more coupled approach. Like with the polar datasets, surrogate models for C_{F_x} and C_{F_z} are generated using a multi-fidelity approach. A set of 200 coarse grid RANS cases is used to sample the operating space defined by Mach number, Reynolds number, angle of attack, and power code (throttle setting). An initial Kriging model is fit to these low-fidelity data and fed to the Olympus mission analysis to assess the aircraft's initial trajectory and key operating points. These operating points, along with a smaller space-filling set of points, 30 in total, are then selected to run with a fine-grid RANS CFD simulation. Like before, hierarchical Kriging is used to combine the datasets into multi-fidelity models for C_{F_x} and C_{F_z}, which are then fed into Olympus for the final mission analysis.

3.3.2 SROR Propulsion Model

The propulsion model for the SROR is developed in NPSS. It is inspired by the CFM Rise concept, but relies on public information and physics-based analysis to complete the model. The SROR has a single puller rotor with swirl recovery vanes behind. Starting with the SR3 geometry, the blades were then optimized using an in-house blade element momentum code that accounted for compressibility effects, sweep impacts, and the interactions between the rotor and the recovery vanes. The engine architecture, in particular, is based on information drawn from two US patents: US 2021/0108597 A1 and US 2020/0308979 A1. This architecture was optimized in NPSS to provide a reasonable level of performance that is representative of next-generation concepts.

3.3.3 Structural Weights Model

On the structures side, weight estimates for the SBW fuselage and empennage are obtained from standard FLOPS regressions. The engine weight comes from WATE++. Weights for the main SBW lifting surface assembly, i.e., the wing, strut, and jury, cannot be adequately captured by the FLOPS weight regressions, which are based on conventional tube and wing aircraft. As such, a physics-based structural sizing approach was used to obtain these weights as part of the Zero Emissions project. This approach was centered around using the Nastran finite element structural analysis tool, and Collier Aerospace's HyperSizer tool as the means for failure analysis and sizing. Aerodynamic loads were provided by an Euler solution from Cart3D software and were integrated with the structure via matching-based extrapolation of loads and displacements (MELD) [36]. The primary structural geometry was modeled with plane-stress assumptions and AS-4 composite tape material properties. A number of strength- and stability-based failure modes were set up as constraints in the global-local component-based sizing process.

A design of experiments was executed with the structural sizing model to fit a surrogate model to the combined wing, strut, and jury assembly weight as a function of aircraft gross weight, engine weight, and planform area. However, engine and planform weights were fixed for this study. The team member who re-implemented the weight surrogate's in the MDAO environment validated its results by comparing to the FLOPS/EDS environment results.

The wing weight surrogates must be called outside the FLOPS execution. A gross weight for the aircraft is assumed initially to obtain the wing weights, which inform the empty weight of the aircraft. These weights then become inputs to the mission analysis. FLOPS iterates over gross weight values until it satisfies the range requirement during mission analysis [3]. The final aircraft gross weight from FLOPS may be different than the input used for the weight surrogates. As such, the aircraft gross weight input to the lifting surfaces weight surrogates is updated iteratively till this input is consistent with the value chosen by FLOPS. In the kinetic mission analysis framework, since the wing weight surrogate is integrated into the optimization, it handles this iteration internally as it designs the gross weight.

3.3.4 Mission Requirements

Finally, a notional single aisle mission is defined for the SBW as shown in Fig. 3.4. Carrying a payload of 150 passengers, the aircraft is sized for a 3,402 nmi design mission range and a 200 nmi reserve mission.

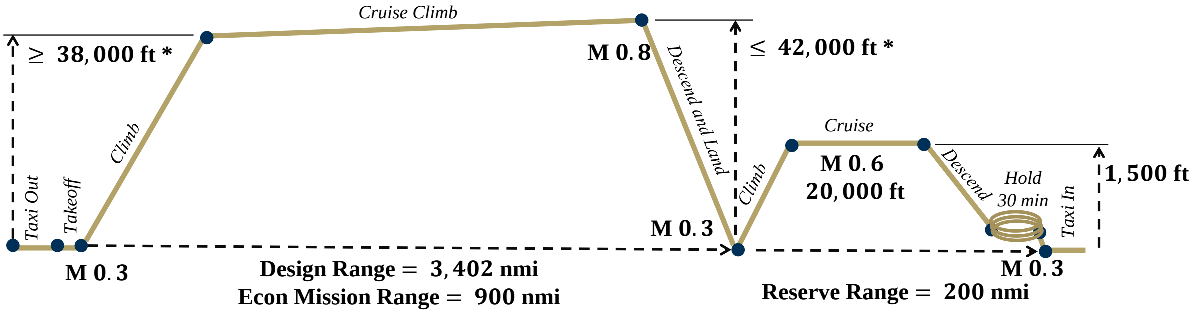

Figure 3.4: Design and reserve mission profiles for the FLOPS model of the SBW

The reserve mission includes a 30-min hold as well at 1,500 ft. An off-design mission with a 900 nmi range is also analyzed, with the same reserve mission requirements. The total fuel for the design and reserve missions is multiplied by a factor of 1.03 to determine the required fuel. In FLOPS, all climb segments are optimized for minimum time to climb. Cruise occurs at maximum specific range, and descents occur at maximum aerodynamic efficiency.

Chapter 4

Results

4.1 Conventional Vehicle Results

The conventional, tube-and-wing configuration analysis for this project was intended to establish a baseline of comparison between the various mission analysis methods being developed and used in this effort. NASA's FLOPS software was considered the control case for this set of experiments, against which the others were compared. It is expected that FLOPS results for this type of configuration are reasonably accurate, since this type of vehicle is well within the domain of data used in developing FLOPS empirical relations and analysis methods.

4.1.1 NASA Aviary Framework Analysis

Using Aviary's Level 2 API implementation, a FLOPS case was converted to an Aviary-compatible input using the conversion script included in the repository. While most variables were successfully converted over, some variables were manually converted by looking up the variable name and augmenting the converted input files. Furthermore, some of the variables that were converted over were erroneously assigned wrong values or were missing units, these entries were also corrected manually.

The mission phases were also implemented in Aviary, which also relies on the Dymos optimal control library [12] to optimize the trajectory according to a selected objective, with respect to select state and control variables. Objective options include vehicle mass, fuel carried plus a time-to-climb penalty, fuel burned, or negative of range plus a time-to-climb penalty [37]. For the present sizing problem, the objective defaults to the fuel-plus-time-to-climb option. Note that this is slightly different from the Olympus objective, which does not include a time penalty. Similar to the Olympus implementation, the Aviary formulation requires specifying phase duration bounds and phase initialization bounds. Thus, the mission segments were added incrementally to ensure the optimizer was able to solve the problem as mission segments were added to the problem.

Initial tests were run next to understand the differences between Aviary's formulation and to alter input files to replicate the mission setup implemented in FLOPS and Olympus. Most notably, during the early phases of development, Aviary's reserve mission analysis was not fully functional. Later, with the reserve mission then functional, its implementation, unlike that of FLOPS and Olympus did not allow for distance state across multiple phases of the reserve portion of the mission to be constrained. Instead, the developers included state constraints for each phase of the reserve portion of the mission. This phase constraint feature was then used in a trial-and-error loop to mimic the trajectory produced by Olympus.

Furthermore, while Olympus defined a *hold* segment as one where the vehicle maintained a steady fuel burn at a fixed throttle setting, the distance state did not accumulate during the phase. In contrast, Aviary, while maintaining a steady fuel burn, also tracked the distance traveled during the phase. Hence, the distance state in the Aviary distance cross-plots shows distance as tracking steadily in contrast to the other two tools.

4.1.2 Results and Comparisons

The following section includes plots that overlay data from all three tools on the same axes, for the sizing mission (design plus reserve portions). However, due to the differences in fidelity, not all data are uniformly available. Additionally, while all three codes were run with both a main mission and a reserve mission, it should be noted that for all the results shown, the FLOPS code does not provide state data for the reserve mission as part of it results. Hence, FLOPS state data for its reserve mission have been excluded. Furthermore, the tables included in this section detail data obtained for overall mission phase types (not separating climb into FAA and non-FAA portions) to streamline the relevant observations.

The following figures illustrate various features of the optimal trajectories found by each of the mission analysis tools, in terms of a few key state variables. Each trajectory represents a sized vehicle of a certain initial ramp weight flying a series of mission phases as defined previously (Table 3.1 and 3.2). Both FLOPS and Aviary were free to design ramp weight (or mass, in Aviary's case) in order to fly the given mission range in such a way to minimize the respective objectivve function. They thus optimized the initial weight point in Table 4.4 and Figure 4.2. Olympus, however Figure 4.1 is a comparison plot containing the timeseries for altitude and velocity magnitude versus distance state curves for the results obtained from all three codes overlaid on the same axes. It can be seen that the converged solutions enforce the distance constraints similarly with the notable exception of the reserve mission results generated by the Aviary tool as discussed in section 4.1.1 previously.

Across the subplots in the figure, it can be seen that the FLOPS controls create sharp jumps in velocity and altitude corresponding to the step climb that was coded into the setup.

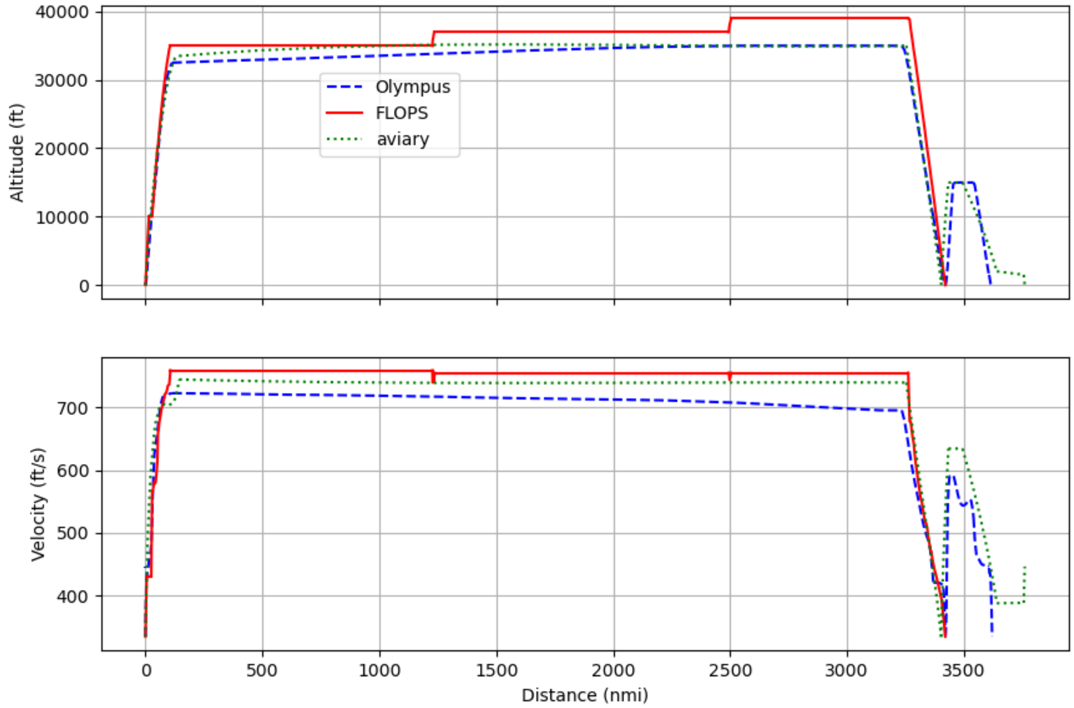

Figure 4.1: A320neo Distance Comparison Cross-plots

Furthermore, it is also worth noting that Olympus predicts an optimal trajectory for the A320neo vehicle has a cruise phase at an overall lower velocity magnitude and altitude, compared to either of the FLOPS or Aviary trajectories (Figure 4.1). It also gradually reduces speed, especially toward the end of the cruise, taking advantage of the slightly nonzero acceleration allowed in that mission phase. This is also illustrated in the durations listed by segment in Table 4.1.

These differences between Olympus, Aviary, and FLOPS may be due to the result of different modeling approaches or differences in optimizer settings. Recall the slightly different objective functions for the tools, FLOPS taking a more granular, per-phase approach than Olympus which simply minimizes the overall mission fuel burn.

Table 4.1: A320neo Results Comparison: Segmentwise Duration (min)

Phase	FLOPS	Olympus	Aviary
Climb	16.35	19.29	22.54
Cruise	434.96	444.79	425.41
Descent	32.45	23.72	29.18

Although not visible in the distance crossplot figure, Table 4.2 shows that the per-phase distances traveled estimated by each tool were of similar order but varying in magnitude. Aviary in particular is short by approximately ten nmi from the distance constraint imposed. While this was not investigated further, it is possible that control over this behavior may be unavailable at the Level 2 API.

Table 4.2: A320neo Results Comparison: Segmentwise Distance (nmi.)

Phase	FLOPS	Olympus	Aviary
Climb	96.80	111.81	144.49
Cruise	3236.30	3125.73	3108.57
Descent	158.70	129.11	148.94

Table 4.3: A320neo Results Comparison: Segmentwise Altitude (ft.)

Phase	FLOPS		Olympus		Aviary	
	Start	End	Start	End	Start	End
Climb	0	35000	0.00	32217.97	0.00	33499.28
Cruise	35000	39000	32217.97	34200.59	33499.28	35000.00
Descent	37000	0	34200.59	10000.00	35000.00	500.00

Figure 4.2 focuses on the beginning and the end of the main mission phases while also introducing a new state, weight to the comparison. The altitude vs distance plot shows that while both FLOPS and Olympus' results show that the A320neo's distance value at the end the descent phase at the main mission range mark, Aviary's solution ends the descent phase earlier.

On a different note, the weight vs distance plot shows a constant offset between FLOPS/Olympus and Aviary due to how the Olympus model was initialized using the FLOPS sized vehicle. This can be seen in Table 4.4 which shows the weight at the start of each segment for each of the tools. Also recall that FLOPS included a 10-nmi cruise midway through its climb to cruising altitude, which neither Olympus nor Aviary models accounted for. Experimenting with removing this segment from the FLOPS mission definition slightly reduced the converged vehicle ramp weight, as may be expected due to eliminating fuel burn for the extra acceleration from cruise to climb.

Again note that Olympus selects a lower, decreasing velocity during the cruise phase, resulting in a longer mission time versus Aviary or FLOPS' analyses, as evident from the data plotted in Figure 4.3. One may hypothesize that the difference in objective — maximum range versus maximum specific range — might account for some of this difference in cruise velocity.

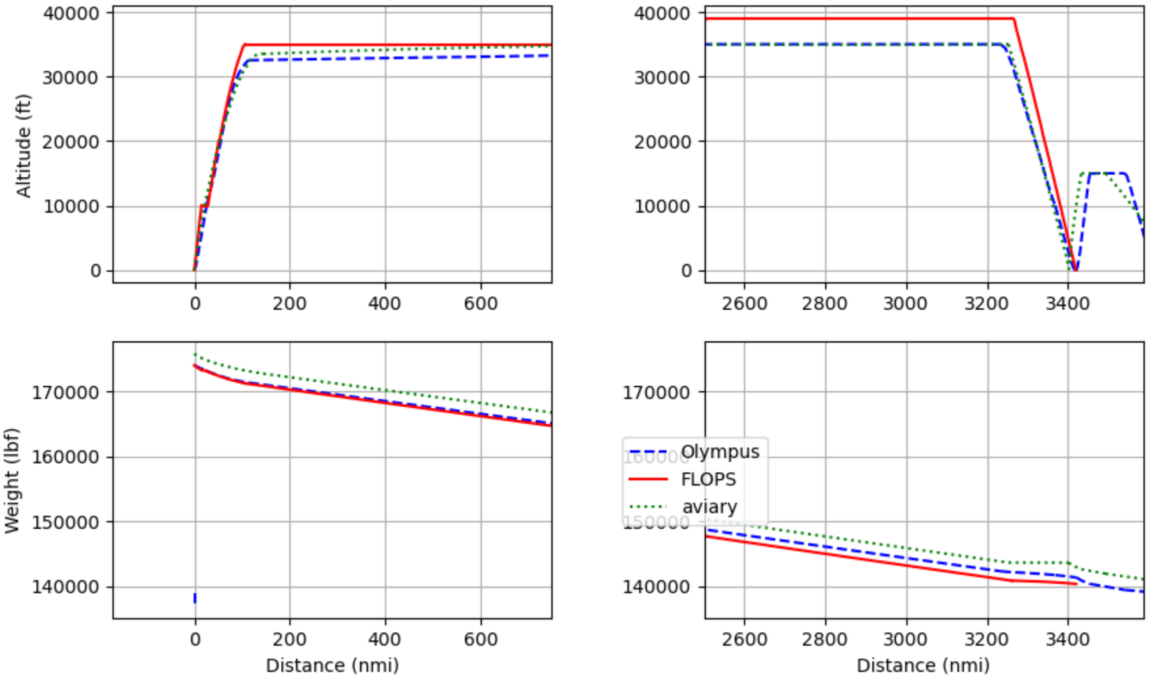

Figure 4.2: A320neo Distance Comparison Cross-plots

Table 4.4: A320neo Results Comparison: Segmentwise Weight (lbf)

Phase	FLOPS	Olympus	Aviary
Climb	174020	174020.00	175792.45
Cruise	147691	171348.01	172736.64
Descent	140860	142280.51	143642.52

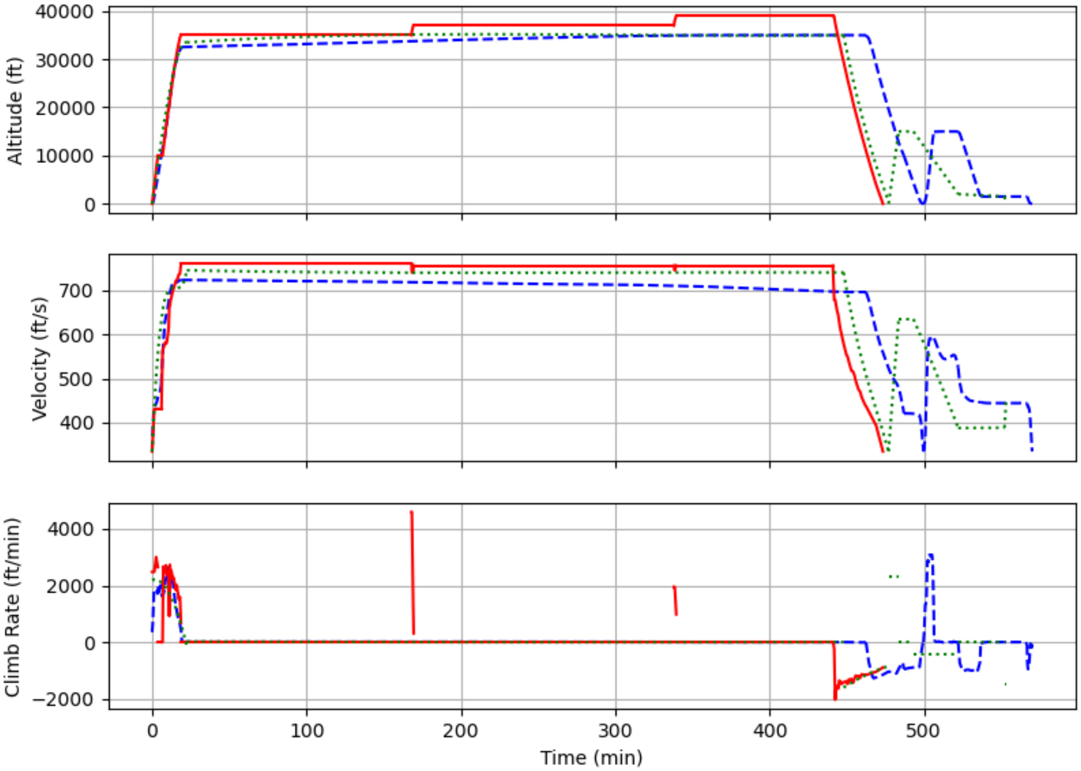

Figure 4.3: A320neo Time History Comparison

The climb rate subplot in Figure 4.3 shows spikes in the FLOPS climb rate command at the step cruise time marks. This discontinuous spiking is consistent with the fact that FLOPS does not directly solve the equations of motion throughout the entire mission, rather allowing somewhat non-physical state transitions especially at the boundaries of mission phases like climb and cruise. Olympus on the other hand enforces continuity in states and state rates such as climb rate, via the governing equations of motion (Eqs. (2.9) and (2.10)) common to all flight phases.

At the level of the overall mission and vehicle final, optimized vehicle ramp weights, range, total sea-level static thrust, wing planform area, and other weights of interest are listed for each type of analysis in Table 4.5. In this table, T/n_e represents thrust per scaled engine at sea-level-static conditions. OEW is the operating empty weight, that is, weight of the aircraft before any mission fuel, passengers, or other cargo is loaded. It includes miscellaneous operational overhead weights such as lubricant, safety equipment, and crew. The payload weight W_P includes passengers and their baggage. The change in weight over the design portion of the mission $W_{f,design}$ represents the fuel burned during the design mission of 3,420 nmi. Total theoretical fuel loaded on the aircraft W_f includes reserve fuel used for the 200-nmi reserve mission, plus the 5% margin on the design mission fuel burned.

The reader should note that none of the fuel weights shown here account for ground segments besides initial taxi-out and takeoff. In the kinetic analysis taxi-out and takeoff fuel burn is not modeled as other mission phases, but simply estimated as a fixed percentage of the ramp weight, based on the fuel fraction from FLOPS (99.4105%). Also note that neither FLOPS nor the kinetic analysis properly constrain fuel load according to fuel tank capacity. This means it is possible for a maximum fuel load ($W_f = W_0 - OEW$) to exceed the specified available fuel tank capacity, and therefore the fuel structural mass may be somewhat inconsistent with the fuel weight shown in Table 4.5.

Differences in the computed fuel weights between the tools here may be attributable partly to different objective functions and constraints. Namely, the FLOPS minimum fuel-to-climb and descent max L/D objective differs from Aviary's composite, fuel-plus-time-to-climb objective, which also differs from the kinetic analysis' minimum overall fuel burn objective objective. In the end, the aircraft designer must select the objective according to the design or analysis goals. As for constraints, the FLOPS analysis was constrained to fly at discrete, fixed altitudes and a fixed cruise Mach number, whereas the kinetic analysis could optimize these for minimum fuel burn in cruise as well. This explains why the gradual deceleration in velocity (Figure 4.1) may be observed in the kinetic analysis compared to the FLOPS analysis. FLOPS had the option to step-climb, choosing altitudes

Table 4.5: Results of Analyses of Conventional Aircraft for 3,420-nmi Sizing Mission

Analysis	W_0 [lbf]	T/n_e [lbf] [1]	S [ft^2]	OEW [lbf]	W_P [lbf]	$W_{f,design}$ [lbf]	W_f [lbf]
FLOPS	175,052	27,076.7	1,341.10	100,965	33,000.0	34,701	41,087.3
Aviary	175,792.4	27,076.7	1,341.10	105,879.15	33,000.0	32,159.5	36,913.3
Olympus	175,052	27,076.7	1,341.10	100,670	33,000.0	33,704.3	41,381.71

The kinetic mission analysis framework also includes the ability to fix ratios of thrust to weight and wing loading, computing the thrust and wing area as functions of these parameters and the sized vehicle's ramp weight, mimicking the FLOPS sizing mode. The result of running this mode with the conventional vehicle is summarized in Table 4.6.

Table 4.6: Sized Vehicle: Parametrically Varied T and S

Analysis	W_0 [lbf]	T/n_e [lbf] [2]	S [ft^2]	OEW [lbf]	W_P [lbf]	ΔW_{design} [lbf]	W_f [lbf]
Olympus	171,857	26,586.3	1,316.6	99,677.9	33,000.0	33,097.6	39,179.22

4.2 Novel Vehicle Results

The mission analyzed is a 4450 nmi sizing mission, which includes a 4250 nmi design mission followed by a 200 nmi reserve mission. Three distinct cases are being run for the SBW SROR, summarized below:

- **Uncoupled mission analysis** - Uses a conventional decomposition of forces: lift, weight, thrust, drag. There is no SROR in the CFD model (i.e., CL and CD do not account for any rotor-airframe interactions).

- **Weakly-coupled mission analysis** - Uses a conventional decomposition of forces: lift, weight, thrust, drag. There is a SROR in the CFD model, and therefore, rotor-airframe interaction impacts are captured in the calculated CL and CD from the geometry.

- **Strongly-coupled mission analysis** - There is no separation of forces and the general force decomposition of F_x and F_z is used instead. There is a SROR in the CFD model, and therefore, rotor-airframe interaction impacts are captured in the calculated CL and CD from the geometry.

All three of the above cases are run in Olympus, while only the first two of these cases are run in FLOPS because it is unable to do a general F_x and F_z force decomposition that is required for the third case.

[1] Thrust per engine, sea-level static max rated thrust
[2] Thrust per engine, sea-level static max rated thrust

4.2.1 Legacy Analysis

The FLOPS mission profile for the SBW SROR uncoupled sizing mission analysis case can be seen in Figure 4.4 while the weight dependence with time can be seen in Figure 4.5. The aircraft climbs up to an altitude of 40,196 ft, does a Mach 0.8 steady speed cruise-climb up to an altitude of around 42,163 ft, and then begins to descend to its final destination. This entire mission takes around 491.74 minutes for the aircraft to complete.

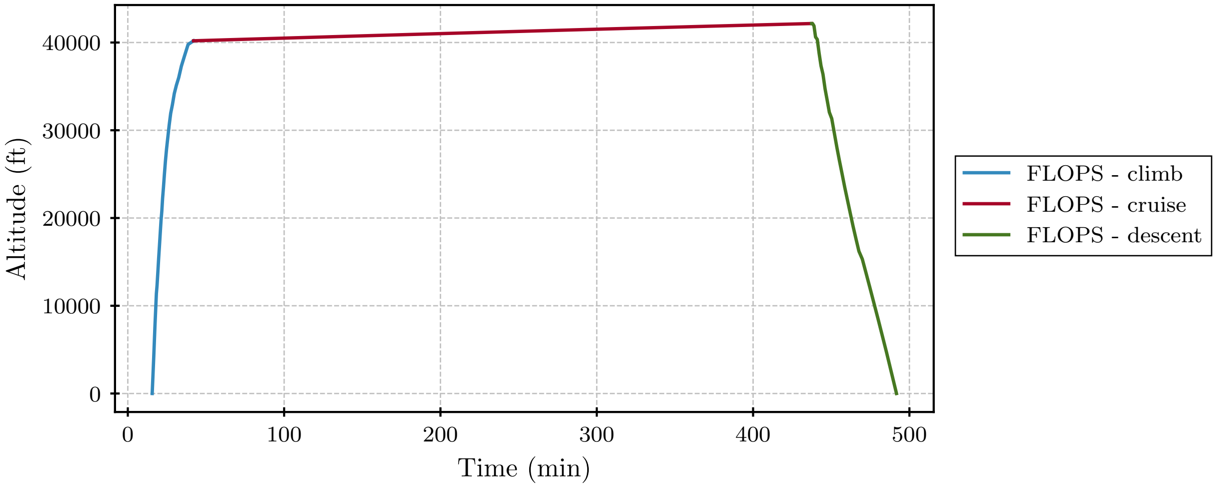

Figure 4.4: Altitude vs Time for SBW SROR FLOPS uncoupled sizing mission analysis

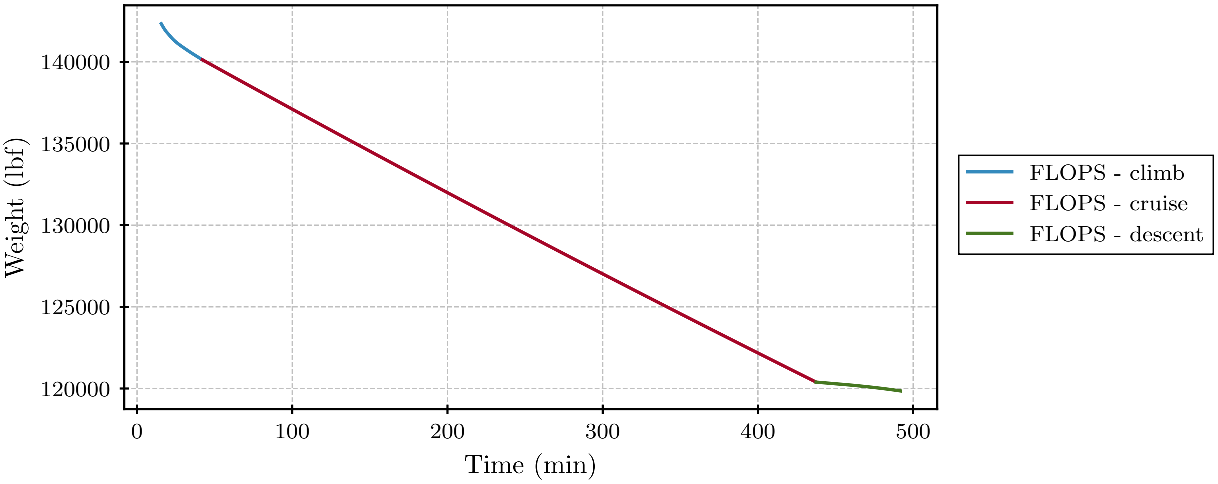

Figure 4.5: Weight vs Time for SBW SROR FLOPS uncoupled aeropropulsive sizing mission analysis

The second case being run in FLOPS is the SBW SROR weakly-coupled aeropropulsive sizing mission analysis case, for which the results can be seen in Figure 4.6 and Figure 4.7 respectively. The aircraft climbs up to an altitude of 38531 ft, which is slightly lower than before, then proceeds to do a Mach 0.8 steady speed cruise-climb. Once the aircraft reaches an altitude of 40469 ft, it starts to descend to its final destination. The entire mission takes around 484.10 minutes to complete, which is a slightly faster time than in the uncoupled sizing mission analysis case.

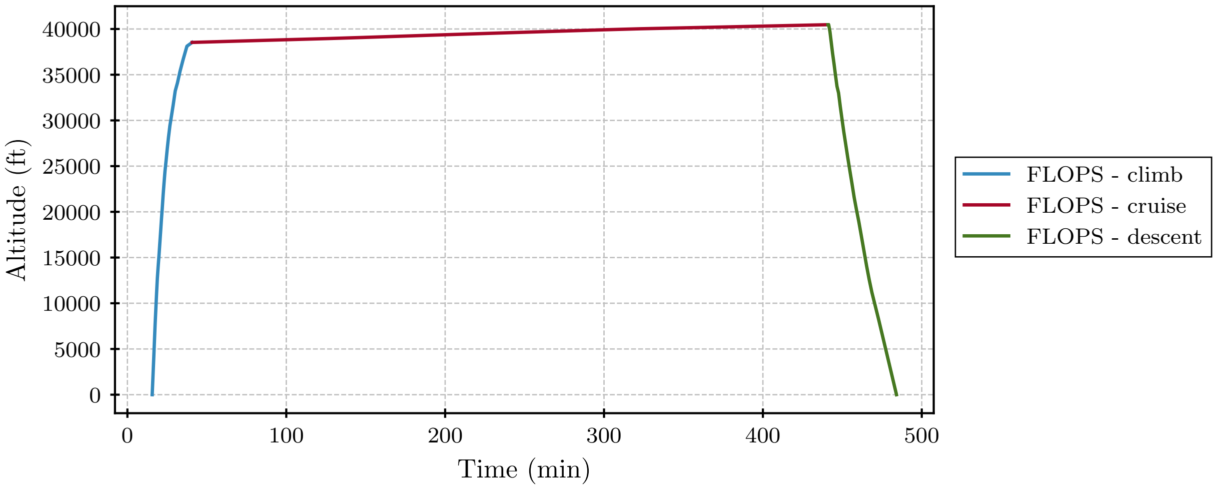

Figure 4.6: Altitude vs Time for SBW SROR FLOPS weakly-coupled aeropropulsive sizing mission analysis

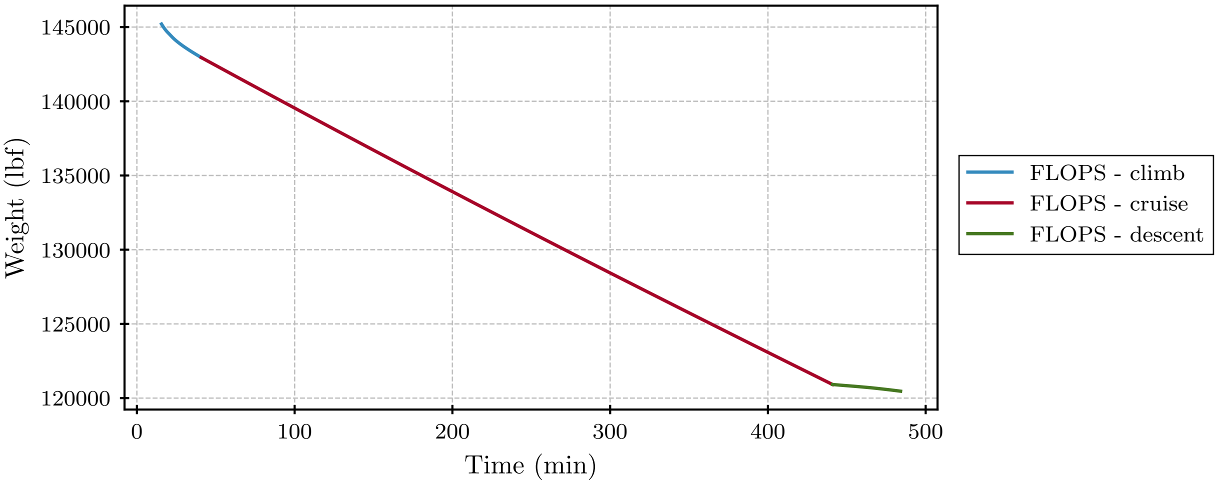

Figure 4.7: Weight vs Time for SBW SROR FLOPS weakly-coupled aeropropulsive sizing mission analysis

4.2.2 Kinetic Analysis

The Olympus mission profile for the first case, namely the SBW SROR uncoupled sizing mission analysis, can be seen in Figure 4.8 while the weight change with time can be seen in Figure 4.9. It can be observed that the aircraft climbs up to an altitude of around 41500 ft after which it reaches its cruise Mach number of 0.8 and proceeds to do a steady speed cruise climb for the majority of the mission. As it approaches its final destination, the aircraft begins its descent phase at an altitude of 44430 ft. There is a slight change in the descent rate initially, however it does level off to a constant value after a short period of time. The 200 nmi reserve mission is included as well in these plots, with the aircraft climbing back up to an altitude of 20000 ft and cruising for a very short amount of time after which it descends to an altitude of 1500 ft. At this 1500 ft altitude, it proceeds with a 30 minute hold segment followed by a final descent to its destination. The duration of the design mission is approximately 482.5 minutes and the duration of the reserve mission is 74.7 minutes, coming up to a total sizing mission duration of 557.2 minutes.

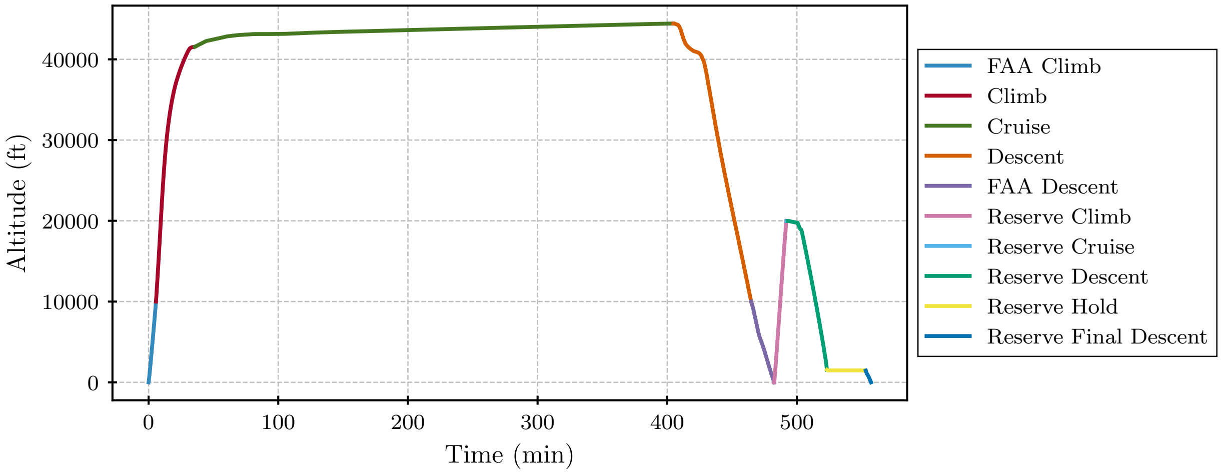

Figure 4.8: Altitude vs Time for SBW SROR Olympus uncoupled sizing mission analysis

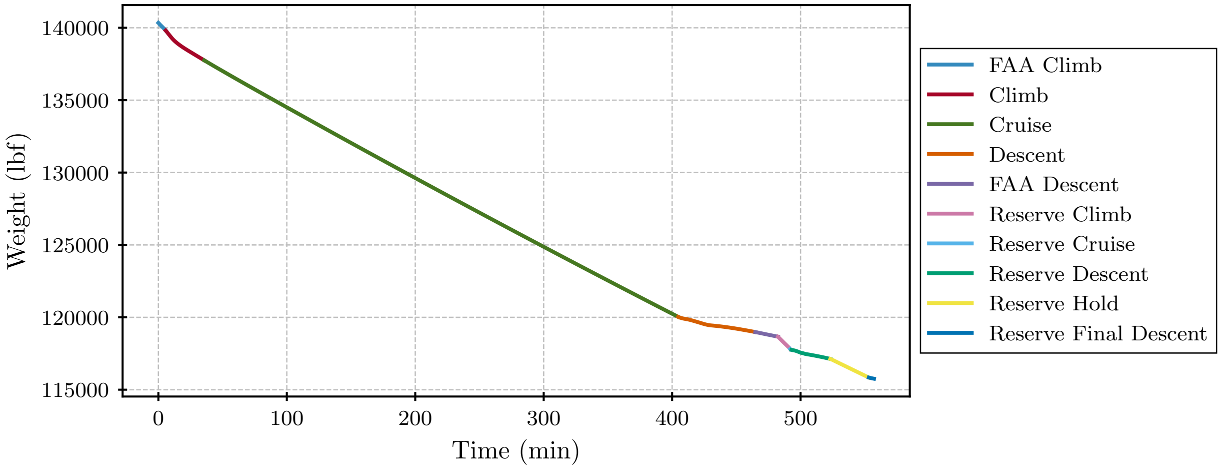

Figure 4.9: Weight vs Time for SBW SROR Olympus uncoupled aeropropulsive sizing mission analysis

The second case of the SBW SROR weakly-coupled aeropropulsive sizing mission analysis is run in Olympus, with the mission trajectory and weight-time dependence being shown in Figures 4.10 and 4.11 respectively. The aircraft climbs up to an altitude of 39750 ft after which it begins its Mach 0.8 steady speed cruise-climb. Towards the very end of the cruise-climb segment, there is a slight increase in the climb rate, followed by a sharp decrease, with the aircraft even losing some altitude before it officially began its descent segment at an altitude of 41040 ft. The descent is very gradual in the beginning, followed by an increase in descent rate later on until a certain constant value is reached. The reserve mission is largely similar to the one in the first case, with the exception that the aircraft starts almost immediately losing altitude once it reaches its 20000 ft cruise altitude. The total mission time is 551 minutes, with a design mission time of 475.6 minutes and a reserve mission time of 75.4 minutes.

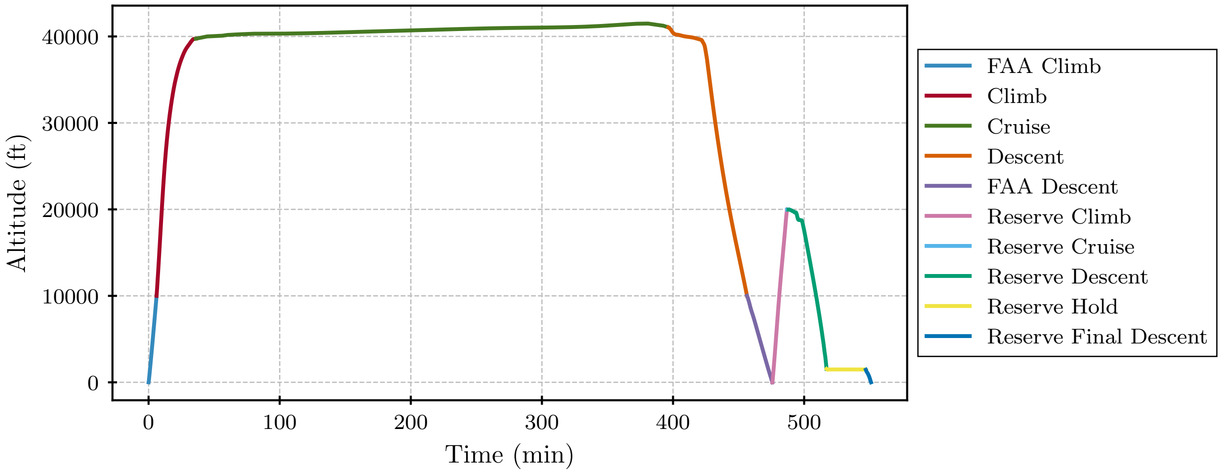

Figure 4.10: Altitude vs Time for SBW SROR Olympus weakly-coupled aeropropulsive sizing mission analysis

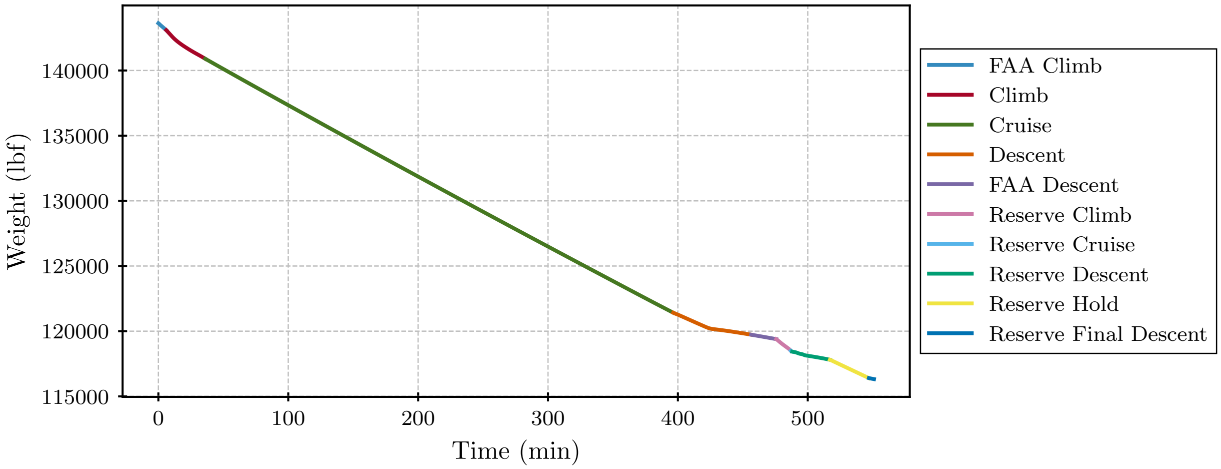

Figure 4.11: Weight vs Time for SBW SROR Olympus weakly-coupled aeropropulsive sizing mission analysis

The results of the final case run in Olympus, namely the SBW SROR strongly-coupled aeropropulsive sizing mission analysis, are shown in Figures 4.12 and 4.13. It can be seen that the aircraft climbs up to an altitude of 38840 ft when it reaches its cruise Mach number of 0.8. There is a sudden decrease in altitude of around 600 ft at the beginning of the cruise-climb phase, followed by a standard gradual increase in altitude until the aircraft comes close to its final destination. The descent happens at a starting altitude of 39500 ft. The reserve mission is somewhat similar to the second case, with the exception being that the first descent segment has a much higher descent rate initially. The first descent also happens immediately when the aircraft reaches 20000 ft, meaning that it is not very fuel efficient for it to cruise at this altitude. The total mission time is 552 minutes, with a design mission of 481 minutes and a reserve mission time of 71 minutes.

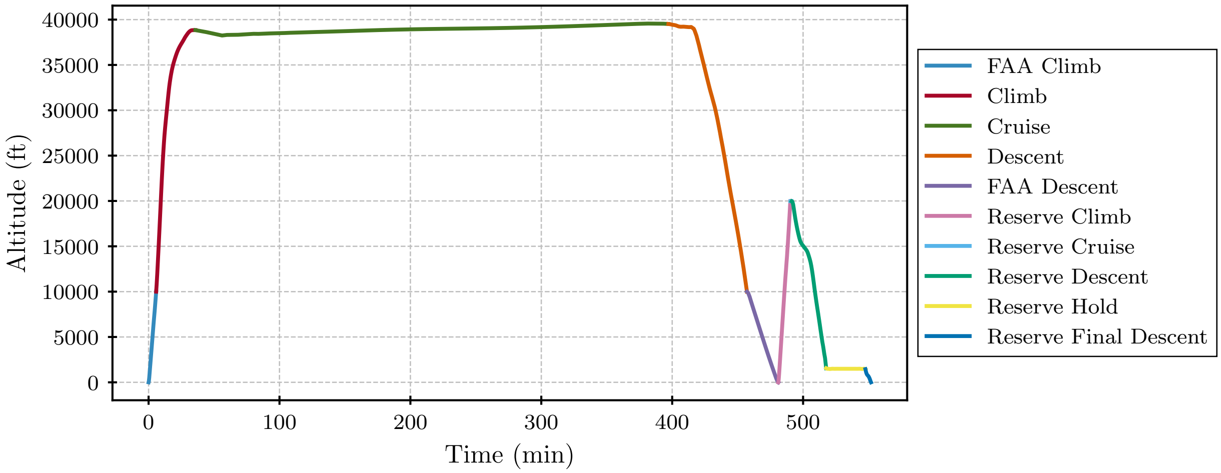

Figure 4.12: Altitude vs Time for SBW SROR Olympus strongly-coupled aeropropulsive sizing mission analysis

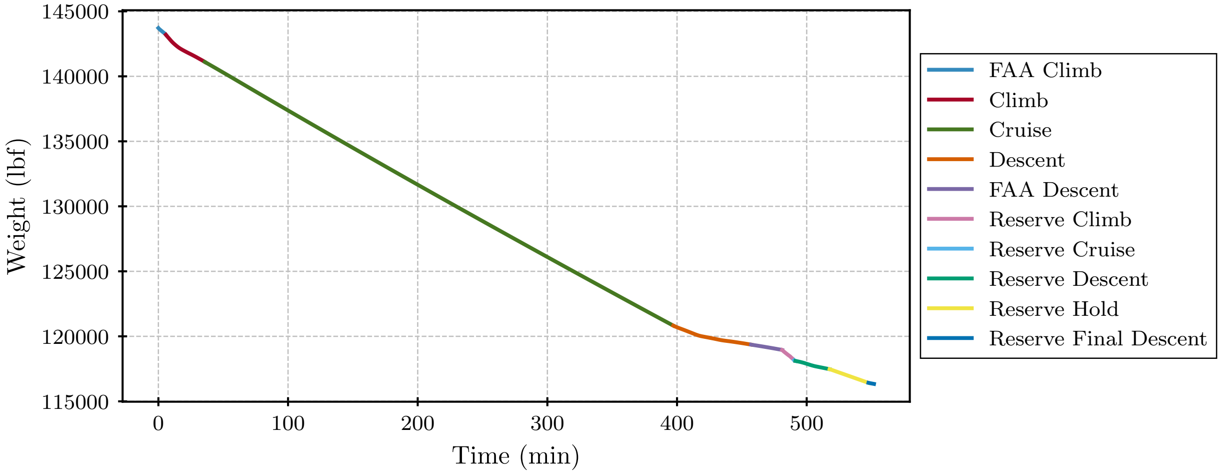

Figure 4.13: Weight vs Time for SBW SROR Olympus strongly-coupled aeropropulsive sizing mission analysis

4.2.3 Comparisons

Uncoupled Mission Analysis

A direct comparison between the FLOPS and Olympus runs for the SBW SROR uncoupled mission analysis can be seen in Figures 4.14, which depicts the change in altitude with distance traveled, and 4.15, which depicts the change in weight with distance traveled. In the first of these plots, it can be observed that the climb segment is largely the same between the two codes up to an altitude of 40,000 ft when the aircraft in the FLOPS run does an immediate change in its rate of climb and starts its cruise climb segment. In the Olympus case, the aircraft continues to climb to a higher altitude, and it has a more gradual transition to the cruise segment. In the Olympus case, the aircraft also begins its descent further away from the final destination than in the FLOPS case, with a much more gradual descent happening compared to the FLOPS case, which is almost instant.

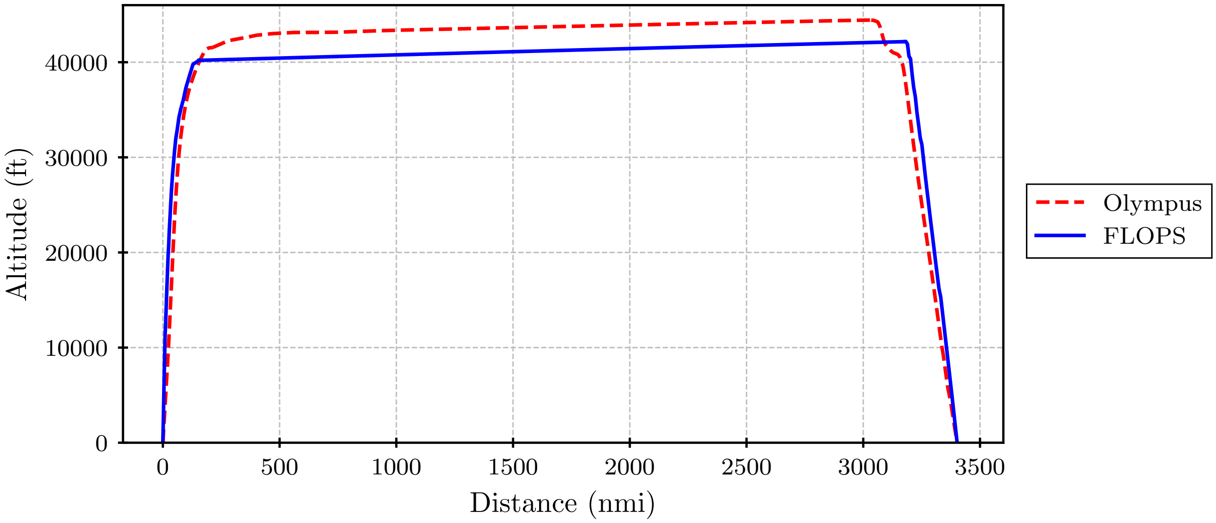

Figure 4.14: Altitude vs Distance (Comparison between Olympus and FLOPS for SBW SROR uncoupled aeropropulsive sizing mission analysis)

The reason why the SBW aircraft can climb to a higher altitude in the Olympus run is because it has a lower ramp weight, as can be seen in Figure 4.15. As the aircraft travels towards its final destination, the gap in instantaneous gross weight between the FLOPS and Olympus runs starts narrowing.

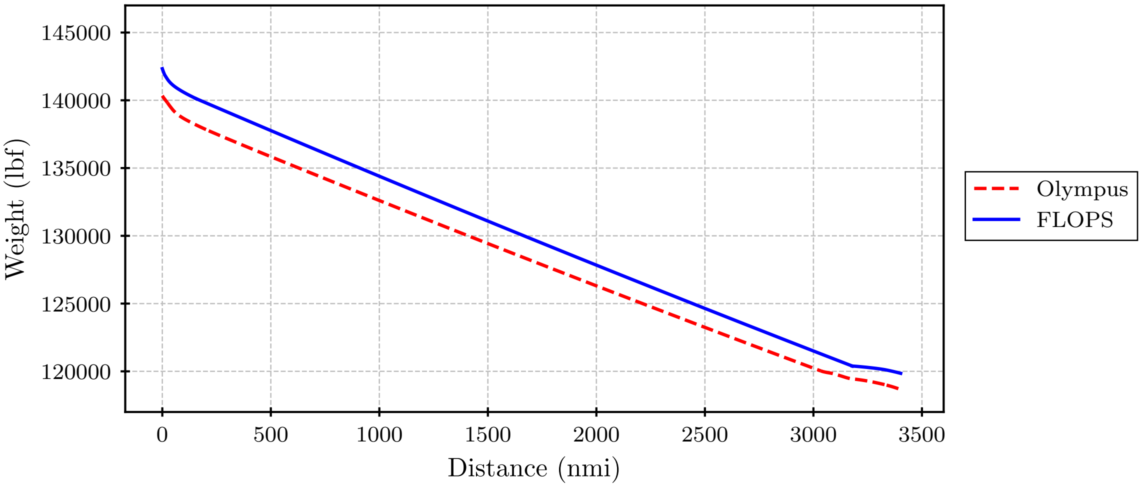

Figure 4.15: Weight vs Distance (Comparison between Olympus and FLOPS for SBW SROR uncoupled aeropropulsive sizing mission analysis)

Weakly-coupled Aeropropulsive Mission Analysis

The second comparison that can be made between FLOPS and Olympus is for the SBW SROR weakly-coupled aeropropulsive sizing mission analysis, shown in Figures 4.16 and 4.17. Again, the climb portion between the two runs is largely the same up to an altitude of around 38500 ft when the aircraft begins its cruise climb segment in the FLOPS case, while it continues climbing to a slightly higher altitude in the Olympus case. The discrepancy in cruising altitude between the two codes in the weakly-coupled aeropropulsive case is however significantly smaller than in the uncoupled case. The descent again happens slightly earlier in the Olympus run. Both the transitions from climb to cruise and from cruise to descent are also much more gradual in the Olympus run compared to the FLOPS run, where the transitions are essentially instantaneous.

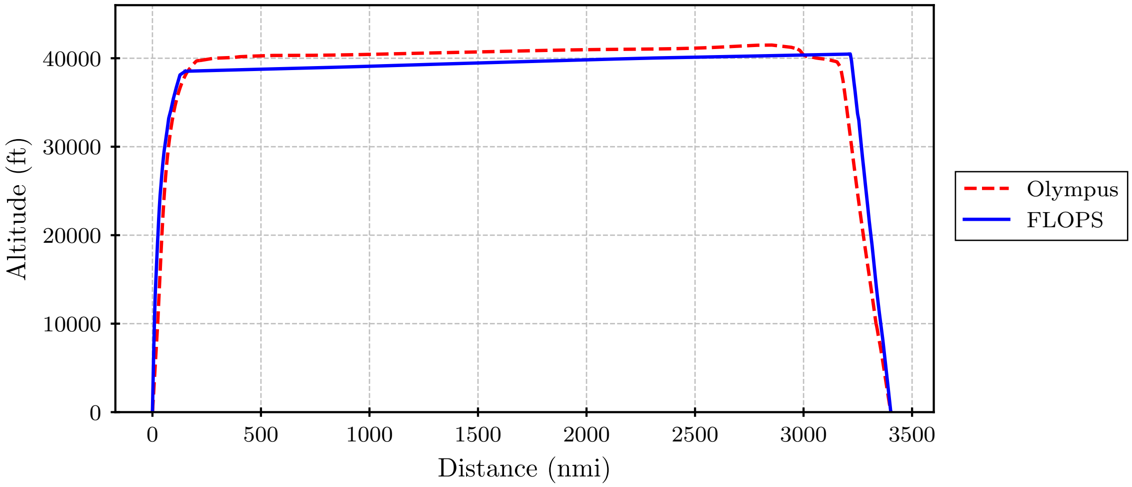

Figure 4.16: Altitude vs Distance (Comparison between Olympus and FLOPS for SBW SROR weakly-coupled aeropropulsive sizing mission analysis)

In Figure 4.17, it can be seen that the converged ramp weight of the SBW in the Olympus run is slightly lower than the ramp weight of the SBW in the FLOPS run, which in turn allows it to reach slightly higher altitudes as a result of less lift being required. The gap in instantaneous gross weight between the two runs starts narrowing as the aircraft gets closer to its final destination.

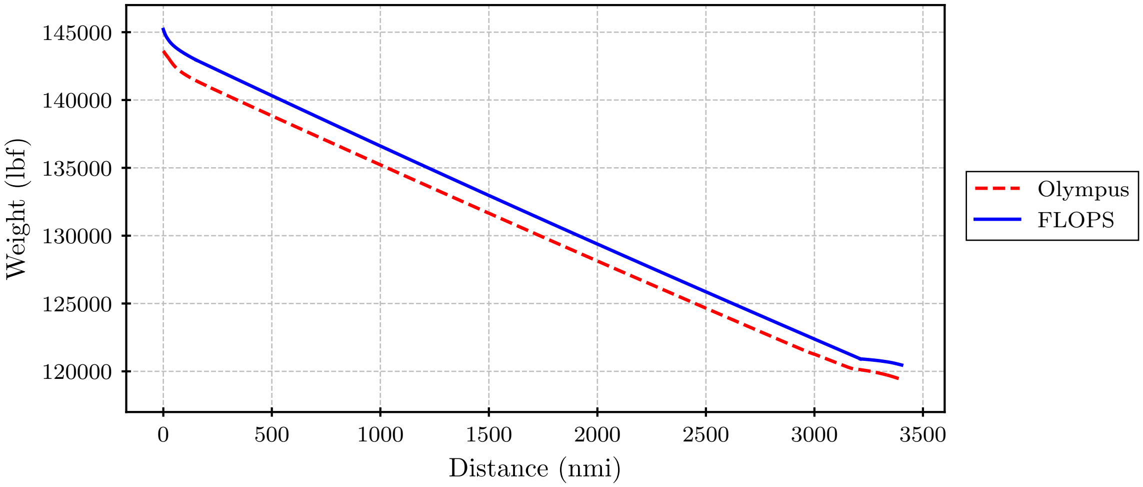

Figure 4.17: Weight vs Distance (Comparison between Olympus and FLOPS for SBW SROR weakly-coupled aeropropulsive sizing mission analysis)

4.2.4 Summary

A bar chart showcasing the ratio of fuel used up during each flight segment for all 5 runs can be seen in Figure 4.18. It can be observed that the vehicles from FLOPS runs expend a higher percentage of fuel during the cruise segment than the vehicles from the Olympus runs. The opposite is true when it comes to the climb and descent segments, with the vehicles from the Olympus runs expending a larger fuel portion during these segments than their FLOPS counterparts. This is especially evident for descent, which can be explained by the fact that FLOPS assumes an idle descent condition and extrapolates its fuel consumption accordingly, while Olympus enforces a throttle value of at least 0.05. The reserve mission fuel usage is similar across all 5 runs with minimal differences.

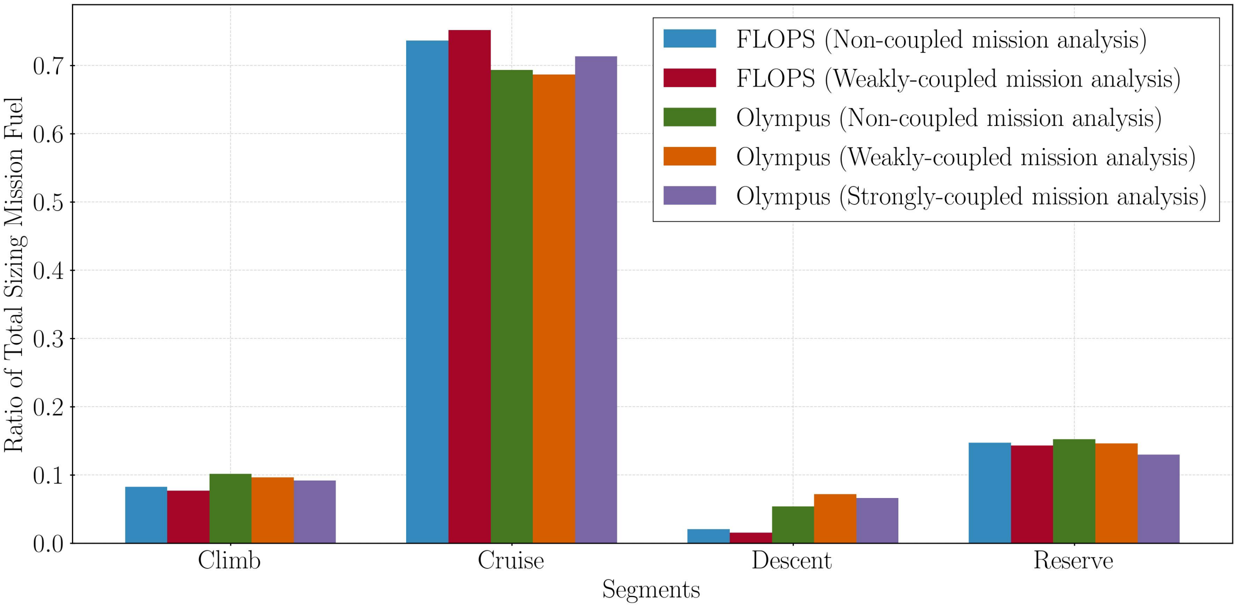

Figure 4.18: Percentage of total sizing mission fuel taken up by each phase for all 5 analysis cases

The main results of interest, including the ramp weight, design mission fuel weight, reserve mission fuel weight, and the total mission fuel weight, for the SBW SROR from all 5 runs are summarized in Table 4.7. It can be seen that the ramp weight and fuel weights are lower in the Olympus runs when compared to their equivalent FLOPS runs. The addition of aeropropulsive effects leads to an increase in ramp weight and fuel weight in both FLOPS and Olympus. The increase in ramp weight due to the addition of aeropropulsive effects caused by the introduction of the SROR in the CFD model is around 2.02% in FLOPS and 2.34% in Olympus. When it comes to the total mission fuel weight, considering aeropropulsive effects leads to a 9.38% increase in FLOPS and a 10.47% increase in Olympus. Finally, using a general force decomposition approach in Olympus only leads to a 0.063% percentage increase in ramp weight and a 0.27% percentage increase in total mission fuel weight. While this difference may seem small, the aircraft's trajectory using the general force decomposition is quite different than the one using the conventional force decomposition, as it was shown earlier. This is also evident in the discrepancy in the design mission and reserve mission fuel weights, with the general force decomposition approach run using 2.19% more fuel during the design mission and 10.97% less fuel during the reserve mission compared to its conventional force decomposition counterpart.

Table 4.7: Summary of SBW SROR sizing mission results for different case studies

Model	Ramp Weight [lbf]	Design Mission Fuel Weight [lbf]	Reserve Mission Fuel Weight [lbf]	Sizing Mission Fuel Weight [lbf]
FLOPS (Uncoupled mission analysis)	142581	22871	3934	26805
FLOPS (Weakly-coupled mission analysis)	145454	25132	4188	29320
Olympus (Uncoupled mission analysis)	140582	21680	3888	25568
Olympus (Weakly-coupled mission analysis)	143871	24234	4140	28374
Olympus (Strongly-coupled mission analysis)	143961	24764	3686	28450

4.3 Influence of Optimizer Settings

Prior to this particular MBSA&E effort, researchers at GT-ASDL had recognized that optimizer settings could account for the difference between a trajectory optimization such as those described here, converging or not. Convergence for the purposes of this work means the optimizer exited with a "success" status message, saying "Optimal Solution Found." Settings are optimizer specific, but the Interior Point OPTimizer IPOPT tool [14] used for this study has many settings, which the documentation writers humorously acknowledge [38], which include a number of tolerances and search parameters such as listed in Table 4.8.

Most of the settings retained values at or near their default values as set by the developers of IPOPT. For others, the researchers tuned them to suit the needs of the present work, usually either to obtain more precise results or to save computational time. Larger tolerances typically led to what seemed to be nonlinearly faster convergence times, as measured in numbers of IPOPT iterations. When debugging or rapidly iterating on certain aspects of the vehicle models, some researchers set tolerances more leniently to help expedite convergence. Though most model runs had iteration maximums (`max_iter`), these did not affect the results, provided the optimizer converged to a point meeting its optimality criteria based on the given tolerances, etc. The overall tolerance (`tol`) for much of the work was large enough that other settings, presumably absolute tolerances like `constr_viol_tol` and `compl_inf_tol` from Table 4.8, seemed to actively determine the stopping point. Another type of setting found to lie between problems which converged and those which did not are the barrier parameter settings `mu_init` and `mu_strategy`. In particular, a version of the SBW model converged with an adaptive `mu_strategy` but stopped converging with the default Fiacco-McCormick strategy (tested due to its usage by the Aviary [4] developers) with a small `mu_init` $= 10^{-5}$. Hopefully, this discussion gives a sense of the level of precision with which the NLP for these optimization problems is solved in this work.

Table 4.8: Subset of Optimizer Settings

Setting	Name in IPOPT Options [38]	Usage in Current Work
Overall nonlinear problem relative error	`tol`	≤ 1, exactly 1 for SBW and 0.001 for conventional vehicle
Constraint violation absolute tolerance	`constr_viol_tol`	usually $\leq 10^{-4}$ default in IPOPT
Complementarity absolute tolerance	`compl_inf_tol`	usually $\leq 10^{-4}$ default in IPOPT
Dual infeasibility absolute tolerance	`dual_inf_tol`	left at IPOPT default: 1
Barrier parameter initial guess	`mu_init`	10^{-5} (default in IPOPT: 10^{-1})
Barrier parameter update strategy	`mu_strategy`	varied throughout project (default in IPOPT: "monotone" Fiacco-McCormick)

Chapter 5

Conclusion

The traditional approach for aircraft mission analysis using the familiar force decomposition into lift, weight, thrust, and drag faces challenges when applied to novel configurations that exhibit strong aero-propulsive coupling. A clear, unambiguous force decomposition is not always possible. The research discussed in this report presented an alternative formulation to the equations of motion that is more generalizable and amenable to modern trajectory optimization strategies. The force decomposition required by the traditional approach is no longer necessary.

A new tool called Olympus was developed that conducted mission analysis using this kinetic formulation of the equations of motion. To demonstrate this new approach and assess the benefits and drawbacks over the traditional methodology, two use cases were defined. The first use case was a conventional tube and wing vehicle that is intended to be a notional representation of an Airbus A320neo single-aisle aircraft. The second use case was a more advanced, novel, single-aisle tube and wing called the Strut Braced Wing. The Strut Braced Wing is intended to be a notional representation of Boeing's Transonic Truss Braced Wing (now X66) concept, developed as part of the NASA-funded Zero Emissions project. However, unlike the X66, the SBW is powered by a large Open Rotor. This rotor wake influences the airflow over the wing and strut, thereby resulting in stronger aero-propulsion interactions relative to a conventional vehicle.

The results from the benchmark A320neo runs verified the Olympus tool's ability to solve the vehicle sizing problem by comparing the results generated by the Aviary tool and FLOPs. With the exception of the ground segments and minor differences in how the constraints were setup, the Olympus tool's results match closely with those from Aviary and FLOPS.

The SBW use case was run with three sets of aerodynamics data. The first set was a table of C_D vs. C_L, Mach and altitude, where the the lift and drag coefficients were computed on the SBW airframe that experienced the SROR wake. As such, the aero-propulsion interactions were accounted for in the aerodynamics data. This was called the "weakly-coupled" approach. In the "uncoupled" data set, the SROR was removed from the CFD model and the mission drag polar data was re-generated. As such, this set did not account for the aero-propulsion interactions. Finally, the third data set ("strongly-coupled") modeled the net streamwise and stream-normal force coefficients as a function of Mach, Reynolds number, power code, and angle of attack. These coefficients added the SROR thrust contributions resolved in the x and z directions of the wind axes to the C_L and C_D calculations. The FLOPS analysis for the SBW was run with both the uncoupled and weakly-coupled data sets, while the Olympus runs used all three data sets.

The strongly-coupled results from Olympus show a 2% higher design mission fuel burn compared to the weakly coupled results, suggesting that the decomposition of the net streamwise force into thrust and drag is not as critical for this aircraft. However, when comparing the strongly-coupled Olympus design mission fuel burn to the non-coupled Olympus mission fuel burn, a 14% difference is observed. In other words, ignoring the aero-propulsion interactions for this configuration when generating the aero data tables results in a 14% under prediction of fuel burn, which is substantial. The difference between the FLOPS results (weakly coupled relative to uncoupled) and the Olympus results for the same pair are comparable at 10% and 12% respectively. In short, the results clearly demonstrate that for concepts with stronger aero-propulsive interactions, the traditional uncoupled approach for handling the aerodynamics and propulsion disciplines separately can result in a substantial mis-estimation of the performance.

Appendices

Conventional Model Data

Table 1: Conventional Vehicle Characteristics

Specification	Variable	Value	Units	Source/Notes
Operational Empty Weight	DOWE	100,962	lbs	
Max Take-Off Weight	MTOW	174,165	lbs	APM
Max Landing Weight	WLDG	148,591	lbs	APM
Ramp Weight	GW	175,047	lbs	APM
No. Economy Passengers	NPT	138	pax	Payload-Range
No. Business Passengers	NPB	-	pax	Payload-Range
No. First Passengers	NPF	12	pax	Payload-Range
No. Flight Crew	NFLCR	2		
No. Flight Attendants	NGALC	3		14 CFR 121.391
Cargo Weight		-	n/a	See Payload-Range Tab
Container Weights	WCON	1,232	lbs	Nordisk UltraLite DPE Tare
Weight per passenger	WPPASS	190	lbs/pax	FAA AC 120-27F
Baggage Weight	BPP	30	lbs/pax	FAA AC 120-27F
Design Payload Weight	PAYLOD	33,000	lbs	Payload-Range
Design Range	DESRNG	3,420	nmi	Payload-Range
Design Fuel Load		41,085	lbs	Calculated
Taxi Out Fuel Weight		882	lbs	Derived (GW - MTOW)
Fuselage Fuel Capacity	FULFMX	-	lbs	
Wing Fuel Capacity	FULWMX	41,085.0	lbs	
Wing Loading	WSR	130.5294	psf	Derived (RAMPWT/Wing Area)
Thrust to Weight Ratio	TWR	0.309357		Derived (Thrust/RAMPWT)
Sea Level Static Thrust	THRSO	27,076.1	lbs	Engine Deck
No. Wing Mounted Engines	NEW	2		
No. Fuselage Mounted Engines	NEF	-		
Maximum Mach Number	VMMO	0.82		
Cruise Mach Number	VCMN	0.78		Payload-Range Diagram
Maximum Cruise Altitude	CH	41,000	ft	TCDS A2NM Max. Operating Alt.

Table 2: A320neo Wing Geometry

Specification	Value	Units
Root Chord Length	23.340	ft
Root Thickness	4.300	ft
Break Chord Length	12.330	ft
Break BL	20.890	ft
Break Thickness	1.510	ft
Tip Chord Length	4.400	ft
Tip BL	56.070 ft	
Tip Thickness	0.590	ft
Dihedral Angle	6	deg
APEX. FS	38.770	ft
L.E. Sweep	28	deg

Table 3: A320neo Control Surfaces Geometry

Specification	Value	Units
LE Slat 1 Area	16.716	sq ft
LE Slat 2 Area	51.481	sq ft
TE FLAP 1 Area	25.595	sq ft
TE FLAP 2 Area	25.503	sq ft
Spoiler 1 Area	27.720	sq ft
Spoiler 2 Area	45.420	sq ft
Aileron Area	15.896	sq ft

Table 4: Fuselage Geometry

Specification	Value	Units
Width	13.040	ft
Height	13.670	ft
Length	124.440	ft
Nose Tip, FS		
Nose Tip, WL	10.850	ft
Length of Passenger Compartment	92.960	ft

Table 5: A320neo Gear Geometry

Specification	Value	Units
Nose Gear Length	4.930	ft
Main Gear Length	7.44	ft

Table 6: Nacelle Geometry

Specification	Value	Units
Nacelle diameter at inlet	6.914	ft
Nacelle diameter at maximum	8.641	ft
Nacelle diameter at bypass exit	6.491	ft
Nacelle Length	11.980	ft

Table 7: Horizontal Tail Geometry

Specification	Measurement	Units
Root Chord	12.31	ft.
Tip Chord	3.97	ft.
TOC @Root	0.10	-
TOC @Tip	0.11	-
Span	40.16	ft.
L.E. Sweep	32.00	deg.
c/4 Sweep	27.52	deg.
AR	4.93	-
TR	0.323	-
S	326.90	ft^2
Dihedral	6	deg

Table 8: Wing Spanwise Geometry Definition

Specification	Measurement		
Break point location (% of semispan)	0.000	0.373	1.000
Chord % of semispan (root, break, tip)	0.413985	0.218699	0.078043
Thickness to Chord Ratio (Root, Break, Tip)	0.184233	0.122466	0.134091
Load Path Sweep (Root, Break, Tip)	19.01017	24.20486	24.20486

Table 9: Vertical Tail Geometry

Specification	Measurement	Units
Root Chord	18.241	ft.
Tip Chord	6.29	ft.
TOC @Root	0.14	-
TOC @Tip	0.11	-
Span	20.00	ft.
L.E. Sweep	40.00	deg.
c/4 Sweep	34.59	deg.
AR	1.63	-
TR	0.345	-
S	245.30	ft^2
Dihedral	0	deg

Table 10: Propulsion System Geometric Data

Specification	Value	Units
Fan Diameter	80.47	in
Nacelle max diameter	95.9	in.
Inlet Tip to fan face	41.24	in.
Fan Length	14.95	in.
Fan Exhaust Length	87.57	in.
Engine Length	127.5	in.
Engine Pod Length	166.2	in.
Total Engine Weight	6300	lb
Inlet Weight	256.54	lb
Nacelle Weight	526.72	lb
Engine Pod Weight	7083.3	lb

Table 11: Propulsion System Performance Data

Specification	Value	Units
Fuel $Flow_{TO}$	0.800416	kg/s
Fuel $Flow_{CO}$	0.661263	kg/s
Fuel $Flow_{AP}$	0.232194	kg/s
Fuel $Flow_{ID}$	0.089743	kg/s
Rated Thrust	27075.99	lbf
$Thrust_{TO}$ 100%	27075.99	lbf
$Thrust_{CO}$ 85%	23014.59	lbf
$Thrust_{AP}$ 30%	8122.80	lbf
$Thrust_{ID}$ 7%	1895.32	lbf
$TSFC_{TO}$	0.234622	lbm/hr/lbf
$TSFC_{CO}$	0.228038	lbm/hr/lbf
$TSFC_{AP}$	0.226873	lbm/hr/lbf
$TSFC_{ID}$	0.375798	lbm/hr/lbf

Bibliography

[1] O'Shea, C. A., "Next Generation Experimental Aircraft Becomes NASA's Newest X-Plane," https://www.nasa.gov/news-release/next-generation-experimental-aircraft-becomes-nasas-newest-x-plane/, June 2023. Accessed April 14, 2025. Avialable from the Internet Archive Wayback Machine.

[2] Galloway, T. L., Schairer, E. T., Bowles, J. V., Waters, M. H., Hague, D. S., MacRae, J. F., Merz, A. W., Woodbury, N. W., and Hague, A. M., "GASP- General Aviation Synthesis Program. Volume 1: Main program. Part 1: Theoretical development," Tech. Rep. NASA-CF-152303-VOL-1-PT-1, National Aeronautics and Space Administration, January 1978.

[3] McCullers, L. A., "Aircraft Configuration Optimization Including Optimized Flight Profiles," *Recent Experiences in Multidisciplinary Analysis and Optimization, Part 1*, edited by Sobieski, NASA Langley Research Center, 1984, pp. 396–412. URL `https://ntrs.nasa.gov/citations/19870002310`.

[4] Gratz, J., Jasa, J., Aretskin-Hariton, E., Moore, K., Marfatia, K., Kirk, J., and Recine, C., "Aviary: An Open-Source Multidisciplinary Design, Analysis, and Optimization Tool for Modeling Aircraft with Analytic Gradients," *AIAA AVIATION FORUM AND ASCEND 2024*, American Institute of Aeronautics and Astronautics, 2024, p. N/A.

[5] Nunez, L. S., Tai, J. C., and Mavris, D. N., *The Environmental Design Space: Modeling and Performance Updates*, American Institute of Aeronautics and Astronautics, 2021, Chap. N/A, p. N/A. https://doi.org/10.2514/6.2021-1422.

[6] Anderson, J. D., *Aircraft performance and design*, Vol. 1221, McGraw-Hill New York, 1999.

[7] Hull, D. G., *Fundamentals of Airplane Flight Mechanics*, Springer-Verlag, Berlin, Heidelberg, 2007. https://doi.org/10.1007/978-3-540-46573-7_2, URL https://doi.org/10.1007/978-3-540-46573-7_2.

[8] Ammann, P., and Offutt, J., *Introduction to software testing*, Cambridge University Press, 2008. In most cases, the slidedecks available online were referenced for this work, available from the book's companion website: https://cs.gmu.edu/ offutt/softwaretest/.

[9] Moore, D., Personal communication, ca. 2021. Face-to-face conversation.

[10] Mattingly, J. D., Heiser, W. H., and Pratt, D. T., *Aircraft engine design*, 2nd ed., AIAA Education Series, American Institute of Aeronautics and Astronautics, 2002. OCLC: 123485836.

[11] Wells, D. P., Horvath, B. L., and McCullers, L. A., "The Flight Optimization System Weights Estimation Method," Tech. Rep. NASA TM–2017–219627 Vol. I, NASA Langely Research Center, June 2017.

[12] Falck, R., Gray, J. S., Ponnapalli, 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. https://doi.org/10.21105/joss.02809, URL https://doi.org/10.21105/joss.02809.

[13] Capristan, F. M., and Welstead, J. R., "An Energy-Based Low-Order Approach for Mission Analysis of Air Vehicles in LEAPS," *AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2018, p. N/A.

[14] 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. https://doi.org/10.1007/s10107-004-0559-y.

[15] Raymer, D. P., *Aircraft design: a conceptual approach*, 2^{nd} ed., AIAA Education Series, American Institute of Aeronautics and Astronautics, 1992.

[16] Minzer, R. A., "The 1976 Standard Atmosphere and its Relationshsip to Earlier Standards," *Review of Geophysics*, Vol. 15, 1977, pp. 375–184.

[17] Digital Dutch, "1976 Standard Atmosphere Calculator," , 1999. Accessed April 19, 2025.

[18] Feagin, R. C., and Morrison, W. D. J., "Delta method, an empirical drag buildup technique," Tech. Rep. 151971, Lockheed-California Company, December 1978.

[19] Lytle, J. K., "The numerical propulsion system simulation: an overview," Tech. Rep. NASA TM–2000–209915, NASA Glenn Research Center, June 2000.

[20] "NIST Digital Library of Mathematical Functions," , N/A. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[21] Aviary Team, "Propulsion," https://openmdao.github.io/Aviary/user_guide/propulsion.html, 2023. Accessed April 19, 2025.

[22] Mavris, D. N., Crossley, W., Tai, J., DeLaurentis, D., Pfaender, H., Hassan, M., Dassage, T., Fazzini, T., Hong, R., Iyengar, N., Sampaio, B., Tran, K., Baltman, E., Jain, S., Ogunsina, K., and Chao, H., "ASCENT Project 010 Aircraft Technology Modeling and Assessment," Tech. rep., Georgia Institute of Technology and Purdue University, 2019.

[23] Aviary Team, "Comparison to FLOPS," https://openmdao.github.io/Aviary/misc_resources/comparison_to_flops.html, 2023. Accessed April 26, 2025.

[24] Aviary Team, "Core Propulsion Subsystem: Pre-Mission Analysis," , 2023. URL ttps://openmdao.github.io/Aviary/theory_guide/propulsion.html#pre-mission-analysis, accessed April 26, 2025.

[25] Schutte, J. S., """Simultaneous multi-design point approach to gas turbine on-design cycle analysis for aircraft engines," Ph.D. thesis, Georgia Institute of Technology, May 2009.

[26] Converse, G. L., and Giffin, R. G., "Extended parametric representation of compressor fans and turbines. Volume 1: CMGEN user's manual," Tech. Rep. NASA CR-174645 Vol. I, General Electric Co., March 1984.

[27] Tong, M. T., and Naylor, B. A., "An object-oriented computer code for aircraft weight estimation," Tech. Rep. NASA TM-2009-215656, NASA Glenn Research Center, December 2009.

[28] Wedge, H. R., Bonet, J., Hollowell, S., Kutzmann, A., Nelson, C., Nelson, C., Baughcum, S., Britt, R. T., Miller, G., and Tai, J., "N+2 Supersonic Concept Development and Systems Integration," Tech. rep., The Boeing Co., August 2010.

[29] Lee, C., Ahuja, J., Decker, K., Gladin, J., Harrison, E., Patel, S., Perron, C., Tarazi, S., Zhu, S., Eggert, C., Burrell, A., Chen, M., Crawford, S., Koerschner, M., Mufti, B., der Linden, J. V., Vegesna, A., Gandhi, S., D'Cruz, R., Moore, S., Krishnan, K. N., and Mavris, D. N., "ASCENT Project 050 Over-Wing Engine Placement Evaluation Final Report," Tech. rep., Georgia Institute of Technology, 2024.

[30] Gipson, L., "Sustainable Flight National Partnership," , February 2022. Available on Internet Archive Wayback Machine. Accessed March 03, 2025.

[31] Airbus SAS, "Aircraft Characteristics," , N/A. URL https://aircraft.airbus.com/en/customer-care/fleet-wide-care/airport-operations-and-aircraft-characteristics/aircraft-characteristics, accessed April 15, 2025.

[32] International Civil Aviation Organization, "ICAO Aircraft Engine Emissions Databank," , N/A.

[33] Cai, Y., Rajaram, D., and Mavris, D. N., *Multi-mission Multi-objective Optimization in Commercial Aircraft Conceptual Design*, American Institute of Aeronautics and Astronautics, 2019, Chap. N/A, p. N/A. https://doi.org/10.2514/6.2019-3577, URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-3577.

[34] Harrison, N. A., Hoffman, K., Lazzara, D. S., Reichenbach, E. Y., Sclafani, A. J., and Droney, C. K., "Subsonic Ultra Green Aircraft Research: Phase IV Final Report – Volume I Mach 0.80 Transonic Truss-Braced Wing High-Speed Design Report," Tech. Rep. NASA CR–20220016017 Vol. I, he Boeing Co., October 2023.

[35] Han, Z.-H., and Görtz, S., "Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling," *AIAA Journal*, Vol. 50, No. 9, 2012, pp. 1885–1896. https://doi.org/10.2514/1.J051354, URL http://arc.aiaa.org/doi/10.2514/1.J051354.

[36] Kiviaho, J. F., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2237–2243. https://doi.org/10.2514/1.J057318, URL https://doi.org/10.2514/1.J057318.

[37] Aviary Team, "Source code for aviary.interface.methods_for_level2," , 2025. URL https://github.com/OpenMDAO/Aviary/blob/62c20687ea38e59afc324ad9b9045ca46e9bcc53/aviary/interface/methods_for_level2.py#L1487-L1578, accessed April 26, 2025.

[38] Vigerske, S., Tasseff, B., and Frey, J., "Ipopt Options," , 2024. URL https://coin-or.github.io/Ipopt/OPTIONS.html, accessed: April 04, 2025.