**INTERNATIONAL UNIVERSITY LIAISON INDONESIA (IULI)**

BACHELOR'S THESIS

---

# PRELIMINARY SIZING OF HOMELAND DEFENSE INTERCEPTOR AIRCRAFT USING AVIARY

---

By

**Samuel Theofilus Tamonob**
11202101008

Presented to the Faculty of Engineering and Life Sciences
In Partial Fulfillment Of the Requirements for the Degree of

BACHELOR'S DEGREE
In
AVIATION ENGINEERING

FACULTY OF ENGINEERING AND LIFE SCIENCES

BSD City 15345
Indonesia
August 9, 2025

# APPROVAL PAGE

PRELIMINARY SIZING OF HOMELAND DEFENSE INTERCEPTOR AIRCRAFT USING AVIARY

SAMUEL THEOFILUS TAMONOB
11202101008
Presented to the Faculty of Engineering
In Partial Fulfillment of the Requirements for the Degree of

BACHELOR'S DEGREE
In
AVIATION ENGINEERING

FACULTY OF ENGINEERING AND LIFE SCIENCES

Triwanto Simanjuntak, PhD ____________
Thesis Advisor Date

Dr. Eng. Ressa Octavianty ____________
Thesis Co-Advisor Date

Dipl.-Ing. Sentot Wahjoe Georitno, M.Si. ____________
Dean of Faculty of Engineering and Life Sciences Date

# EXAMINERS APPROVAL PAGE

M. Ilham Adhynugraha, PhD
Examiner 1, BRIN (National Research and Innovation Agency) Date

Dewi Habsari Budiarti, PhD
Examiner 2, BRIN (National Research and Innovation Agency) Date

# STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgment is made in the thesis.

| | |
|---|---|
| Samuel Theofilus Tamonob | ____________________ |
| Student | Date |

# ABSTRACT

Preliminary Sizing of Homeland Defense Interceptor Aircraft using Aviary

by

Samuel Theofilus Tamonob

Triwanto Simanjuntak, PhD, Advisor

Dr. Eng. Ressa Octavianty, Co-Advisor

The increasing threat from aerial incursions such as cruise missiles, stealth aircraft, and unmanned systems demands interceptor specifically designed for homeland defense. This study presents the preliminary sizing of an unmanned Homeland Defense Interceptor (HDI) using the open-source AVIARY framework, which integrates parametric design and mission analysis within OpenMDAO. Based on the 2025 AIAA Undergraduate Design RFP, the HDI is evaluated across three mission profiles: Defensive Counter-Air Patrol, Point Defense Interception, and Intercept/Escort, using benchmarking data from the F-22 Raptor, Saab JAS 39C Gripen, MiG-29 Fulcrum, and Bayraktar Kzlelma. Through iterative simulation and constraint analysis, the HDI has a Gross Takeoff Weight $(GTOW)$ of 36,066 lb (16,359 Kg), fuel weight of 14,683 lb (6660.1 Kg), a wing area of 486.61 ft$^2$ (45.17 m$^2$), Aspect Ratio of 3.98, Horizontal Tail Area of 154.73 ft$^2$ (14.37 m$^2$), Vertical Tail Area of 53.98 ft$^2$ (5.01 m$^2$) , Wing Loading of 72.04 lb/ft$^2$ (351.73 kg/m$^2$) and a Thrust-to-Weight ratio of 0.98, while meeting key loiter, dash, and interception requirements. The results demonstrate AVIARYs effectiveness as a multidisciplinary conceptual design tool and confirm the feasibility of a tailored unmanned interceptor within defined performance.

Keyword: *Conceptual Design, Aircraft sizing, AVIARY, Interceptor, Homeland Defense, OpenMDAO*

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest and most sincere gratitude to my Lord and Savior, Jesus Christ. His unwavering love, grace, and guidance have been the foundation of my strength throughout this journey. In moments of uncertainty, He provided clarity; in times of challenge, He gave me perseverance; and in every success, He reminded me of the importance of humility and gratitude. This accomplishment would not have been possible without His divine presence and continuous blessings.

I would also like to extend my heartfelt thanks to my supervisors, Triwanto Simanjuntak, PhD., and Dr. Eng. Ressa Octavianty, for their invaluable support, expert guidance, and continuous encouragement throughout the course of this work. Their insights and feedback have been instrumental in shaping the direction and quality of my research.

Special appreciation goes to AVIARY development communities. Their open-source contributions, documentation, and collaborative spirit have played a crucial role in the progress of this project. I am grateful for the opportunity to learn from such a dedicated and innovative group of professionals.

To my family, thank you for your unconditional love, patience, and belief in me. Your support has been a constant source of motivation, and your sacrifices have allowed me to focus and persist in my academic endeavors.

Finally, to my friends and peers, thank you for the camaraderie, shared struggles, and words of encouragement that made this journey not only possible but also meaningful and fulfilling. Your presence has been a source of both strength and joy.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **AIAA** | **A**merican **I**nstitute of **A**eronautics and **A**stronautics |
| **NASA** | **N**ational **A**eronautics and **S**pace **A**dministration |
| **HDI** | **H**omeland **D**efense **I**nterceptor |
| **RFP** | **R**equest **F**or **P**roposal |
| **UCAV** | **U**nmanned **C**ombat **A**erial **V**ehicle |
| **NGAD** | **N**ext **G**eneration **A**ir **D**ominance |
| **IADS** | **I**ntegrated **A**ir **D**efense **S**ystem |
| **SAMs** | **S**urface to **A**ir **M**issiles |
| **GTOW** | **G**ross **T**akeoff **W**eight |
| **CG** | **C**entral of **G**ravity |
| **EOM** | **E**quation **O**f **M**otion |
| **SFC** | **S**pecific **F**uel **C**onsumption |
| **AR** | **A**spect **R**atio |
| **nmi** | **n**autical **m**iles |
| **APU** | **A**uxiliary **P**ower **U**nit |
| **RCS** | **R**adar **C**ross **S**ection |
| **ICNIA** | **I**ntegrated **C**ommunications **N**avigation and **I**dentification **A**vionics |
| **MFDs** | **M**ulti **F**unction **D**isplays |
| **OBOGS** | **O**n **B**oard **O**xygen **G**enerating **S**ystem |
| **OBIGGS** | **O**n **B**oard **I**nert **G**as **G**enerating **S**ystem |
| **AIM** | **A**eronautical **I**nformation **M**anual |
| **FAA** | **F**ederal **A**viation **A**dministration |
| **EASA** | **E**uropean Union **A**viation **S**afety **A**gency |
| **FLOPS** | **Fl**ight **O**ptimization **S**ystem |
| **GASP** | **G**eneral **A**viation **S**ynthesis **P**rogram |
| **OpenMDAO** | **O**pen Source **M**ultidisciplinary **D**esign **A**nalysis and **O**ptimization |
| **MDO** | **M**ultidisciplinary **D**esign **O**ptimization |

| | |
|---|---|
| **IPOPT** | **I**nterior **P**oint **O**ptimizer |
| **SNOPT** | **S**parse **N**onlinear **O**ptimizer |
| **SLSQP** | **S**equential **L**east **S**quares **P**rogramming |
| **CFD** | **C**omputational **F**luid **D**ynamics |
| **DATCOM** | **D**ata **C**omendium |
| **DCAP** | **D**efense **C**ombat **A**ir **P**atrol |
| **PDI** | **P**oint **D**efense **I**ntercept |

*Dedicated to my parents*

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

The increasing instability of world political and military developments points to the necessity of securing national airspace. The United States today could be targeted with a variety of airborne weapons including hijacked commercial jets or cruise missiles or drones that are difficult to stop. This requires a quick-reaction air defense system that can identify, intercept and negate such threats before they strike critical infrastructure or civilian targets.

Current air superiority fighters such as the *F-22 Raptor* and *F-35 Lightning II*, while technologically advanced, were primarily developed for high-end conflict scenarios and are associated with high operational costs and limited production quantities. These characteristics make them less ideal for widespread deployment in a homeland defense role where affordability, operational simplicity, and high sortie rates are crucial.

(A) F-22 Raptor **LockheedMartinF222025**

(B) F-35 Lightning II **LockheedMartinF352025**

To address this capability gap, the AIAA Undergraduate Aircraft Design Team RFP 2025 calls for the development of a new class of interceptor aircraft which is the *Homeland Defense Interceptor (HDI)*. The HDI must be unmanned, highly maneuverable, and optimized for three specific mission profiles:

1. Defensive Counter-Air (DCA) Patrol, involving loitering near high-value assets to intercept threats.
2. Point Defense Interception, for rapid engagement of local airborne threats.
3. Intercept/Escort, entailing high-speed pursuit or protection of allied aircraft.

This research utilizes AVIARY, an open-source tool developed by NASA on the OpenMDAO framework. AVIARY supports multidisciplinary aircraft sizing through parametric inputs and integrated mission analysis capabilities, making it highly suitable for early-phase design studies like this one.

## 1.2 Problem Statement

The conceptual design of modern military aircraft faces increasing complexity, driven by the demanding interplay of advanced performance metrics and stringent operational effectiveness requirements. Traditional sizing methods are often insufficient in this environment due to:

- Siloed by discipline, requiring manual coordination of aerodynamic, propulsion, and weight estimates.
- Time-consuming and inefficient for rapid design iterations.
- Difficulties in adapting swiftly to evolving mission constraints or integrating technological updates.

This research aims to address these challenges by applying AVIARY as a design tool for preliminary sizing. The tools integration of weight, performance, mission, and geometry estimation enables a more streamlined and data-driven approach to concept development for the HDI.

## 1.3 Research Objectives

The primary objectives of this research are as follows:

1. To interpret the mission profiles and technical constraints defined in the AIAA 2025 RFP.
2. To use the AVIARY framework to develop a preliminary aircraft sizing model that meets those mission requirements.
3. To evaluate aircraft performance across three mission types: DCA patrol, point defense, and intercept/escort.
4. To determine key design parameters such as takeoff weight, wing area, thrust requirement, and fuel load.
5. To benchmark the HDI design against existing aircraft, including both manned and unmanned interceptors.

## 1.4 Research Scope and Limitation

### 1.4.1 Scope

This research focuses exclusively on the preliminary sizing of a homeland defense interceptor aircraft. The specific areas of focus include:

- Development of a parametric sizing model using AVIARY.
- Inclusion of three distinct mission profiles with performance constraints.
- Estimation of key design parameters: gross weight, wing area, aspect ratio, thrust-to-weight ratio, and fuel capacity.
- Use of empirical models from Raymer and benchmark aircraft data to provide sizing inputs.
- Iterative mission-constrained design refinement using AVIARYs optimization tools.

### 1.4.2 Limitations

Despite the comprehensive approach, this study is limited by several factors:

- AVIARY does not currently support supersonic aerodynamic modeling; thus, any supersonic effects are excluded.
- No structural modeling or stress analysis is performed at this stage.

- The study does not account for stealth characteristics, radar cross-section, or ECM systems integration.
- Engine modeling is simplified to thrust and fuel flow estimates without full cycle thermodynamics.

## 1.5 Significance of the Study

This study contributes to the field of preliminary aircraft design in the following ways:

- Demonstrates AVIARY, the application of a modern, open-source, multidisciplinary tool for aircraft conceptual sizing.
- The findings can inform future research and development efforts in aircraft design and aerospace engineering, particularly in the context of homeland security.

# CHAPTER 2
# LITERATURE REVIEW

## 2.1 Interceptor Aircraft Concept

### 2.1.1 Development of Interceptor Aircraft

The development of interceptors aircraft have been a subject of ongoing research for decades. Since the Cold War era, aircraft interception has evolved with technological advancements in aerodynamics, propulsion systems, avionics, and missile technology **InterceptorAircraft2025**. Earlier interceptor designs, such as the MiG-21 and the F-106 Delta Dart as shown in Figure 2.1a, were focused on high-speed interception, relying heavily on supersonic performance and missile armament. Over the years, newer designs have incorporated advanced stealth technologies, agility, and multi-role capabilities, exemplified by modern aircraft such as the F-22 Raptor 1.1a and Mikoyan MiG-29 Fulcrum 2.1b.


(A) Convair F-106 Delta Dart **ConvairF106Delta2025**

(B) MiG-29 Fulcrum **MikoyanMiG292025**

Homeland aircraft interceptors are specialized military aircraft designed to respond rapidly to aerial threats within a nation's airspace. The primary mission of an interceptor is to engage and neutralize incoming aircraft or missiles, often before they can reach critical targets. To meet these operational needs, interceptors must excel in high-speed flight, fast climb rates, and superior maneuverability.

These requirements present unique challenges for aircraft designers, who must also balance the interceptors range, payload capacity, and fuel efficiency.

### 2.1.2 Future Interceptor Aircraft

The pursuit of advanced interceptor aircraft has become a major focus in modern aerospace defense discussions. Initiatives such as the U.S. Next Generation Air Dominance (NGAD) and Russias MiG-41 project reflect a growing trend toward aircraft capable of hypersonic flight, enhanced stealth, and sophisticated electronic warfare capabilities. These platforms are being designed to counter both conventional aerial threatslike fighter jets and bombersand newer challenges such as hypersonic missile systems.

At the same time, the role of Unmanned Combat Aerial Vehicles (UCAVs) in air defense is gaining prominence. UCAVs present several advantages over manned aircraft, including reduced operational risk, enhanced agility, and lower life-cycle costs. Modern demonstrators such as the Kratos XQ-58A Valkyrie shown in Figure 2.2b, Northrop Grumman X-47B, and Bayraktar Kzlelma shown in Figure 2.2a exemplify the growing feasibility of unmanned interceptors capable of fulfilling high-speed, high-threat missions. Furthermore, with the integration of artificial intelligence and real-time data processing, UCAVs are increasingly capable of operating autonomously in dynamic combat environments. These trends suggest that future interceptor aircraft will likely be unmanned, highly networked, and optimized using multidisciplinary design frameworks to meet evolving defense needs with greater speed, efficiency, and adaptability.

(A) Bayraktar Kizilelma **BayraktarKizilelma2025**


(B) Kratos XQ58A Valkyrie **KratosXQ58Valkyrie2025**

## 2.2 Homeland Defense and Air Threats

### 2.2.1 Overview of Homeland Defense Systems

Homeland defense systems are developed to safeguard a nations airspace against a wide range of aerial threats, including hostile aircraft, unmanned aerial vehicles (UAVs), and missiles. These systems operate as part of a comprehensive Integrated Air Defense System (IADS), which merges early warning mechanisms, ground-launched missile systems, and interceptor aircraft.

A key feature of homeland defense is the implementation of a multi-layered protection strategy, which utilizes a combination of airborne and ground-based sensors, command and control infrastructure, and interceptor platforms. This layered approach allows threats to be detected and neutralized at various distances and altitudes. Components of this system typically include:

- Long-range surveillance system, such as over-the-horizon radar systems.
- Medium-range protection via surface-to-air missile (SAM) systems shown in Figure 2.3 like the Patriot or S-400.
- Short-range, rapid-response interceptors to engage threats that manage to breach the outer layers of defense.

Command and Control (C2) systems are vital to ensuring effective coordination within homeland defense operations. These systems support quick decision-making and seamless integration across different defensive layers. By leveraging data fusion technologies, C2 systems can combine inputs from multiple sourcesincluding ground radar stations, satellites, and airborne early warning platforms like AWACS to deliver real-time situational awareness and guide response actions effectively.

### 2.2.2 Emerging Aerial Threats

The nature of aerial threats is changing rapidly, prompting the need for homeland defense strategies to evolve accordingly. Todays threats are more varied and sophisticated, necessitating that interceptor technologies advance to meet these new challenges. Key emerging threats include:

FIGURE 2.3: surface-to-air missiles (SAMs) **SurfacetoairMissile2025**

- **Hypersonic Missiles:**
  Missiles like Russias Avangard and Chinas DF-ZF shown in Figure 2.4 can exceed Mach 5 and are capable of executing unpredictable maneuvers. Their extreme velocity and agility make them exceptionally hard to detect and intercept using conventional air defense systems.

FIGURE 2.4: China Hypersonic Missile DF-ZF **DFZF2025**

- **Stealth Aircraft:**
  Aircraft such as the Chengdu J-20 and Sukhoi Su-57 shown in Figure 2.5a and 1.1b are designed with stealth technologies that reduce radar visibility, making them difficult to detect and target. As stealth capabilities continue to improve, interceptors will require enhanced sensor fusion, as well as integration into network-centric warfare systems, to successfully engage these aircraft.
- **Unmanned Aerial Vehicles (UAVs):**
  Both reconnaissance and armed UAVs, including Turkeys Bayraktar TB2 and the U.S. MQ-9 Reaper shown in Figure 2.6, are being increasingly utilized

(A) Sukhoi Su-57 **SukhoiSu572025**

(B) Chengdu J-20 **ChengduJ202025**

in modern combat. These drones often fly at low altitudes and may operate in large groups, potentially overwhelming traditional defense systems. Swarm tactics where multiple UAVs coordinate attacks simultaneously pose a particularly complex challenge.

FIGURE 2.6: MQ-9 Reaper **GeneralAtomicsMQ92025**

- **Cruise Missiles:**
  Weapons such as the U.S. Tomahawk shown in Figure 2.7 and Russias Kalibr remain serious threats due to their low-altitude flight paths and ability to avoid radar through terrain-following methods. Their precision over long distances demands that interceptor systems be equipped with advanced guidance and long-range engagement capabilities.

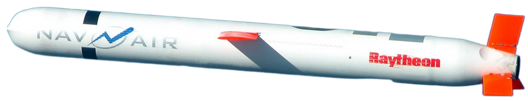



FIGURE 2.7: U.S. Tomahawk **TomahawkMissileFamily2025**

## 2.3 Design Requirements for Homeland Defense Interceptor Aircraft

The Homeland Defense Interceptor (HDI) is being developed in response to the design goals outlined in the 2024-2025 AIAA Undergraduate Design Team Project Request for Proposal **aiaaRFPHDIAircraft**. This document lays out the key missions, performance expectations, and operational requirements that shape the foundation of the HDIs design. At its core, the RFP emphasizes the need for a mission-tailored aircraft that meets strict operational standards while filling a critical gap in the U.S. homeland air defense system.

Theres a clear demand for a new kind of interceptorone thats affordable, yet capable of responding to a wide variety of aerial threats, from hijacked passenger planes to autonomous cruise missiles. Current high-end fighters like the F-22 and F-35, while powerful, are simply too expensive and not well-suited for mass production or large-scale homeland defense missions.

1. **Mission Requirement**
   The RFP specifies two primary design missions: Defensive Counter-Air (DCA) Patrol and Point Defense Interception. Additionally, an Intercept/Escort mission is considered **aiaaRFPHDIAircraft**. The mission profiles are as follows:

   - Defensive Counter-Air (DCA) Patrol Mission
     This mission is designed to safeguard friendly airspace by keeping a watchful presence over a designated area and staying ready to respond to any potential threats in the sky. The aircraft essentially acts as both a deterrent and a first line of defense against any unauthorized intrusions.
     To effectively carry out this role, the aircraft needs to remain on station for up to four hours during a combat air patrol. This requirement pushes the endurance well beyond what most modern supersonic fighters are typically built to handle as shown im Figure 2.8.

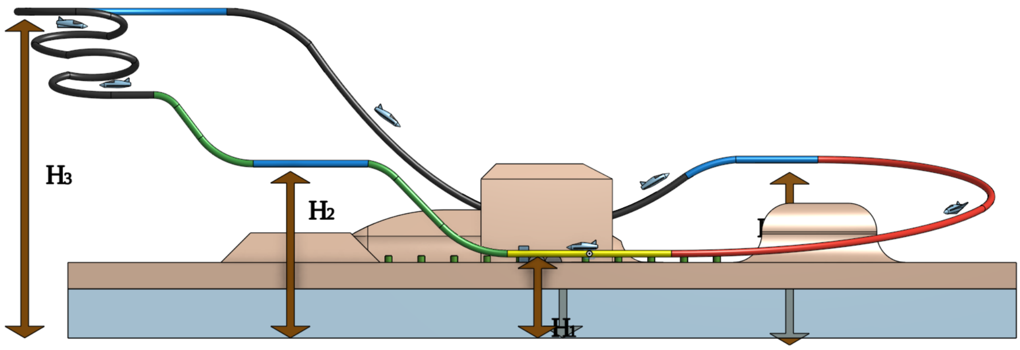



FIGURE 2.8: Mission Profile: Defensive Counter-Air Patrol Mission

TABLE 2.1: DCAP - Mission Segments Description

| Phase | Description |
|---|---|
| 1 | Take-off and acceleration |
| 2 | Climb from sea level to optimum cruise altitude |
| 3 | Cruise out 300 nm at optimum speed and altitude |
| 4 | Combat air patrol for 4 hours at best loiter speed at 35,000 ft |
| 5 | Dash 100 nm at maximum speed at 35,000 ft |
| 6 | Combat maneuvers at 35,000 ft, maximum thrust and fuel flow. Fire all missiles and retain gun ammunition |
| 7 | Climb/accelerate to optimum speed and altitude |
| 8 | Cruise back 400 nm at optimum speed and altitude |
| 9 | Descend to sea level |
| 10 | Reserves: Fuel for 30 minutes at sea level at maximum endurance speed |

- Point Defense Interception Mission

  This mission focuses on delivering a rapid response to urgent, localized aerial threats such as incoming missiles or enemy aircraft targeting key infrastructure or high-value assets.

  Unlike the extended patrol mission, this operation does not involve loitering. Instead, it emphasizes quick reaction times and longer dash segments to reach the threat area as fast as possible. Despite these differences in timing and distance, the aircraft still performs similar combat maneuvers and carries the same weapon load. Because the mission profile closely mirrors that of a Defensive Counter-Air (DCA) mission, performance parameters like speed (Mach number) and altitude remain largely unchanged as shown in Figure 2.9.

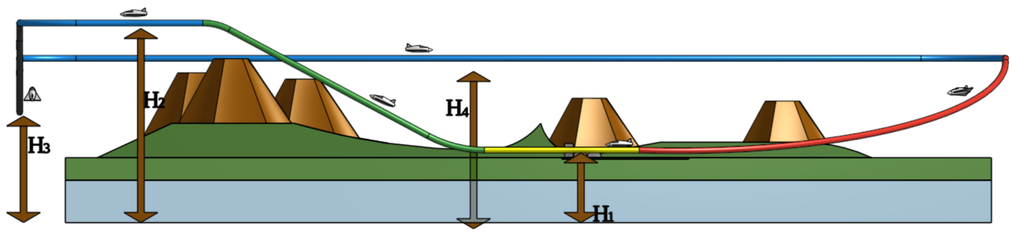



FIGURE 2.9: Mission Profile: Point Defense Intercept Mission

TABLE 2.2: PDI - Mission Segments Description

| Phase | Description |
|---|---|
| 1 | Take-off and acceleration |
| 2 | Climb from sea level to 35,000 ft and accelerate to maximum speed |
| 3 | Dash 200 nm at maximum speed at 35,000 ft |
| 4 | Combat maneuvers at 35,000 ft, maximum thrust and fuel flow. Fire all missiles and retain gun ammunition. |
| 5 | Climb/accelerate to optimum speed and altitude |
| 6 | Cruise back 200 nm at optimum speed and altitude |
| 7 | Descend to sea level |
| 8 | Reserves: Fuel for 30 minutes at sea level at maximum endurance speed |

- Intercept/Escort Mission

  This mission involves escorting friendly aircraftsuch as bombers or reconnaissance platforms through contested or potentially hostile airspace to ensure they can complete their objectives and return safely.

  Unlike the previous missions, this one is not bound by the strict performance criteria outlined in the original request for proposal. Instead, it was designed to evaluate the interceptors combat radius and its ability to operate effectively at lower speedsan important factor when supporting slower-moving or civilian aircraft during escort operations as shown in Figure 2.10.

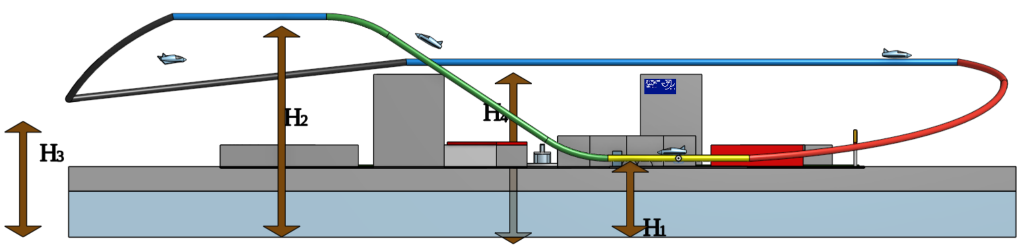



FIGURE 2.10: Mission Profile: Escort Mission

TABLE 2.3: Escort - Mission Segments Description

| Phase | Description |
|---|---|
| 1 | Take-off and acceleration |
| 2 | Climb from sea level to 35,000 ft and accelerate to maximum speed |
| 3 | Dash out at maximum speed at 35,000 ft |
| 4 | Escort for 300 nm at minimum practical airspeed while retaining all weapons |
| 5 | Climb/accelerate to optimum speed and altitude |
| 6 | Cruise back at optimum speed and altitude |
| 7 | Descend to sea level |
| 8 | Reserves: Fuel for 30 minutes at sea level at maximum endurance speed |

2. **Design Constraints**
3. **Component List**

   The total payload of a homeland defense interceptor aircraft plays a crucial role in determining the operational capabilities and mission flexibility. Payload refers to the combined weight of all mission-specific equipment and systems carried on board, excluding the basic aircraft structure, fuel, and propulsion systems. For a homeland defense interceptor, the payload is typically composed of weapons, sensors, communication systems, and additional mission-related equipment as shown in Table 2.5. The payload that consist of air-to-air weapons have the total weight of 802 lbm.
4. **Other Requirements**
   - Standard day conditions with no wind.
   - Fuel type: JP-8 or Jet-A, with self-sealing fuel tanks.

TABLE 2.4: List of Design Constraints

| **Mission Performance** | **Performance Requirement** |
|---|---|
| Intercept Radius | 200 nm |
| DCA Mission CAP endurance | At 300 nm, duration = 4 Hours radius |
| Maximum Mach Number | At 35,000 ft: M = 1.6 |
| **Maneuver Weight (50% Internal Fuel)** | |
| 1-g Specific Excess Power Military Thrust | 0.9M/Sea Level = 200 ft/s, 0.9M/15,000 ft = 50 ft/s |
| 1-g Specific Excess Power Maximum Thrust | 0.9M/Sea Level = 700 ft/s, 0.9M/15,000 ft = 400 ft/s |
| 5-g Specific Excess Power Maximum Thrust | 0.9M/Sea Level = 300 ft/s, 0.9M/15,000 ft = 50 ft/s |
| Sustained Load Factor Maximum Thrust | 0.9M/15,000 ft |
| Maximum Instantaneous Turn Rate | At 35,000 ft = 18.0 deg/s |

- Operation from NATO-standard runways (8,000 ft) and austere bases.

## 2.4 Aircraft Design

### 2.4.1 Aircraft Design Process

The development of an aircraft is a structured process that evolves through several critical stages to ensure the final product meets performance, safety, and mission requirements. The general sequence of the aircraft design process are illustrated in Raymer's book **raymerAircraftDesignConceptual2006a** that consists of the following phases:

1. **Design and Mission Requirement**
   The first step involves defining the mission profile and operational roles of the aircraft. These requirements are established based on military or commercial needs and include parameters such as range, payload, speed,

TABLE 2.5: Weight Payload Component List

| No. | Items | Weight (lb / kg) | Volume (ft$^3$ / m$^3$) |
|---|---|---|---|
| | **Avionics** | | |
| 1 | ICNIA | 100 / 45.36 | 3 / 0.085 |
| 2 | 3 X MFDs | 20 / 9.07 | 1.5 / 0.042 |
| 3 | Head-up Display | 35 / 15.88 | 1.6 / 0.045 |
| 4 | Data Bus | 10 / 4.54 | 0.5 / 0.014 |
| 5 | INEWS | 100 / 45.36 | 3 / 0.085 |
| | **Flight and Propulsion Control System** | | |
| 6 | Vehicle Management System | 50 / 22.68 | 1 / 0.028 |
| | **Fire Control Systems** | | |
| 7 | IRSTS | 50 / 22.68 | 1 / 0.028 |
| 8 | Active Array Radar | 450 / 204.12 | 6 / 0.170 |
| | **System and Equipment** | | |
| 9 | Electrical Systems (2 Engines) | 300 / 136.08 | 4 / 0.113 |
| 10 | Auxiliary Power Unit (APU) | 100 / 45.36 | 1 / 0.028 |
| 11 | Ejection Seat | 160 / 72.57 | 2 / 0.057 |
| 12 | OBOGS | 35 / 15.88 | 1 / 0.028 |
| 13 | OBIGGS | 35 / 15.88 | 1 / 0.028 |
| | **Air-to-Air Weapons** | | |
| 14 | AIM - 120 AMRAAM | 327 / 148.32 | |
| 15 | M61A1 20 MM Cannon | 275 / 124.74 | |
| 16 | Ammunition feed system (500 rounds) | 200 / 90.72 | |
| | **Total Weight:** | **2,347 / 1,064.96** | |

altitude, maneuverability, and endurance. This phase sets the foundation for all subsequent design decisions by identifying the critical design drivers.

2. **Conceptual Design**
   In this phase, various design concepts are explored and evaluated. Designers develop multiple configurations and perform preliminary sizing calculations to determine approximate weight, wing area, thrust requirements, and fuel capacity. Analytical tools and historical data are often used to evaluate the feasibility of different design options. The goal is to identify the most promising configuration that satisfies the mission requirements.

3. **Preliminary Design**
   The selected concept is further refined in this phase. Detailed analyses are conducted in aerodynamics, propulsion, structures, flight mechanics, and systems integration. At this stage, design trade-offs are performed to optimize the aircraft performance, and more accurate estimations of weight, balance, and system layout are developed. Computational tools such as OpenMDAO and AVIARY may be employed to simulate and analyze mission performance and subsystem interactions.
4. **Detail Design**
   This phase involves finalizing the design for production. Every aircraft component is fully defined, including dimensions, materials, and manufacturing processes. Engineering drawings, system schematics, and structural details are produced. The integration of all subsystemssuch as avionics, hydraulics, and landing gearis completed to ensure compatibility and functionality.
5. **Prototype, Testing, and Certification**
   A prototype of the aircraft is constructed and subjected to a series of ground and flight tests to verify design performance and safety compliance. These tests are essential for validating the design assumptions and identifying potential issues. Following successful testing, the aircraft undergoes a certification process to ensure it meets regulatory and safety standards set by aviation authorities such as the FAA or EASA.

This systematic process as shown in Figure 2.11 ensures that the final aircraft design is not only capable of meeting the specified mission objectives but also safe, efficient, and economically viable.

### 2.4.2 Preliminary Sizing Methodologies

Preliminary aircraft sizing is a critical phase in the design process that establishes the basic parameters of an aircraft to meet specified performance requirements. Initial estimates from empirical methods inform analytical calculations, which in turn reveal inconsistencies or areas for optimization. This necessitates revisiting initial assumptions and refining design parameters until a consistent and feasible

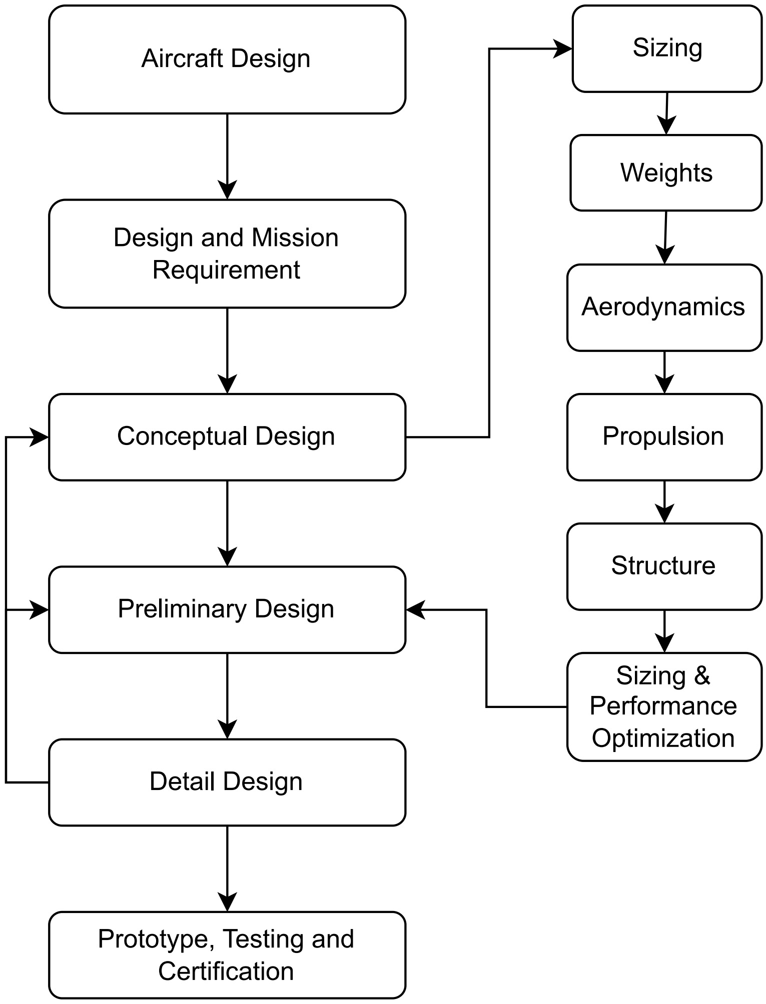



FIGURE 2.11: Aircraft Design Phases

solution that satisfies all requirements is achieved. The interplay between various disciplines aerodynamics, propulsion, structures, and performance means that changes in one area invariably affect others, demanding a continuous feedback loop. This iterative and interconnected nature of aircraft design, spanning from initial requirements through detailed design and into operations, is famously visualized by the design spiral, as illustrated in Figure 2.12. This model emphasizes how each stage of design feeds back into and refines earlier stages, underscoring the necessity for a holistic approach.

1. Mission Definition and Requirements

The first step in any sizing methodology is defining the mission profile of the aircraft. For an interceptor, the mission typically includes:

- Scramble and intercept: The ability to quickly launch and climb to a high altitude to engage a hostile aircraft.
- High-speed engagement: The capacity to reach and sustain supersonic speeds during combat.
- Endurance and loiter: The ability to loiter for extended periods, especially for homeland defense missions where constant patrol may be required.
- Return to base: After mission completion, sufficient fuel for safe return.

2. Initial Weight Estimation
   A commonly used method for estimating aircraft weight is Raymers approach **raymerAircraftDesignConceptual2006a**, which divides weight estimation into statistical regression models based on historical data for similar aircraft. This method provides initial guesses for empty weight fractions, which are iteratively refined during the design process. The fuel fraction for military aircraft like interceptors is typically higher due to the need for high thrust and supersonic flight.

   (a) Takeoff Gross Weight
   The design process starts by calculating to estimate the Take-off Weight $W_O$ aircraft. The $W_O$ is a very important design parameter as it sizes the takeoff weight The $W_O$ equation used in their analysis, based on input geometry and mission data, provides a starting point for estimating fuel and engine requirements.

$$\begin{aligned} W_0 &= W_{\text{crew}} + W_{\text{payload}} + W_{\text{empty}} + W_{\text{fuel}} \\ &= \frac{W_{\text{crew}} + W_{\text{payload}}}{1 - \left( \frac{W_{\text{empty}}}{W_0} + \frac{W_{\text{fuel}}}{W_0} \right)} \end{aligned} \tag{2.1}$$

   where:

   - $W_{\text{fuel}}$: Fuel weight.

- $W_{\text{payload}}$: Weight of passengers or equipment.
- $W_{\text{empty}}$: Empty weight of the aircraft.

(b) Fuel weight fraction

$$\frac{W_f}{W_i} = 1.06 \times (1 - \frac{W_x}{W_i}) \tag{2.2}$$

where:

- $\frac{W_f}{W_i}$: Fuel Fraction for each mission.
- $\frac{W_x}{W_i}$: Weight fraction for each mission.

This equation provides a baseline for sizing interceptors, where the objective is to minimize the $GTOW$ while ensuring sufficient fuel for rapid response is crucial.

3. Wing Loading and Thrust-to-weight Ratio

An essential factor in the performance of supersonic aircraft is achieving an optimal balance between wing loading $(W/S)$ and thrust-to-weight ratio $(T/W)$. For supersonic interceptorsdesigned for rapid response and high-speed agilitythese parameters must be carefully optimized. Lower wing loading contributes to better lift characteristics at lower speeds, whereas a higher thrust-to-weight ratio enables fast acceleration and the ability to sustain supersonic speeds. The following equation of wing loading $(W/S)$ and thrust-to-weight ratio $(T/W)$:

(a) Wing Loading:

$$\frac{W}{S} \tag{2.3}$$

where:

- $W$: Weight of the wing.
- $S$: Area of the wing.

(b) Thrust-to-weight-Ratio:

$$\frac{T}{W} = \frac{qC_D}{W/S} \tag{2.4}$$

where:

- $\frac{T}{W}$: Total thrust of the engines.
- $\frac{W}{S}$: Required Wing Loading.
- $q$: Dynamic pressure
- $C_D$: Drag coefficient
- $n$: Load factor

4. Lift-to-Drag ratio
   In high-speed flight conditions, a strong thrust-to-weight ratio is vital for countering aerodynamic drag and enabling quick speed changes. Additionally, the lift-to-drag ratio $(L/D)$ significantly influences cruise performance, as it reflects the aircraft's aerodynamic efficiency. This efficiency is governed by the drag equation in combination with the lift characteristics. The drag equation, used in conjunction with lift, defines the overall aerodynamic efficiency:

   - Lift-to-Drag ratio:

$$\frac{L}{D} = \frac{C_L}{C_D} \tag{2.5}$$

   where:

   - $C_L$: Coefficient of lift .
   - $C_D$: Coefficient of drag.

   For interceptors, optimizing the L/D ratio is key to ensuring the aircraft can sustain supersonic speeds while conserving fuel for long-range missions.
5. Required Thrust
   The required thrust is estimated using the thrust-to-weight ratio, which is a critical parameter in assessing aircraft performance, particularly for fighters and interceptors. It is calculated using:

$$T = W_0 \times T/W \tag{2.6}$$

where:

- $W_0$: Takeoff gross weight
- $T/W$: Required thrust-to-weight ratio

Raymer **raymerAircraftDesignConceptual2006a** notes that values for $T/W$ typically range between 0.3 to 0.6 for subsonic aircraft and up to 1.2 or higher for highly agile supersonic fighters, depending on mission roles.

6. Wing Area
   Wing area is determined based on the wing loading $W/S$, which relates the aircrafts weight to the surface area of the wing:

$$S = \frac{W_0}{W/S} \tag{2.7}$$

where:

- $W_0$: Takeoff gross weight
- $\frac{W}{S}$: Required Wing Loading

This parameter plays a key role in determining takeoff and landing performance, stall speed, and maneuverability. According to Raymer **raymerAircraftDesignConce** and Nikolai **nicolaiFundamentalsAircraftAirship2010** theories, wing loading values are selected based on a trade-off between aerodynamic efficiency and operational performance such as field length and rate of climb.

7. Wing Span
   Given the wing area and desired aspect ratio *AR*, the wingspan *b* is calculated using the inverse form of the aspect ratio equation

$$AR = \frac{b^2}{S} \tag{2.8}$$

where:

- $AR$: Aspect ratio
- $S$: Wing Area
- $b$: Wing Span

Solving for *b*, the designer can define the span based on aerodynamic efficiency needs. As it stated by Raymer **raymerAircraftDesignConceptual2006a** and Nikolai **nicolaiFundamentalsAircraftAirship2010**, a higher aspect ratio generally reduces induced drag and is preferred for long-endurance or high-altitude missions.

8. Horizontal and Vertical Tail
   Stability and control surfaces are sized using tail volume coefficients. These are empirical values based on historical aircraft data and are defined separately for horizontal and vertical stabilizers.
   (a) Horizontal Tail Area
   $$S_h = \frac{V_h \times S \times c}{l_h} \tag{2.9}$$
   where:
   - $V_h$: Horizontal tail volume coefficient
   - $S$: Wing Area
   - $c$: Mean Aerodynamic Chord
   - $l_h$: Distance from the aircraft CG to the horizontal tails aerodynamic center

   Using Raymer's typical values of $V_h$ range from 0.7 to 1.1 for fighters. This ensures adequate pitch stability and control authority.
   (b) Vertical Tail Area
   $$S_v = \frac{V_v \times S \times b}{l_v} \tag{2.10}$$
   where:
   - $V_v$: Vertical tail volume coefficient
   - $S$: Wing Area
   - $b$: Wing Span
   - $l_v$: Distance from the aircraft CG to the vertical tails aerodynamic center

   Raymer and Nikolai suggest that values of $V_v$ generally range from 0.02 to 0.07 for military aircraft, depending on mission requirements and whether the aircraft carries external stores or operates at high yaw rates.

9. Fuel Sizing and Endurance

Accurate fuel sizing plays a vital role in defining the interceptor's range and endurance capabilities, particularly during prolonged loitering missions typical of homeland defense operations. To estimate the aircraft's range, the Breguet Range and Endurance equation from Raymer's approach is commonly applied, incorporating key parameters such as lift-to-drag ratio, flight speed, and Specific Fuel Consumption (SFC).

- Range:

$$R = \frac{v}{c} \cdot \frac{L}{D} \cdot \ln\left(\frac{W_i}{W_f}\right) \tag{2.11}$$

- Endurance:

$$E = \frac{\frac{L}{D}}{C} \ln\left(\frac{W_{i-1}}{W_f}\right) \tag{2.12}$$

where:

- $R$ is the range of the aircraft.
- $E$ is the endurance of the aircraft.
- $v$ is the velocity.
- $c$ is the specific fuel consumption (fuel burn per unit thrust per unit time)
- $\frac{L}{D}$ is the Lift-to-Drag ratio
- $W_f$ is the final weight of the aircraft.
- $W_i$ is the initial weight of the aircraft.

## 2.5 Optimization in Interceptor Aircraft Design

Designing interceptor aircraft involves distinct challenges stemming from their critical role in national defense missions. These aircraft must be capable of rapid deployment, combining high speed, maneuverability, and endurance, all while maintaining low structural weight and cost-effectiveness. The integration of advanced computational tools and multidisciplinary design optimization (MDO) platforms such as AVIARY and OpenMDAO, they have significantly advanced this design process, allowing engineers to analyze complex trade-offs and develop optimized, mission-tailored configurations.

### 2.5.1 Objectives in Interceptor Optimization

The primary goal of optimization in interceptor design is to ensure the aircraft meets stringent operational requirements while adhering to technical and economic constraints. Key objectives include:

1. Maximizing Agility and Maneuverability:
   Interceptors often engage in high-speed aerial combat, requiring tight turn radii, rapid climb rates, and effective evasion capabilities. Achieving superior agility involves optimizing the thrust-to-weight ratio ($T/W$) and minimizing moment of inertia **satheConceptualDesignStudies2010**.
2. Sustaining Supercruise Performance:
   Supercruise capabilitysustained supersonic flight without afterburnersis critical for interceptors to reach targets quickly and efficiently. This requires precise optimization of aerodynamic efficiency ($L/D$), propulsion system performance, and overall weight **joinerConceptualSizingNext2018**.
3. Maximizing Operational Range:
   Although interceptors prioritize speed, they must also have sufficient range to engage targets at significant distances or conduct patrol missions. This involves a delicate balance between fuel capacity, structural weight, and aerodynamic design **takasuPreliminarySizingSupersonic1996a**.
4. Minimizing Radar Cross-Section (RCS):
   In modern air combat, stealth capabilities are indispensable. Optimizing the shape and materials of the airframe reduces the RCS, enhancing survivability against advanced detection systems **satheConceptualDesignStudies2010**.
5. Ensuring Mission Effectiveness:
   Optimization ensures the aircraft meets specific mission profiles, including rapid climb to engagement altitude, high-speed interception, and safe return to base, while carrying adequate payloads **satheConceptualDesignStudies2010**.

### 2.5.2 Design Variables in Interceptor Optimization

The optimization process relies on key design variables that significantly impact aircraft performance. These include:

- Thrust-to-Weight Ratio ($T/W$):
  A higher $T/W$ ratio ensures rapid acceleration, shorter take-off distances, and superior climb performance. Typical interceptors achieve $T/W$ values greater than 1, enabling sustained vertical climbs **satheConceptualDesignStudies2010**.
- Wing Loading ($W/S$):
  Wing loading affects both lift generation and maneuverability. Lower $W/S$ values enhance agility but may reduce structural efficiency and aerodynamic performance at higher speeds **takasuPreliminarySizingSupersonic1996a**.
- Fuel Fraction:
  Fuel weight as a fraction of total take-off weight is a critical parameter for determining range and endurance. For interceptors, fuel fractions typically range between 2540% of the total take-off weight **joinerConceptualSizingNext2018**.
- Aerodynamic Parameters:
  Variables such as wing sweep angle, aspect ratio, and airfoil profile determine the drag characteristics at various speeds. For supercruise, optimizing these parameters minimizes wave drag at transonic and supersonic velocities **satheConceptualDesignStudies2010**.
- Propulsion Parameters:
  Engine thrust, bypass ratio, and specific fuel consumption ($SFC$) are optimized to balance high-speed performance with fuel efficiency **satheConceptualDesignStudi**

### 2.5.3 Multidisciplinary Design Optimization (MDO)

The development of modern interceptor aircraft demands a holistic approach, utilizing multidisciplinary design optimization (MDO) to address the complex interactions among aerodynamic performance, propulsion, structural integrity, and mission-specific requirements. Tools like OpenMDAO and AVIARY streamline this process by automating the interrelated analyses and enabling iterative design improvements.

1. Aerodynamic Optimization:
   Aerodynamic efficiency is paramount for interceptors. Using advanced computational fluid dynamics (CFD) and low-fidelity methods for early-stage design, designers optimize $L/D$ for both subsonic maneuverability and

supersonic cruise **takasuPreliminarySizingSupersonic1996a**. Reducing wave drag is critical at supersonic speeds, achieved through optimized wing sweep angles, slender fuselages, and area-ruling principles **satheConceptualDesignStudies2**

2. Propulsion System Optimization:
   Interceptors depend on propulsion systems that deliver high thrust-to-weight ratios while maintaining fuel efficiency during super-cruise. Enhancements in engine materials and thermal management systems support performance during prolonged high-speed missions**satheConceptualDesignStudies2010**. Optimizing specific fuel consumption (SFC) across various mission phasessuch as steep climbs and long-duration patrolsis key to improving operational range **satheConceptualDesignStudies2010**.
3. Structural Weight Optimization:
   A critical aspect of design is minimizing structural weight without compromising strength. Techniques like topology optimization help eliminate excess material while ensuring structural resilience against aerodynamic and inertial stresses **satheConceptualDesignStudies2010**. High-performance materials such as titanium alloys and carbon fiber composites are favored for their excellent strength-to-weight characteristics **takasuPreliminarySizingSupersonic19**
4. Stealth Integration:
   Reducing radar detectability requires thoughtful design of the aircrafts shape and material selection. Smooth surface transitions and the application of radar-absorbent materials are essential in decreasing the radar cross-section (RCS), thus improving survivability in contested environments **satheConceptualDesignStud**

### 2.5.4 Trade Studies in Optimization

Trade studies are essential for evaluating competing design priorities and achieving an optimal balance between conflicting performance objectives in interceptor aircraft. Several critical trade-offs commonly assessed during the design process include:

- Weight vs. Agility:

Lowering structural weight can improve maneuverability, but excessive weight reduction may negatively impact payload capabilities and structural integrity **joinerConceptualSizingNext2018**.

- Fuel Capacity vs. Range and Weight:
  While larger fuel reserves increase operational range, they also add to the aircraft's weight, potentially diminishing its agility. Trade studies help identify the ideal fuel fraction suited to different mission scenarios. **takasuPreliminarySizingSu**
- Supercruise vs. Subsonic Handling:
  Designs tailored for efficient supercruise such as specific wing and fuselage configurations can hinder handling at lower speeds. Designers use trade analyses to fine-tune parameters like wing aspect ratio and control surface geometry **satheConceptualDesignStudies2010**.
- Stealth vs. Aerodynamic Efficiency:
  Enhancing stealth through features like internal weapon bays and blended surfaces often results in increased drag. Careful trade-off evaluations are necessary to balance reduced radar visibility with acceptable aerodynamic efficiency **joinerConceptualSizingNext2018**.

### 2.5.5 Case Studies in Optimization

- Supersonic Fighter Jets:
  Optimization analyses of lightweight interceptorsexemplified by the F-22 Raptorunderscore the necessity of harmonizing structural mass, aerodynamic performance, and propulsion efficiency to maintain sustained supercruise capability **joinerConceptualSizingNext2018**.
- UCAV Optimization:
  Research on unmanned combat aerial vehicles (UCAVs) frequently employs genetic algorithms to find optimal trade-offs among stealth characteristics, payload capacity, and mission endurance. These findings offer valuable guidance applicable to interceptor aircraft development **satheConceptualDesignStudies20**

## 2.6 Overview of OpenMDAO and AVIARY

### 2.6.1 OpenMDAO

OpenMDAO (Open-Source Multidisciplinary Design Analysis and Optimization) developed by NASA, is a powerful open-source computational framework designed to solve complex multidisciplinary optimization challenges. It serves as a valuable platform for integrating various engineering domains, making it especially useful in contemporary aircraft design processes **grayOpenMDAOOpensourceFramework2**

OpenMDAOs modular and flexible architecture supports iterative workflows and promotes collaboration across different technical disciplines. OpenMDAO is particularly effective in implementing Multidisciplinary Design Optimization (MDO), a crucial approach for resolving trade-offs between competing design goals such as minimizing weight, enhancing aerodynamic performance, and improving fuel efficiency.

### 2.6.2 Workflow and Architecture of OpenMDAO

OpenMDAO is developed to efficiently manage interactions between models, solvers, and optimization routines through a unified interface, making it highly suitable for multidisciplinary design applications in aerospace engineering. Its modular structure promotes scalability, re-usability, and clarity in complex system-level analyses. The key components of OpenMDAO are described as follows:

- **Problem**:
  This serves as the top-level entity that governs the overall optimization process. It defines the design variables, objectives, and constraints, and acts as the central interface for running analyses and optimizations.
- **Components**: These are discrete computational units responsible for specific tasks, such as thrust estimation or aerodynamic drag prediction. Each component is designed with explicit input and output definitions, enabling modular and independent operation.
- **Groups**:

Components are organized into hierarchical groups representing major subsystems or disciplinary models (e.g., aerodynamics, propulsion). Grouping allows for structured model development, easier maintenance, and enhanced modularity.

- **Solvers**:
  OpenMDAO integrates both nonlinear and linear solvers to address the coupling between different components. Nonlinear solvers, such as Newton-based methods, are used for converging system-level interactions, while linear solvers like Krylov methods handle subproblem linearizations.
- **Driver**:
  The driver controls the optimization algorithm, iteratively modifying design variables to optimize the specified objective function under defined constraints. Common drivers include:
  - ScipyOptimizeDriver: Performs gradient-based optimization using methods from the SciPy library.
  - GeneticAlgorithmDriver: Used for global search in non-convex problems.
- **Data Connections**: OpenMDAO utilizes explicit and implicit connections to manage data exchange between components and subsystems. These connections ensure data consistency and facilitate the propagation of design sensitivities, which is essential for accurate multidisciplinary optimization.

### 2.6.3 Benefits of OpenMDAO in Aircraft Sizing Design

The adoption of OpenMDAO in aircraft design offers several key benefits that significantly enhance the design process:

- Enhanced Design Precision:
  OpenMDAO effectively reduces design inaccuracies by capturing the interdependencies between engineering disciplinesfor instance, accounting for how structural deflections influence aerodynamic behavior.
- Accelerated Prototyping:

The framework's modular structure allows engineers to rapidly construct and evaluate conceptual models, thereby shortening the design iteration cycle and facilitating early-stage decision-making.

- Resource Efficiency:
  By automating repetitive and complex computational tasks, OpenMDAO minimizes the need for manual calculations, leading to substantial savings in both time and computational effort for multidisciplinary projects.
- Versatile Optimization Capability Optimization:
  OpenMDAO supports both local and global optimization strategies, enabling the exploration of nonlinear and non-convex design landscapes to identify optimal solutions across a wide range of design scenarios.

### 2.6.4 AVIARY

AVIARY is a sophisticated multidisciplinary optimization framework developed within the OpenMDAO ecosystem, specifically designed to streamline aircraft design and analysis. It allows the users to design aircraft, and optimize trajectories simultaneously, add their own subsystem and use gradient-based optimization effectively. By integrating key disciplines such as aerodynamics, propulsion, and structural analysis into a unified workflow, AVIARY supports both conceptual and detailed design studies with high efficiency **AviaryDocumentation**.

Leveraging advanced methodologies inspired by NASAs FLOPS (Flight Optimization System) and GASP (General Aviation Synthesis Program), AVIARY enhances model integration and optimization capabilities. Its modular architecture allows for the construction of highly customizable aircraft models, accommodating diverse configurations and mission profiles.

With its flexibility and robust features, AVIARY empowers users at all levelsenabling beginners to create foundational models, intermediate users to develop sophisticated designs, and advanced practitioners to tackle complex and innovative projects.

### 2.6.5 Tools that build within Aviary

AVIARY integrates a suite of computational tools, including OpenMDAO, Dymos, and several legacy NASA-developed systems that have been adapted for its architecture, such as FLOPS and GASP. At its core, AVIARY is built entirely within the OpenMDAO frameworkan open-source, Python-based platform for multidisciplinary modeling and optimization. As a result, OpenMDAO handles much of the underlying computational infrastructure, including numerical solving, derivative propagation, and parallel execution. This tight integration also enables seamless compatibility with other models or tools developed within the OpenMDAO environment, enhancing modularity and extensibility across design applications.

AVIARY incorporates foundational methodologies and equations from earlier NASA-developed aircraft design tools, most notably FLOPS and GASP. These legacy tools, which were originally created for designing conventional, particularly commercial, aircraft, have had their computational methods systematically ported into AVIARY, ensuring continuity and reliability in performance prediction. Structurally, AVIARY is entirely developed within the OpenMDAO framework, which serves as the backbone of its architecture. Many of its discipline-specific modelssuch as aerodynamics, weight estimation, and propulsionare implemented as individual OpenMDAO systems. In contrast, the framework responsible for integrating these models and conducting mission-level simulations, including time integration and trajectory optimization, is developed using Dymos, which builds upon OpenMDAOs core capabilities.

1. **Dymos**:
   Dymos is an open-source library tailored for optimizing dynamic systems, with a primary focus on trajectory optimization. Built within the OpenMDAO framework, it offers advanced time integration schemes, support for transcription methods, and automated results post-processing. Dymos enables AVIARY to perform coupled mission-level design and optimization by simulating time-dependent flight paths under a wide range of operational scenarios.
2. **OpenMDAO**:

OpenMDAO serves as the core computational infrastructure for AVIARY. Developed by NASA, it is a powerful, open-source platform designed for multidisciplinary analysis and optimization (MDAO). One of its key strengths lies in its efficient handling of derivative computations, which allows for robust gradient-based optimization. Through OpenMDAO, AVIARY benefits from modular integration of various subsystemssuch as aerodynamics, propulsion, and structureswhile enabling parallel processing and scalable system coupling.

3. **FLOPS**:
   Flight Optimization System (FLOPS) **wellsFlightOptimizationSystem** is a legacy suite of computational modules originally developed for the conceptual and preliminary design of advanced aircraft. Its models have been selectively integrated into AVIARY to enhance accuracy and expand analytical coverage. It consists of six primary modules:
   - Weights:
     Uses empirical/statistical relationships to estimate component weights, with additional analytical capabilities for complex wing geometries.
   - Aerodynamics:
     Generates drag polar for performance evaluation and allows for input of custom polar with scaling options for varying configurations.
   - Propulsion data scaling and interpolation:
     Interpolates and scales engine deck data to deliver performance parameters across different flight regimes, adapting to various thrust requirements.
   - Mission Performance:
     Computes flight performance across multiple segmentssuch as climb, cruise, descent, acceleration, turning, and holdingbased on integrated aerodynamic, weight, and propulsion data. It also supports supersonic performance assessments and fuel reserve estimations.
   - Takeoff and Landing:
     Assesses takeoff field lengths (including one-engine-out and abort scenarios), landing distances, approach speeds, and climb gradients, ensuring compliance with FAR Part 25 or MIL-STD-1793 standards.

- Program Control:
  Facilitates point design analysis, parametric sweeps, and nonlinear optimization routines. It allows optimization against objectives like minimum gross weight, fuel burn, or emissions, supporting a wide range of input configurations.

4. **GASP**
   Developed by NASA's Ames Research Center, the General Aviation Synthesis Program (GASP) **haugeGASPGeneralAviation1978** is a computational tool designed to support the preliminary design of aircraft, particularly in the general aviation category. Originally focused on small, fixed-wing aircraft powered by configurations such as single piston engines with fixed-pitch propellers or twin turboprops/turbofans, GASP has since evolved to accommodate more advanced and unconventional configurations, including hybrid wing-body designs and truss-braced wing architectures.
   GASP operates through a series of integrated modules representing key technical disciplinessuch as aerodynamics, propulsion, structures, and performancewhich are computationally linked to reflect the interactions between design variables throughout the sizing and analysis process. This integration allows for consistent evaluation of configuration trade-offs and ensures that cross-disciplinary effects are considered in real time.
   The tool enables users to perform rapid parametric studies, making it especially useful for tasks such as comparative analysis of aircraft concepts, evaluation of performance metrics and economic factors, sensitivity analysis, and the investigation of how emerging technologies influence design outcomes. GASP thus serves as a valuable component in the broader suite of tools used by AVIARY for conceptual design and optimization of both conventional and advanced aircraft systems.

### 2.6.6 Core Functionalities

1. **Core Subsystem**
   AVIARY is organized into five primary subsystems, each responsible for modeling a fundamental discipline in aircraft design:

- Aerodynamics:
  This subsystem employs empirical methods to evaluate aerodynamic performance. It also allows the import of tabular data, such as drag polar and lift curves, which enhances accuracy by enabling users to leverage high-fidelity aerodynamic data from external sources.
- Propulsion:
  The propulsion module uses pre-supplied engine data for interpolation and performance analysis. However, it does not support detailed engine cycle modeling, and therefore, must be supplemented by external tools when such modeling is required.
- Mass:
  Mass estimation is based on empirical relationships that are primarily tailored to conventional aircraft configurations. Nevertheless, AVIARY allows external subsystem integration to support weight prediction for non-traditional or unconventional aircraft designs.
- Geometry:
  The geometry module is built around statistical relationships optimized for conventional tube-and-wing designs. However, like the mass module, it is extendable via external subsystems to accommodate novel geometries and emerging configurations.
- Mission Analysis:
  This subsystem handles trajectory analysis using equations of motion based on two degrees of freedom (2DOF) and height-energy formulations. It supports multiple integration methodsincluding collocation and analytic shootingmaking it suitable for trajectory optimization in both conventional and advanced mission profiles.

2. **Pre-mission, Mission, and Post-mission**
   AVIARY organizes the simulation workflow into three sequential phases:

   - Pre-mission:
     During this phase, the aircraft is initially sized, and all pre-computed values necessary for the mission simulation are generated. This includes data such as weight estimations, fuel requirements, and aerodynamic properties.

- Mission:
  This phase simulates the aircraft's flight along a prescribed trajectory. The performance is evaluated at discrete points in time (nodes), capturing dynamic behaviors such as fuel consumption, thrust, drag, and altitude variations throughout the mission.
- Post-mission:
  After mission simulation, this phase computes final performance metrics and conducts post-processing operations. It includes calculations related to landing performance, economic evaluations, emissions, and any other output parameters necessary for further analysis. The post-mission phase has access to outputs from both pre-mission and mission phases, allowing for holistic system evaluation.

3. **Integration of External Subsystem**
   One of AVIARYs key strengths is its extensibility. Users are provided with the capability to incorporate custom or external subsystems alongside the existing core modules. This design philosophy ensures that while the framework maintains a consistent structure, it also remains adaptable to the diverse needs of modern aircraft design projects.
   The integration of external subsystems is designed to be both user-friendly and highly flexible, accommodating even complex models or non-traditional configurations. These subsystems can be added as standard OpenMDAO components or groups and can be integrated as shown in Figure 2.16 into any of AVIARYs three primary system phases:

   - Pre-Mission: External subsystems can be added here to assist in sizing, environmental modeling, or generating input data needed before the mission simulation.
   - Mission: Users may inject dynamic subsystems that interact with the trajectory, such as adaptive control models, energy storage management, or real-time sensor models.
   - Post-Mission: Subsystems in this phase may handle economic assessments, detailed noise analysis, emissions modeling, or additional post-processing tasks.

Some of the examples of the external subsystem category:

- Acoustics
- Battery Modeling
- Motor Modeling
- Structural Analysis
- Thermal Management System
- Sensor Packages

4. **Optimizer in AVIARY**
   There are 3 types of Optimizer Driver available in Aviary which are *SNOPT*, *IPOPT* and *SLSQP*. Each optimizer have their own driver, setting and values to be used as shown in Table 2.6

TABLE 2.6: Comparison of Optimizers, Drivers, and Settings

| Optimizers | Drivers | Settings |
|---|---|---|
| **SNOPT** | `om.pyOptSparseDriver()` | Major iterations limit: 50<br>Major optimality tolerance: $1 \times 10^{-4}$<br>Major feasibility tolerance: $1 \times 10^{-6}$<br>`iSumm`: 6 |
| **IPOPT** | `om.pyOptSparseDriver()` | **tol**: $1 \times 10^{-9}$<br>**mu_init**: $1 \times 10^{-5}$<br>**max_iter**: 50<br>**print_level**: 5 |
| **SLSQP** | `om.ScipyOptimizeDriver()` | **tol**: $1.0 \times 10^{-9}$<br>**maxiter**: 50<br>**disp**: True |

5. **Types of aircraft and missions AVIARY can design**
   AVIARY is purposefully designed to offer broad flexibility in modeling and optimizing a wide range of aircraft configurations and mission profiles.

While many of its core subsystemssuch as those for aerodynamics, propulsion, and weight estimationwere originally developed with transport-category aircraft in mind, they are built with modularity that allows for easy augmentation or replacement to suit other aircraft types.
This adaptability has been demonstrated in prior applications of AVIARY, which include the successful design of single- and twin-aisle commercial jets, regional aircraft, and advanced configurations such as truss-braced wings and hybrid-electric propulsion systems. The framework's structure supports modifications to accommodate both conventional and unconventional designs, making it suitable for modern aerospace innovation.
Addition to airframe versatility, AVIARY also supports custom mission definitions, enabling users to design and optimize a variety of flight profiles. Mission architectures can be defined with multiple phases, and users can control which parameter such as altitude, throttle setting, or flight path angle are subject to optimization. This open mission architecture makes AVIARY well-suited for trajectory optimization and mission-level trade studies across civil, military, and experimental aircraft platforms.

### 2.6.7 Key Features of AVIARY

- **Open-Source Flexibility:**
  Built upon the Python-based OpenMDAO platform, AVIARY is a fully open-source aircraft design environment. This allows users to extend or customize existing modules by adding new design variables, analysis methods, or mission profiles, ensuring that the framework remains adaptable across a broad range of aircraft design projects and research applications.
- **Multidisciplinary Integration:**
  AVIARY integrates multiple engineering disciplinesincluding aerodynamics, structures, propulsion, and mission analysis into a cohesive modeling framework. This multidisciplinary integration supports complex subsystem interactions, enabling detailed trade studies and comprehensive system-level optimization.
- **Iterative Optimization:**

The design process in AVIARY employs iterative computational loops to refine aircraft parameters. Variables such as wing area, engine thrust, and fuel weight are continuously adjusted until the design meets or exceeds specified performance targets, ensuring convergence to an optimal solution.

- **Scalability:**
  The framework is designed to support a spectrum of applications, from rapid conceptual sizing studies to high-fidelity, physics-based simulations. This scalability makes AVIARY a versatile tool for educational purposes, research explorations, and industrial aircraft development.
- **Support for MDO:**
  By leveraging advanced optimization algorithms within OpenMDAO, AVIARY effectively manages competing design objectivessuch as maximizing range, speed, and fuel efficiencyto identify balanced, optimal aircraft configurations.

### 2.6.8 Draw and Run Mission Profile in AVIARY

AVIARY includes a user-friendly built-in graphical interface for mission definition, accessible via the command line. This interactive tool simplifies the process of defining flight missions by allowing users to specify flight phases with parameters such as altitude, Mach number, and optimization settings.

1. **Draw the Mission Profile in Aviary**
   To use the graphical interface, open a command prompt (terminal window) and run the following command:

```
aviary draw_mission
```

FIGURE 2.17: Aviary draw-mission

This will open a new window with two blank axes and some options as shown in Figure 2.18.

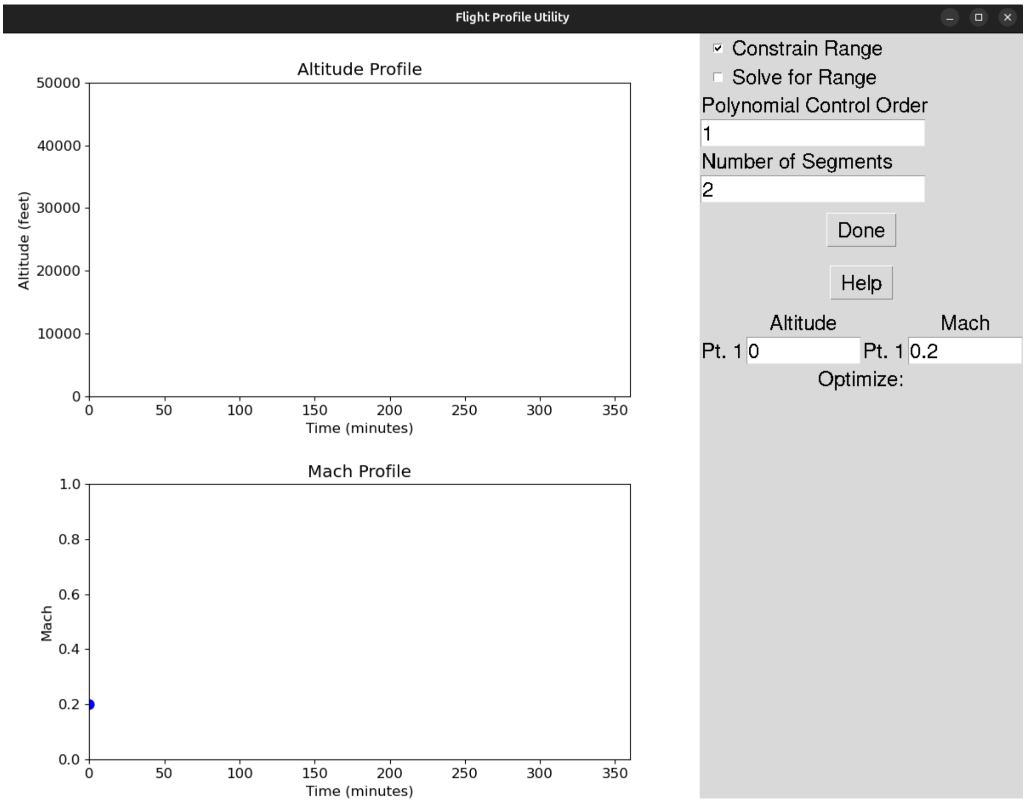



FIGURE 2.18: Flight Profile Utility

The Main component of this GUI flight profile utility are:

- Altitude Profile:
  Displays the altitude variation over time in a graphical plot. Users can add new altitude points by clicking directly on the plot and adjust existing points through drag-and-drop actions. The interface also supports changing units and axis limits via menu options for greater flexibility.
- Mach Profile:
  Similar to the altitude profile, this plot shows the Mach number variation throughout the mission. Users can interactively add and modify Mach points by clicking and dragging on the plot, with options to customize units and axes limits through the menu.
- Optimize Option:
  The GUI allows users to select which flight phases, altitude points, and

Mach values should be included as design variables in the optimization process by simply toggling check boxes. This feature enables targeted and efficient mission optimization tailored to specific design goals.

The AVIARY mission definition interface offers intuitive interactive features that enable precise control over the flight profile:

- Adding Points:
  Users can add new data points by clicking directly on either the Altitude or Mach plot area. Each point on the Altitude plot corresponds to an altitude value (default in feet) at a specific time (default in minutes). Similarly, points on the Mach plot represent Mach numbers at given times.
- Dragging Points:
  Existing points on both Altitude and Mach plots are draggable, allowing users to adjust values dynamically for fine-tuning the flight trajectory.
- Removing Points:
  Points can be deleted by pressing the X button located next to the respective point in the tabular list, facilitating easy modification and cleanup of the flight profile.

As for example as shown in Figure 2.19, it define a relatively simple mission with three phases: climb, cruise, and descent.

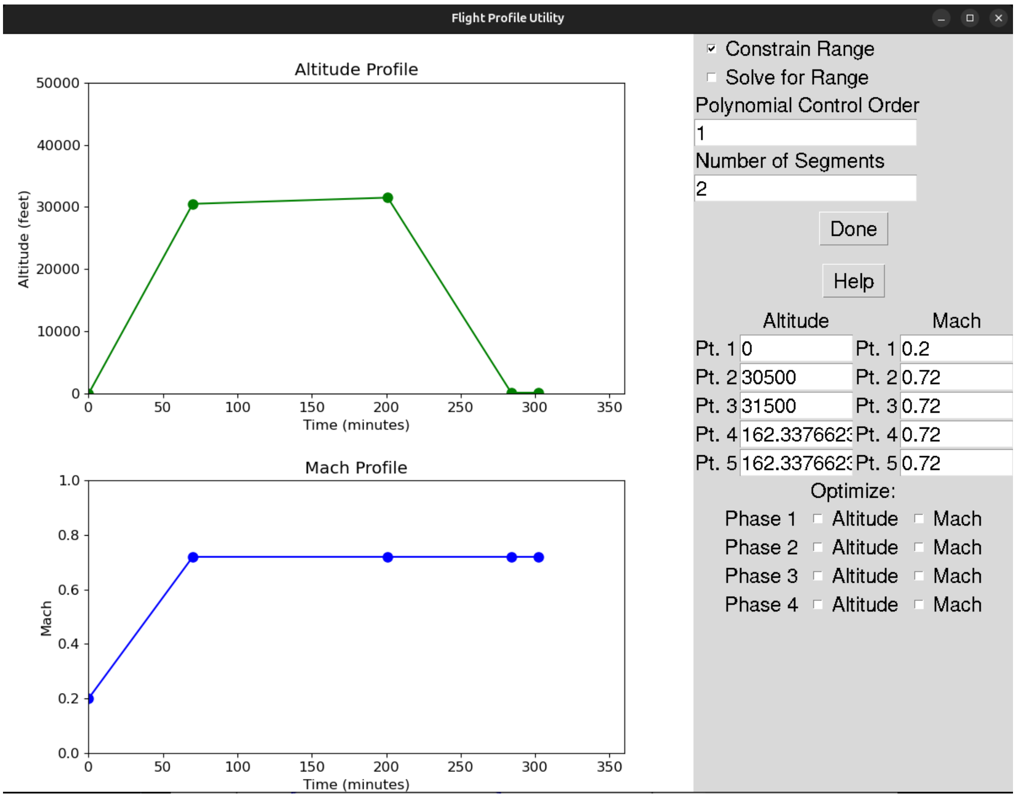



FIGURE 2.19: Drawn Flight Profile Utility

Once done running the GUI of the Flight Profile utility, hit the **Done** button in the top right corner and has been saved to a file called `outputted_phase_info.py` as shown in Appendix B B.4.

The outputted file contains the phase-info dictionary, which holds the flight profile data structured for use in mission simulations. This file could be use for the desire phase parameter for the desired aircraft design model.

2. **Run Mission Simulation**

   Once the mission flight profile has been generated, the next step is to simulate the mission using the `run_mission` command as shown in Figure 2.20. This command executes the simulation based on the parameters defined in the `phase_info` dictionary, which is typically saved in a Python file such as `outputted_phase_info.py`.

   If you have saved the phase information under a different filename, you can specify it using the `phase_info` flag followed by the path to the desired file. In the example provided, a benchmark aircraft configuration in `.csv` format is used as input. However, any valid AVIARY aircraft definition file can be used to simulate a custom mission case.

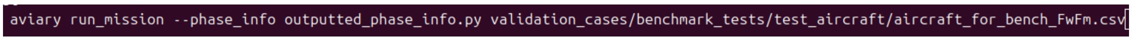



FIGURE 2.20: Aviary Run Mission Simulation

### 2.6.9 Levels in AVIARY Framework

AVIARY is structured into three levels that increase in complexity and fidelity, allowing users to build up from basic design evaluations to detailed, integrated aircraft analyses. Each level progressively incorporates more sophisticated models, enabling a thorough exploration of design possibilities across various parameters. This section presents an overview of Level 1, Level 2, and Level 3 within the AVIARY framework **UserInterfaceLevels**.

1. **Level 1: Baseline Sizing and Conceptual Design**
   Level 1 of AVIARY serves as the starting point for users who only need basic functionality on basic aircraft sizing and initial design parameter estimation. This level offers a straightforward and accessible approach to aircraft analysis and optimization to establish the general characteristics of an aircraft without delving into detailed subsystem analysis. Users at this level can use AVIARYs expected method to define an aircraft and specify the desired mission through a CSV file. As illustrated in Figure 2.21, the main

   inputs for Level 1 include the aircraft model, the types of optimizer, the number of iterations, and the default mission parameters. These inputs feed into the Level 1 process, which then produces a set of design variables, objective metrics, and constraint reports as its output. This forms the basis for subsequent, more sophisticated phases of design and optimization.

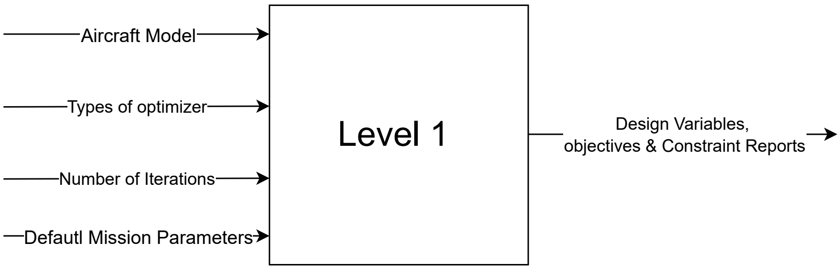



FIGURE 2.21: Level 1 Input Diagram

- Aircraft Model
  AVIARY provides samples of the aircraft model. They are all `.csv` files. This file contains details about the aircraft parameters and its mission. For instance, the file named `aircraft_for_bench_GwGm.csv` represents a mission model for a large single-aisle aircraft.

```
aircraft:air_conditioning:mass_coefficient,1.65,unitless
aircraft:anti_icing:mass,551,lbm
aircraft:apu:mass,928,lbm
......
mission:design:cruise_altitude,37500,ft
mission:design:gross_mass,175400,lbm
mission:design:mach,0.8,unitless
......
settings:equations_of_motion,2DOF
settings:aerodynamics_method,GASP
settings:mass_method,GASP
......
actual_takeoff_mass,0
climb_range,0
cruise_mass_final,0
......
INGASP.ALR,1.11
INGASP.BENGOB,0.05
INGASP.CINP,0.11
```

FIGURE 2.22: Example Aircraft Model Input File

In an AVIARY input file, you'll find a list of variables used by AVIARY, along with their assigned values and corresponding units, as shown in Figure 2.22.

- Run Level 1

  By default, AVIARY allows the optimizer to perform up to 50 iterations to reach an optimal solution. However, in certain cases such as benchmarking, testing, or reducing computation time during early design evaluationsit is often desirable to limit the number of iterations. To reduce runtime for such a scenario, the maximum number of optimizer iterations can be limited to a single iteration. For example, to run the `aircraft_for_bench_GwGm.csv` model with this configuration, the model is executed via the command-line interface (CLI) as shown in Figure 2.23 with a specified option to restrict iterations. This approach allows users to quickly validate input configurations or perform rapid assessments without waiting for full convergence.

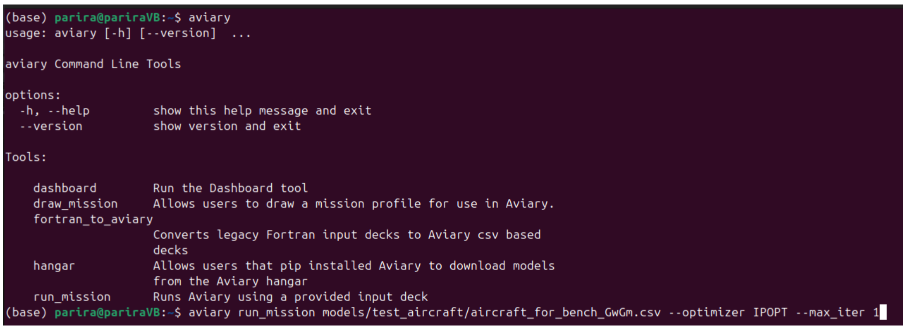



FIGURE 2.23: Run Level 1 in CLI

2. **Level 2: Mission Analysis and Subsystem Integration**

   Level 2 in the AVIARY framework builds upon the preliminary sizing results generated in Level 1 by introducing mission-specific performance evaluations and more detailed subsystem interactions. This level serves as a bridge between conceptual design and comprehensive system-level analysis, enabling a more realistic simulation of the aircraft's behavior under operational conditions. A key feature of Level 2 is its support for the integration of user-defined external subsystems. AVIARY facilitates this through

a flexible interface, allowing users to incorporate custom OpenMDAO systems without requiring an in-depth understanding of the internal Python architecture or the full OpenMDAO framework. To include a new subsystem, users simply implement a builder class that conforms to AVIARYs expected format. At this level, the analysis begins to consider critical design factors such as fuel efficiency, aerodynamic drag optimization, and the coupling between structural and propulsion systems. These enhancements provide a more holistic view of aircraft performance, supporting iterative design improvements and mission-tailored optimization.
As illustrated in Figure 2.24, the main inputs for Level 2 include the aircraft model, the types of optimizer, the number of iterations, the mission parameters, the objective to be optimized, and the AVIARY-specific problem formulation. This information feeds into the Level 2 process, which then produces updated design variables, objective metrics, and constraint reports, reflecting the multidisciplinary nature of the aircraft design. The results enable designers to perform more realistic and robust evaluations of their concepts, guiding further design decisions.

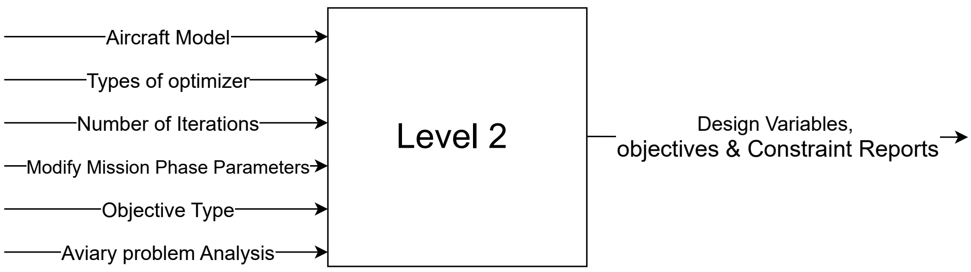



FIGURE 2.24: Level 2 Input Diagram

In the Python script, Level 1 prepares the basic set of arguments and then calls the `run-aviary()` function to execute the sizing analysis. In contrast, Level 2 introduces additional arguments into the script to support the enhanced fidelity and integration of mission-specific and subsystem-level details which are:

- Aircraft Filename:

Specifies the path to the aircraft definition file (`models/test-aircraft/aircraft_for_bench_GwGm.csv`). This file contains key sizing parameters such as geometry, weight, propulsion, and aerodynamic properties.

- Phase Info:
  A dictionary that defines the mission phases (e.g., climb, cruise, descent), including boundary conditions such as altitude, Mach number, and duration for each phase.

- Objective Type:
  Specifies the optimization objective for the mission analysis. Common objectives include minimizing fuel burn, maximizing range, or minimizing takeoff gross weight.

- Analysis Scheme:
  Indicates the numerical method used for solving the equations of motion, such as transcription schemes in Dymos.

- Optimizer:
  Selects the gradient-based optimizer to be used during the simulation (e.g., *SNOPT*, *SLSQP*, or *IPOPT*).

- Record File Name:
  Defines the name of the output file used to store optimization results and iteration history for post-processing.

- Restart File Name:
  Allows the simulation to resume from a previously saved state, reducing computation time in iterative design workflows.

- Max Iteration:
  Sets the upper limit for the number of optimization iterations, controlling runtime and convergence behavior.

All of the arguments listed earlier are fairly straightforward, except for the

Objective Type. In Level 2, the Objective Type is set to None by default. This means the system automatically chooses the objective based on the Problem Typebut only when using the GASP-based mission method. If you are using the FLOPS-based mission method, then the objective must be defined manually. There are three options for problem type that is shown in Table 2.7.

TABLE 2.7: Types of problems for the Objective Type

| Problem Type | Objectives |
|---|---|
| Sizing | Mission.Objective Fuel |
| Alternate | Mission.Objective Fuel |
| Fallout | Mission.Objective.Range |

- **Run Level 2**
  To run Level 2, you need to create a Python script to reproduce the run of the `aircraft_for_bench_GwGm` model, which was originally used as an example in Level 1.
  However, unlike Level 1 where only basic arguments were needed, Level 2 requires additional arguments and options for more detailed mission analysis and optimization. This includes specifying things like the aircraft file, phase information, optimizer settings, and how many iterations to run as shown in Appendix B B.1.
  In the python script, we have the flexibility to modify the phase info parameters for the desired aircraft from default phase info of FLOPS and GASP as shown in Appendix B B.2 and B.3.
  The users are granted the flexibility to define external subsystems-through the Phase Info configuration. As described in the Custom Subsystem subsection of the AVIARY documentation, multiple categories of external subsystems can be incorporated into the modeling process. These external components may include user-defined aerodynamic models, propulsion modules, or mission-specific control systems. Here are the example python script to applied in the Phase Info as shown in Appendix B B.5.

As shown in that example, it insert the external subsystem of battery and simple weight. This is a simplified example of a component that computes a weight for the wing and horizontal tail. It does not provide realistic computations but rough estimates to `Aircraft.Wing.MASS` and `Aircraft.HorizontalTail.MASS`. When this external subsystem is added to your pre-mission phase, Aviary will compute these weights in its core subsystem as usual, but then the wing mass and tail mass values will be overridden by this external subsystem.

3. **Level 3: Detailed Multidisciplinary Optimization (MDO)**
Level 3 represents the highest level of fidelity within the AVIARY framework, emphasizing comprehensive multidisciplinary design and optimization (MDO). At this stage, the design process incorporates highly detailed subsystem models and captures the complex inter-dependencies between aerodynamics, propulsion, structural components, mission performance, and other aircraft systems. This level is particularly valuable for refining aircraft concepts based on realistic performance data and simulating mission-specific behavior with high accuracy. Iterative optimization processes are employed to adjust critical parameterssuch as wing geometry, thrust levels, and fuel consumptionensuring that the aircraft configuration meets all operational and performance constraints. By integrating detailed simulations with optimization loops, Level 3 enables engineers to conduct system-level trade studies and arrive at well-balanced, optimal aircraft designs that are ready for advanced development or prototyping phases as illustrate in Figure 2.25.

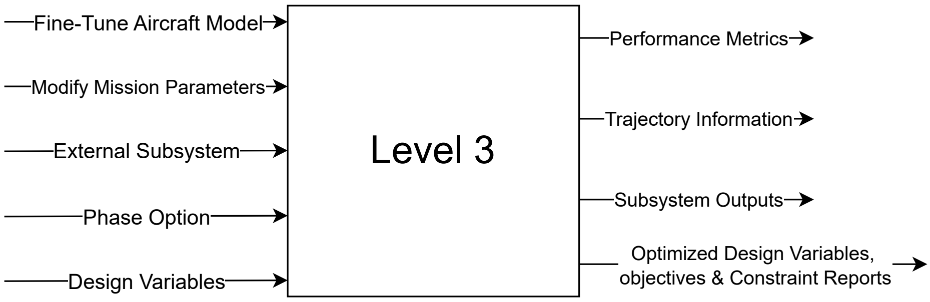



FIGURE 2.25: Level 3 Input Diagram

In the `aviary/validation-cases/benchmark-tests` directory, AVIARY provides a comprehensive full-mission benchmark test for the NASA N3CC aircraft model shown in Appendix B B.6. This example demonstrates the setup and execution of a high-fidelity, multidisciplinary design and optimization (MDO) analysis using AVIARY's Level 3 interface. Level 3 offers the highest degree of user control, allowing for detailed customization of the aircraft model, mission profile, and optimization parameters. By following this example, users can gain insights into constructing and analyzing complex aircraft models beyond the constraints of predefined templates.
This structured approach facilitates a comprehensive analysis of the N3CC aircraft's performance across various mission segments, including takeoff, climb, cruise, descent, and landing. By leveraging AVIARY's Level 3 capabilities, users can perform detailed trade studies and optimization tasks tailored to specific design requirements.

### 2.6.10 Outputted Result Files

The output files generated by AVIARY vary depending on the type and level of analysis performed. Using the `aircraft_for_bench_GwGm.csv` aircraft model, it generates several output files in `.HTML`:

1. **Driver_scaling_report.html**

This HTML report summarizes the driver scaling information used during the optimization process as shown in Figure 2.26. After defining all the design variables, objectives, and constraints, and once the optimization problem has been properly configured, this file serves as a centralized summary of the problem setup. The contents are typically divided into three main sections:

- Design Variables:
  Lists all the variables involved in the optimization process, organized by phase.
- Constraints:
  Displays the imposed limits or requirements on the system, also grouped by mission phase.
- Objectives:
  Indicates the primary performance metrics being optimized.

In addition to these sections, the report includes **Jacobian information**, which presents the sensitivities (derivatives) of responses with respect to the design variables. This information is crucial for assessing the behavior of the optimization algorithm and diagnosing potential issues related to scaling or convergence.

**Design Variables**

| name | size | indices | Driver | | Model | | ref | ref0 | scaler | adder | upper | lower |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | value | units | value | units | | | | | | |
| mission:takeoff:ascent_t_initial | 1 | | 0.333333 | | 10 | s | 30 | | 0.0333333 | | 3.33333 | 0 |
| mission:takeoff:ascent_duration | 1 | | 2.5 | | 25 | s | 10 | | 0.1 | | 100 | 0.1 |
| tau_gear | 1 | | 0.2 | unitless | 0.2 | unitless | 1 | | | | 1 | 0.01 |
| tau_flaps | 1 | | 0.9 | unitless | 0.9 | unitless | 1 | | | | 1 | 0.01 |
| mission:design:gross_mass | 1 | | 1.00229 | lbm | 175400 | lbm | 175000 | | 0.00000571429 | | 2.28571 | 0.0000571429 |
| mission:summary:gross_mass | 1 | | 1.00229 | lbm | 175400 | lbm | 175000 | | 0.00000571429 | | 2.28571 | 0.0000571429 |
| traj.accel.t_duration | 1 | | 0.129353 | | 13 | s | 100.5 | | 0.00995025 | | 1.99005 | 0.00995025 |
| ⊞ traj.accel.states:velocity | 3 | | \|\| **1** \|\| | | \|\| **250** \|\| | knot | 250 | 150 | 0.01 | −150 | \|\| **1.2** \|\| | \|\| **0** \|\| |
| ⊞ traj.accel.states:mass | 3 | | \|\| **1.15764** \|\| | | \|\| **173646** \|\| | lbm | 150000 | 0 | 0.00000666667 | | \|\| **6.66667e+15** \|\| | \|\| **0** \|\| |
| ⊞ traj.accel.states:distance | 4 | | \|\| **0.658315** \|\| | | \|\| **3.29158** \|\| | nmi | 5 | 0 | 0.2 | | \|\| **30** \|\| | \|\| **0** \|\| |
| ⊞ traj.ascent.states:flight_path_angle | 12 | | \|\| **0.139626** \|\| | | \|\| **0.139626** \|\| | rad | 1 | 0 | 1 | | \|\| **20** \|\| | \|\| **10** \|\| |
| ⊞ traj.ascent.states:altitude | 12 | | \|\| **0.5** \|\| | | \|\| **500** \|\| | ft | 1000 | 0 | 0.001 | | \|\| **0.7** \|\| | \|\| **0** \|\| |
| ⊞ traj.ascent.states:velocity | 12 | | \|\| **0.925** \|\| | | \|\| **185** \|\| | knot | 200 | 0 | 0.005 | | \|\| **3.5** \|\| | \|\| **0** \|\| |
| ⊞ traj.ascent.states:mass | 12 | | \|\| **1.15764** \|\| | | \|\| **173646** \|\| | lbm | 150000 | 0 | 0.00000666667 | | \|\| **6.66667e+15** \|\| | \|\| **0** \|\| |
| ⊞ traj.ascent.states:distance | 12 | | \|\| **1** \|\| | | \|\| **10000** \|\| | ft | 10000 | 0 | 0.0001 | | \|\| **1.5** \|\| | \|\| **0** \|\| |

FIGURE 2.26: Design Variables - Driver Scaling Report

2. **Inputs.html**

The Inputs.html file provides a sortable and filterable report of all input variables used across the different phases of the `aircraft_for_bench_GwGm.csv`. This HTML report allows for easy inspection and debugging of model inputs and supports an interactive interface for enhanced usability, as illustrated in Figure 2.27.

Upon initial loading, the report displays only those input variables originating from IVC (short for IndepVarComp), which are typically externally provided or user-defined inputs. However, users can expand the view by selecting or deselecting the corresponding checkbox to include other inputs as needed.

The interface allows users to filter input variables based on a variety of criteria, including:

- Input name
- Source name
- Units
- Shape
- Tags
- Value (Val)
- Design Variable status

This report is especially valuable during model setup and verification stages, as it allows users to trace variable definitions, verify initial conditions, and ensure consistency across different mission phases and subsystem inputs.

**Inputs Report for aviary**

| Input Name | Source Name | Source is IVC | Source is DV | Units | Shape | Tags | Val | Min Val | Max Val |
|---|---|---|---|---|---|---|---|---|---|
| aircraft:fuel:fuel_margin | _auto_ivc.v4 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |
| aircraft:air_conditioning:mass | _auto_ivc.v7 | ✓ | | lbm | (1,) | [] | [0] | 0 | 0 |
| aircraft:air_conditioning:mass_scaler | _auto_ivc.v8 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |
| aircraft:anti_icing:mass | _auto_ivc.v9 | ✓ | | lbm | (1,) | [] | [0] | 0 | 0 |
| aircraft:anti_icing:mass_scaler | _auto_ivc.v10 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |
| aircraft:apu:mass | _auto_ivc.v11 | ✓ | | lbm | (1,) | [] | [0] | 0 | 0 |
| aircraft:apu:mass_scaler | _auto_ivc.v12 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |
| aircraft:avionics:mass | _auto_ivc.v13 | ✓ | | lbm | (1,) | [] | [0] | 0 | 0 |
| aircraft:avionics:mass_scaler | _auto_ivc.v14 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |
| aircraft:blended_wing_body_design:cabin_area | _auto_ivc.v15 | ✓ | | ft**2 | (1,) | [] | [0] | 0 | 0 |
| aircraft:blended_wing_body_design:passenger_leading_edge_sweep | _auto_ivc.v16 | ✓ | | deg | (1,) | [] | [45] | 45 | 45 |
| aircraft:canard:area | _auto_ivc.v17 | ✓ | | ft**2 | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:aspect_ratio | _auto_ivc.v18 | ✓ | | unitless | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:characteristic_length | _auto_ivc.v19 | ✓ | | ft | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:fineness | _auto_ivc.v20 | ✓ | | unitless | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:laminar_flow_lower | _auto_ivc.v21 | ✓ | | unitless | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:laminar_flow_upper | _auto_ivc.v22 | ✓ | | unitless | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:mass | _auto_ivc.v23 | ✓ | | lbm | (1,) | [] | [0] | 0 | 0 |
| aircraft:canard:mass_scaler | _auto_ivc.v24 | ✓ | | unitless | (1,) | [] | [1] | 1 | 1 |

FIGURE 2.27: Input Reports for AVIARY

3. **N2.html**

   The N2 diagram, also known as an eXtended Design Structure Matrix (XDSM), is a valuable visualization tool provided by OpenMDAO to help users understand and navigate complex system interactions within a model. This matrix-style diagram captures the functional and physical interfaces between system elements, making it easier to systematically identify, define, tabulate, and analyze relationships across the model structure.

   In the context of AVIARY, the N2 diagram becomes particularly insightful when running the run-mission command for a given aircraft model, such as `aircraft_for_bench_GwGm.csv`. An example of the resulting N2 output is illustrated in Figure 2.28.

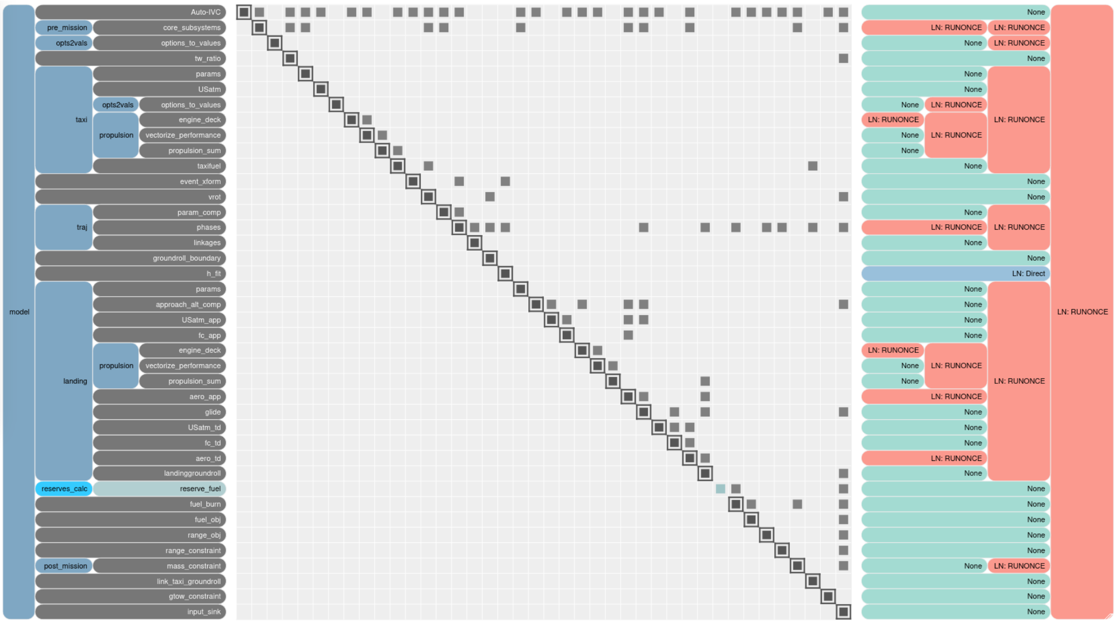



FIGURE 2.28: N2 Diagram Report

The diagram provides a hierarchical view of all model components and their data connections. For mission-level analysis, users can zoom into the traj (trajectory) system, and further into the phases group. Within this group, the mission phasessuch as groundroll, climb, cruise, and desc2are clearly delineated, as shown in Figure 2.29. This structure confirms the expected configuration and data flow across mission segments, assisting both in debugging and model validation.

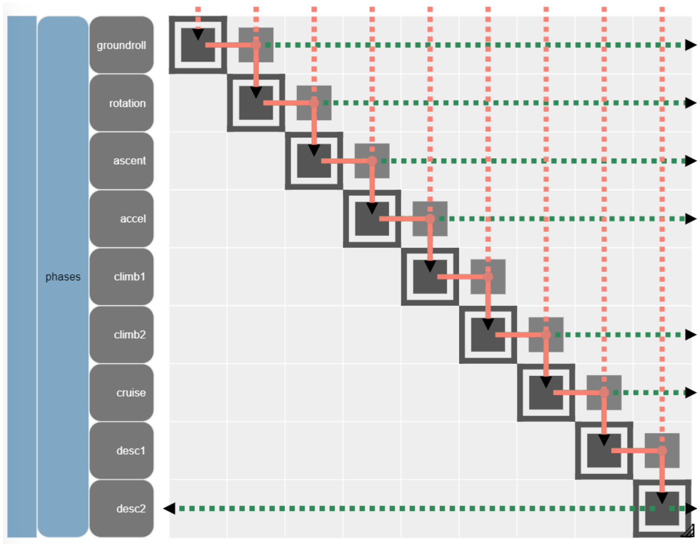



FIGURE 2.29: N2 Phases Diagram

The solid arrow connections in the N2 diagram visually represent the data flow between mission phases. Specifically, they indicate how outputs from one phase serve as inputs for the next. This chaining of phases reflects the sequential nature of flight dynamics within AVIARYs mission analysis. For example, the final state variables of the groundroll phase become the initial conditions for the climb phase, and so forth. This clear representation ensures traceability of data and allows users to verify that each phase is properly initialized from the output of its predecessor, which is critical for maintaining physical and computational continuity throughout the mission simulation.

4. **Opt_Report.html**
   This file is an OpenMDAO generated optimization report, which presents all design information in unscaled, physical units for clarity. At the top, it provides a concise summary of the optimization status, including whether convergence was achieved. Below that, it details the objective function, design variables, constraints, and optimizer settings, as illustrated in Figure 2.30. This report is essential for evaluating the performance and feasibility of the optimized aircraft configuration.

**OpenMDAO Optimization Report for Problem aviary (All values are in unscaled, physical units)**

| | |
|---|---|
| Problem: | aviary |
| Script: | /home/parira/anaconda3/bin/aviary |
| Optimizer: | pyOptSparse_IPOPT |
| Number of driver iterations: | 4 |
| Number of model evals: | 4 |
| Number of deriv evals: | 3 |
| Execution start time: | 2024-11-26 23:03:10 |
| Wall clock run time: | 00 hours 00 minutes 31 seconds 87.4 milliseconds |
| Exit status: | FAIL |

**Objectives**

| name | val | ref | ref0 | adder | scaler | units |
|---|---|---|---|---|---|---|
| mission:objectives:fuel | [4.79539004] | 1.0 | | | | |

**Design Variables**

| name | alias | size | min | max | mean | lower | upper | equals | ref | ref0 | units | visual |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mission:takeoff:ascent_t_initial | | 1 | 0.334 | 0.334 | 0.334 | 0 | 3.33 | | 30 | None | | 0.0 3.3333333333333335 0.334 |
| mission:takeoff:ascent_duration | | 1 | 1.36 | 1.36 | 1.36 | 0.1 | 100 | | 10 | None | | 0.1 100.0 1.36 |
| tau_gear | | 1 | 0.244 | 0.244 | 0.244 | 0.01 | 1 | | 1 | None | unitless | 0.01 1.0 0.244 |
| tau_flaps | | 1 | 0.873 | 0.873 | 0.873 | 0.01 | 1 | | 1 | None | unitless | 0.01 1.0 0.873 |
| mission:design:gross_mass | | 1 | 0.998 | 0.998 | 0.998 | 5.71e-05 | 2.29 | | 1.75e+05 | None | lbm | 5.71428571428571[illegible]142857142856 0.998 |
| mission:summary:gross_mass | | 1 | 0.998 | 0.998 | 0.998 | 5.71e-05 | 2.29 | | 1.75e+05 | None | lbm | 5.71428571428571[illegible]142857142856 0.998 |
| traj.accel.t_duration | | 1 | 0.299 | 0.299 | 0.299 | 0.00995 | 1.99 | | 100 | None | | 0.00995024875621[illegible]0497512437811 0.299 |

FIGURE 2.30: OpenMDAO Optimization Report

If IPOPT is used as the optimizer, an additional output file named `IPOPT.out` is generated, whereas SLSQP optimization using the pyOptSparseDriver produces `SLSQP.out`. In practice, IPOPT and SNOPT tend to deliver better convergence behavior for complex AVIARY optimization problems due to their advanced algorithmic features. However, SLSQP remains popular due to its accessibilitybeing included by default in the SciPy librarywhich makes it a convenient option for many users, especially during early-stage development or academic work.

5. **Traj_Linkage-Report**
   This is a Dymos-generated linkage report that visualizes the connections between mission phases using a customized N2 diagram, as illustrated in Figure 2.31. This report details how various flight phases are linked either through continuity constraints or direct variable connections. It provides a clear graphical representation of the data flow and consistency requirements across mission segments, such as altitude, Mach number, fuel state, and time.

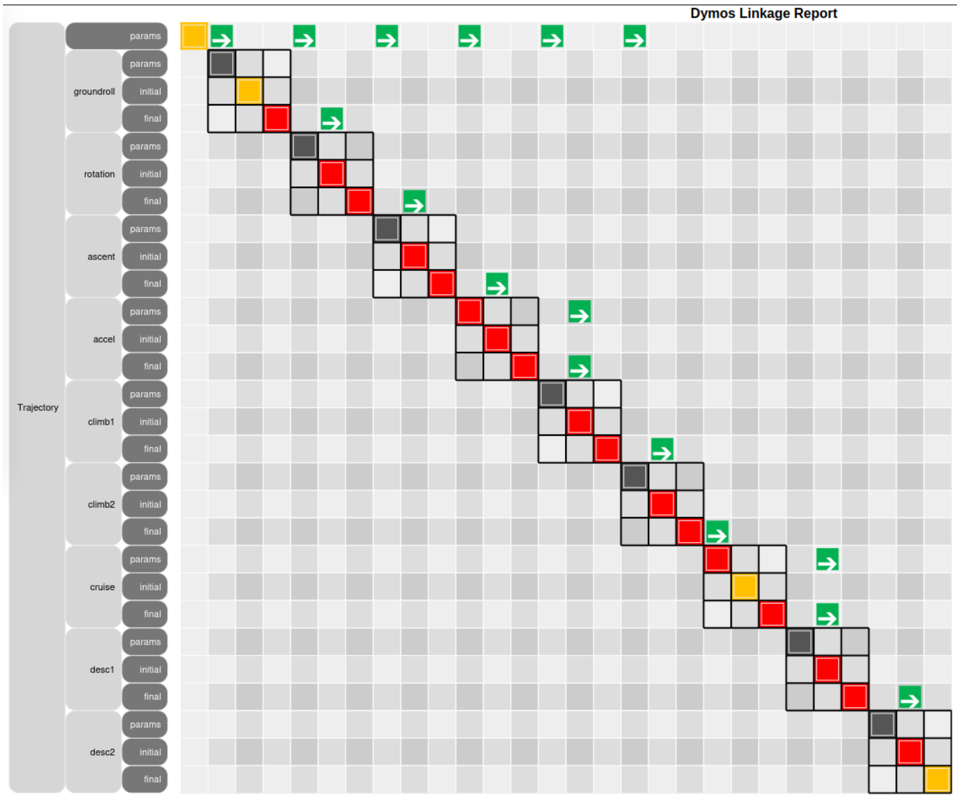



FIGURE 2.31: Dyamos Linkage Report

This tool is particularly useful for identifying inconsistencies or unintended linkages between phases. For instance, it helps ensure that fixed boundary conditions such as takeoff and landing weights, or flight path continuity, are properly enforced. By providing a systematic overview of these linkages, the Traj-Linkage-Report plays a vital role in validating mission setup and improving model robustness within the AVIARY framework.

6. **Traj_Result_Report.html**
   This file is one of the most important outputs generated by AVIARY. It allows users to visualize and analyze the optimized mission trajectory of the aircraft. The report includes various tabs containing timeseries data and phase-specific parameters. Based on `aircraft_for_bench_GwGm.csv` aircraft model, the mission is divided into several phases: groundroll, rotation, ascent, acceleration, climb1, climb2, cruise, descent1, and descent2. In the timeseries tab, users can choose which phases to display, as illustrated in

Figure 2.32.

This is one of the key output files generated by the AVIARY framework. It offers a detailed visualization and analysis of the optimized mission trajectory, making it essential for assessing aircraft performance over time and during various mission phases. The report features multiple tabs that display both timeseries data and parameters specific to each phase.

For the aircraft model `aircraft_for_bench_GwGm.csv`, the mission is broken down into several discrete phases: groundroll, rotation, ascent, acceleration, climb1, climb2, cruise, descent1, and descent2. Within the timeseries tab, users can select and isolate specific mission phases to examine variables such as altitude, Mach number, fuel flow, angle of attack, and thrust-to-weight ratio. This functionality is illustrated in Figure 2.32.

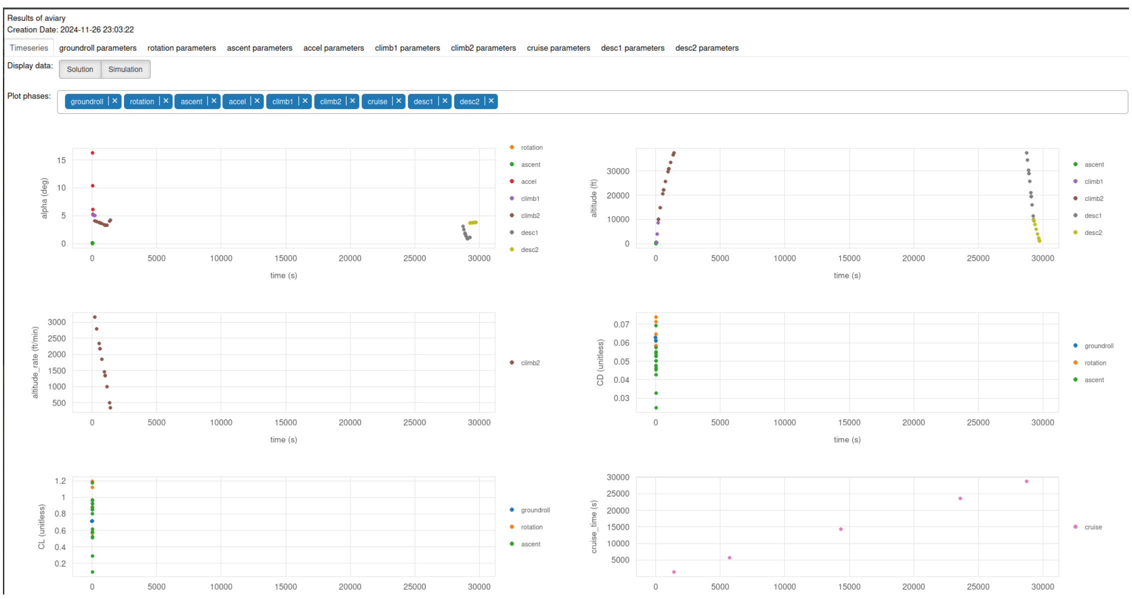
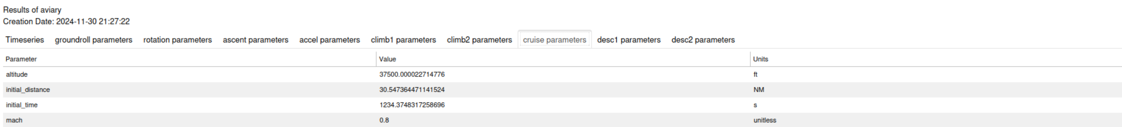



FIGURE 2.32: Trajectory Result Report

Each phase-specific tab contains a detailed summary of key performance metrics and state variables relevant to that segment. For instance, the cruise phase tab, as shown in Figure 2.33, includes data such as cruise altitude, speed, fuel consumption rate, and lift-to-drag ratio. These visualizations are essential for analyzing the behavior and efficiency of the aircraft in each flight phase and for verifying whether mission and design constraints are being satisfied. This report supports diagnostic validation, performance tuning, and iterative design improvements, making it a powerful tool in both academic research and engineering practice.


Results of aviary
Creation Date: 2024-11-30 21:27:22

Timeseries | groundroll parameters | rotation parameters | ascent parameters | accel parameters | climb1 parameters | climb2 parameters | cruise parameters | desc1 parameters | desc2 parameters

| Parameter | Value | Units |
|---|---|---|
| altitude | 37500.000022714776 | ft |
| initial_distance | 30.547364471141524 | NM |
| initial_time | 1234.3748317258696 | s |
| mach | 0.8 | unitless |

FIGURE 2.33: Cruise Parameter from Trajectory Result Report

## 2.7 Comparative Analysis of AVIARY with Other Design and Optimization Tools

Numerous other framework tools exist for aircraft sizing and design optimization as shown in Table 2.8. However, AVIARY offers unique advantages:

- **Openness and Flexibility**: As an open-source tool built on Python, AVIARY allows customization of modules and the incorporation of new variables as required by specific design goals.
- **Multidisciplinary Optimization (MDO)**: AVIARY is highly integrated with OpenMDAO, making it particularly suited for MDO applications, which are essential in balancing the diverse parameters of interceptor design.
- **Modularity**: AVIARYs modular approach enables engineers to focus on specific design aspects, such as aerodynamics or propulsion, without needing to overhaul the entire design.

TABLE 2.8: Comparative Analysis of AVIARY with Other Design and Optimization Tools

| Tool | Design Capabilities | Optimization | Analysis Level |
|---|---|---|---|
| AVIARY | Conceptual, Preliminary | Gradient-Based (OpenMDAO) | Multidisciplinary |
| OpenVSP | Conceptual | Manual or Local Search | Low Fidelity |
| SUAVE | Conceptual | Metaheuristic (PSO, GA) | Medium Fidelity |
| XFLR5 | Detailed (2D/3D Aero) | None | Aerodynamics (low Reynolds number) |
| FAST-OAD | Conceptual, Preliminary | Gradient-Based (OpenMDAO) | Multidisciplinary |

## 2.8 Interceptor Aircraft Design Trade Studies

Trade studies play a crucial role in aircraft sizing and design by enabling the systematic evaluation and comparison of different design approaches, configurations, and technologies. Through this analytical process, trade studies help engineers and researchers gain valuable insights into how to balance often competing objectives such as performance, cost, and operational effectiveness.

In the context of supersonic aircraft, mission-specific sizing, and unmanned combat aerial vehicles (UCAVs), various trade studies have been conducted to address key design challenges. These include aerodynamic optimization for reduced drag and improved efficiency, stealth features to minimize radar cross-section, multi-objective trade-offs balancing speed, range, and payload, and mission adaptability to meet diverse operational requirements. Each trade study contributes a unique perspective that informs and guides the overall design process toward optimized and feasible aircraft solutions.

### 2.8.1 Supersonic Aircraft Design

- **Takasu et al. (1996):**
  The study demonstrated effective techniques for sizing supersonic transports with a focus on reducing maximum takeoff weight (MTOW) and improving fuel efficiency. These findings are applicable to interceptors requiring supersonic cruise capabilities for rapid threat response. Mixed-flow turbofan engines with optimized bypass ratios were highlighted as critical for achieving high-speed and efficient propulsion **takasuPreliminarySizingSupersonic1996a**.
- **Furukawa & Makino (2007) :**
  Focused on silent supersonic technologies, the research emphasized aerodynamic efficiency and noise reduction. While primarily aimed at civil supersonic aircraft, the findings on drag reduction and sonic boom minimization are relevant for interceptors operating in civilian airspace **furukawaConceptualDesignAero**

### 2.8.2 Mission Requirement and Aircraft Sizing

- **Parker (1986):**

Highlighted the importance of matching mission-specific scenarios, such as rapid interception and air superiority roles, with design parameters like thrust-to-weight ratio and wing loading. Iterative sizing approaches were employed to balance performance with cost, making this methodology particularly useful for interceptor aircraft that must achieve rapid deployment and high maneuverability **parkerMissionRequirementsAircraft1986**.

### 2.8.3 Unmanned Combat Aerial Vehicle (UCAV) Design

- **Sathe & Pant (2010):**
  Their methodology for UCAV sizing, which included optimizing takeoff weight and wing loading through constraint analysis, provides insights applicable to manned interceptors. The studys focus on low-altitude dash speed and maneuverability constraints aligns with interceptor requirements for high-speed engagements **satheConceptualDesignStudies2010**.

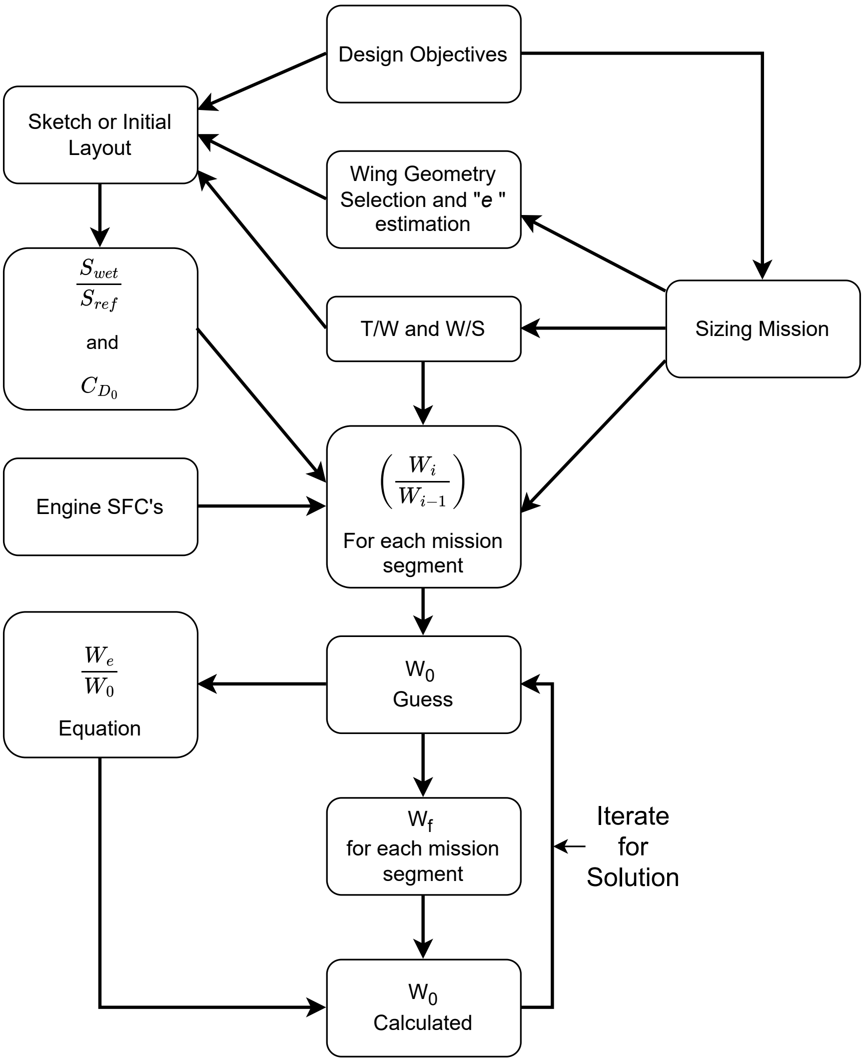



FIGURE 2.12: Sizing Flowchart Diagram **raymerAircraftDesignConceptual2006a**

FIGURE 2.13: Aviary **AviaryDocumentation**

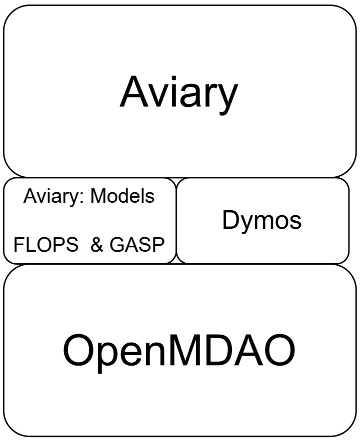



FIGURE 2.14: Aviary Building Tool Blocks **AviaryDocumentation**

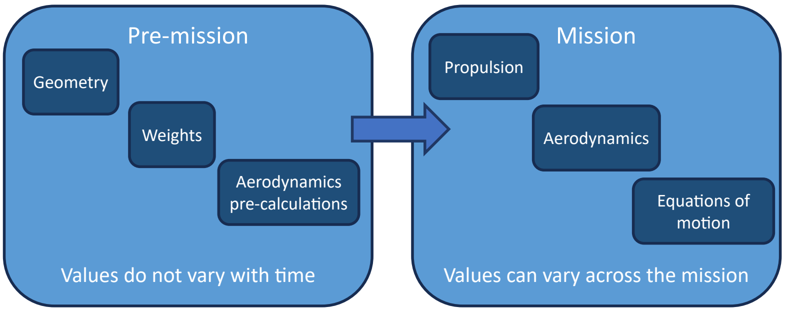



FIGURE 2.15: Pre-Mission and Mission System Diagram **AviaryDocumentation**

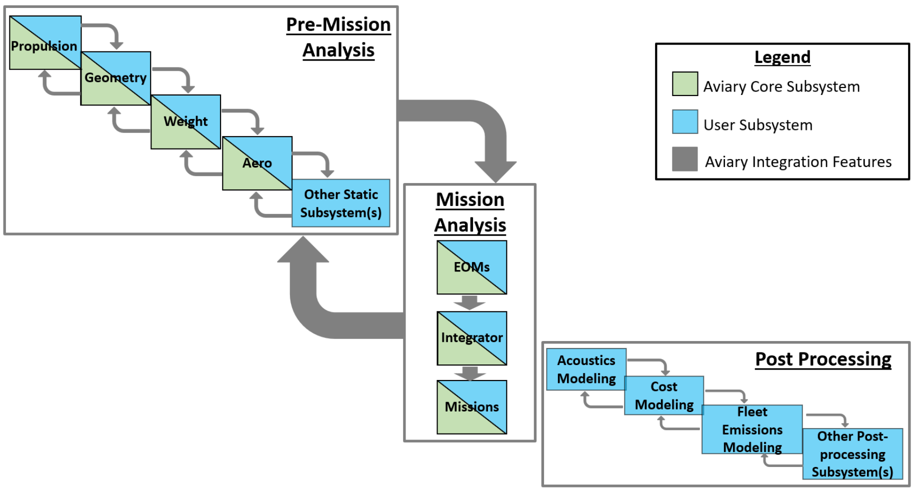



FIGURE 2.16: Add External Subsystem into the Aviary Analysis **AviaryDocumentation**

# CHAPTER 3
# RESEARCH METHODOLOGY

This chapter defines the research methodology employed to perform the preliminary sizing of the Homeland Defense Interceptor (HDI) aircraft using AVIARY. The methodology is structured to systematically translate the mission requirements defined in the Request for Proposal (RFP) into a feasible aircraft configuration through a step-by-step design and analysis process. The focus of this study is on the conceptual design phase, particularly on preliminary sizing activities, supported by parametric simulation and optimization.

## 3.1 Research Framework

The research methodology employed in this thesis follows a structured aircraft design process focused on the conceptual design phase, particularly the preliminary sizing of the Homeland Defense Interceptor (HDI). The framework is illustrated in the flowchart in Figure 3.1 and the yellow-highlighted blocks indicate the specific phases covered in this thesis, focusing on the preliminary sizing and early optimization using AVIARY.

### 3.1.1 Research Flowchart

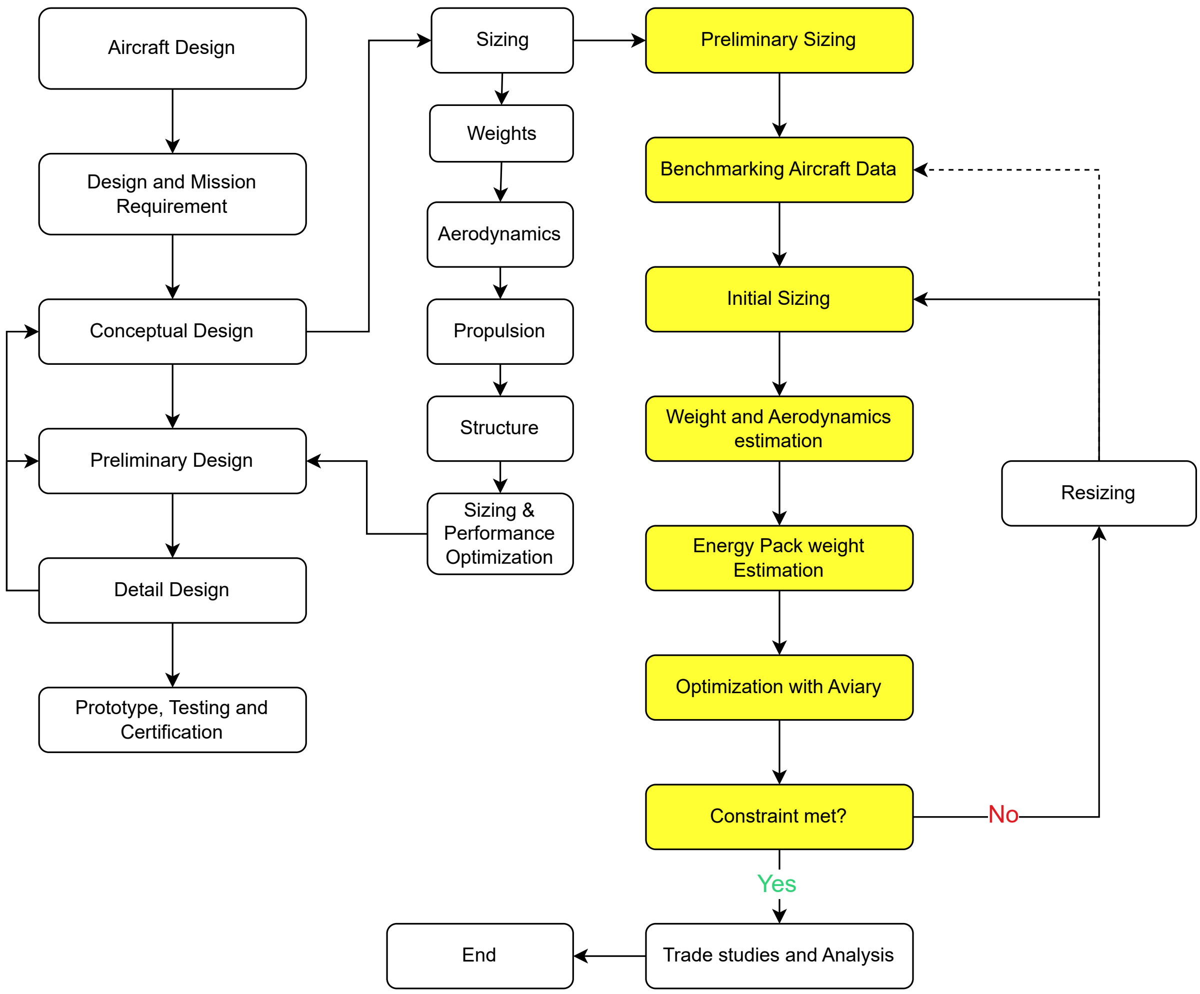



FIGURE 3.1: Research Flowchart

Figure 3.1 represents the steps taken throughout the entire process of making this thesis. The thesis scope is highlighted with yellow color.

### 3.1.2 Computational Environment

To ensure the reliability, reproducibility, and performance of the simulation and aircraft sizing processes in this research, all computational work was conducted in a controlled and well-documented software environment. The selected tools and libraries were chosen based on their compatibility with the OpenMDAO framework and the requirements of the Aviary aircraft sizing tool as shown in Table 3.1. This section summarizes the key components of the computational environment used throughout the study.

TABLE 3.1: Computational Environment Used for Simulation

| **Component** | **Details** |
|---|---|
| Operating System | Ubuntu 22.04 LTS |
| Python Version | 3.11 |
| Primary Framework | OpenMDAO 3.34.2 |
| Aircraft Sizing Tool | Aviary 0.9.3 |
| Numerical Libraries | NumPy, SciPy |
| Plotting / Visualization | Matplotlib |
| Data Handling | Pandas |
| Environment Management | Python Virtual Environment (venv) |
| Source Code Management | Git |
| Execution Platform | Intel Core i7 CPU, 16 GB RAM |
| Additional Notes | Simulations executed via custom scripts |

### 3.1.3 Design Constraints

Table 2.4 shows the design constraints of the homeland defense interceptor aircraft. Although several performance constraints are outlined in the design requirements including intercept radius, maximum Mach number, and various maneuvering capabilities this study focuses solely on one: the Defensive Counter-Air (DCA) Combat Air Patrol (CAP) endurance. Specifically, the aircraft is required to maintain a four-hour patrol at a distance of 300 nautical miles from base.

This constraint was chosen as the primary sizing driver not only because of its operational significance, but also due to current limitations in the AVIARY framework. At the time of this study, AVIARY does not support direct definition or enforcement of variables such as instantaneous turn rate, specific excess power, or load factor constraints, which prevented those from being included in the sizing analysis.

## 3.2 AVIARY Framework

In this study, Level 2 AVIARY framework are used and describes the overall aircraft sizing and mission analysis loop as shown in Figure 3.2. The process begins by preparing four major input components:

- **Aircraft Model (CSV file):**
  Defines the baseline aircraft configuration, including geometry, weight estimates, propulsion type, and aerodynamic properties.
- **Initial Weight and Aerodynamic Estimations:**
  Provides early estimates for takeoff gross weight $(W_0)$, empty weight, fuel weight, and basic aerodynamic characteristics such as lift-to-drag ratio $(L/D)$, based on historical data and preliminary calculations.
- **Design Mission Profile:**
  Specifies a mission sequence including different flight segments (e.g., climb, cruise, dash, loiter, combat) with associated performance requirements like altitude, Mach number, and load factors.
- **Energy Pack Estimation:**
  An initial estimate of propulsion system mass and fuel requirements based on mission energy needs.

Once these inputs are established, AVIARY conducts an Initial Sizing Block to predict major aircraft parameters such as takeoff weight, wing area, empty operating weight, and fuel consumption. This block also generates mission profile data plots and a performance summary.

Following initial sizing, a Constraint Analysis is performed to verify whether the aircraft meets all mission performance and weight requirements. If the constraints are not satisfied (e.g., insufficient fuel, inadequate range, or performance shortfalls), the aircraft is resized based on updated weights and design adjustments.

If the design passes the constraint checks, it proceeds to the AVIARY Mission Analysis, where detailed evaluation is performed through four modules:

- Aerodynamics
- Structure
- Propulsion

- Mission Analysis

Each analysis block updates and refines the aircrafts flight performance, fuel usage, structural mass distribution, and thrust capabilities.

Finally, a checkpoint evaluates whether the updated takeoff weight and mission performance metrics are satisfied. If they are, the sizing and analysis are complete. If not, the optimization loop continues until all mission and design constraints are fully met.

The outcome of this process is a final, feasible preliminary aircraft design that satisfies all mission requirements and forms the basis for further development.

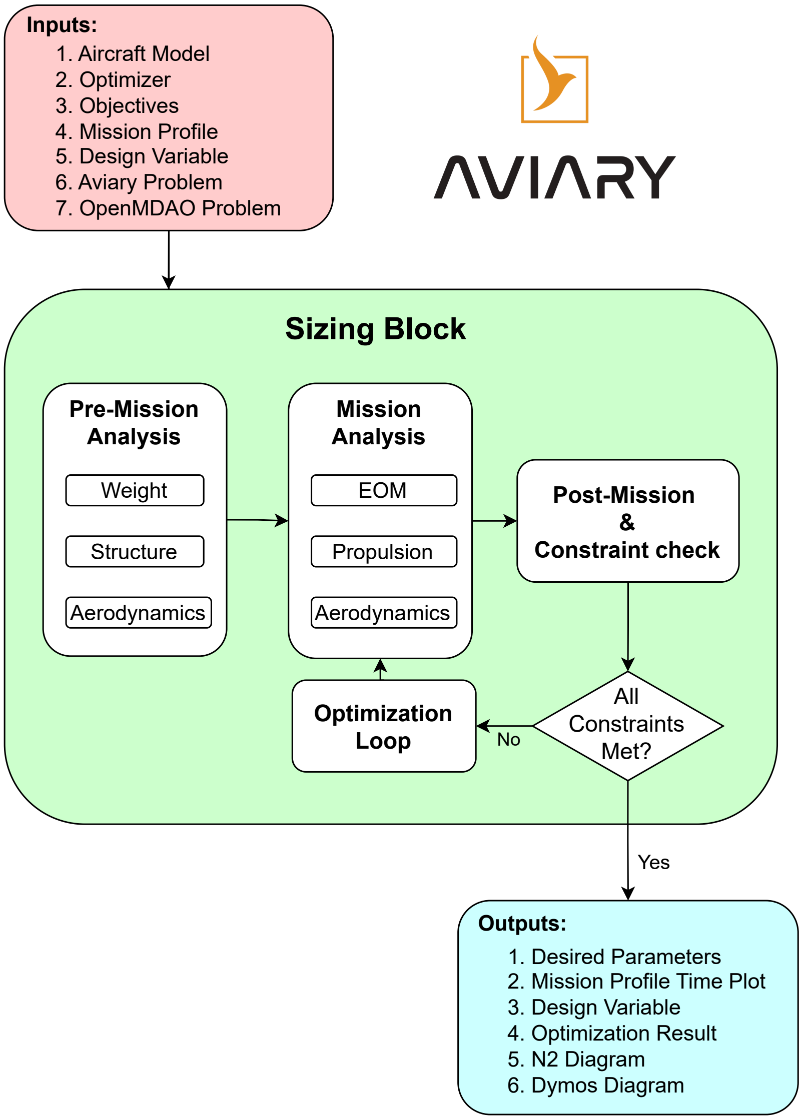



FIGURE 3.2: AVIARY Framework

## 3.3 Sizing Approach

The sizing methodology involved the following steps:

### 3.3.1 Benchmarking Aircraft Data Collection

To ensure the validity and relevance of the preliminary sizing process, a benchmark analysis was conducted using publicly available data from a variety of operational manned and unmanned combat aircraft. The goal of this study is to better understand the typical trends in key performance parameters specifically payload versus range, and thrust-to-weight ratio versus wing loading and to use these insights as a reference for validating the design outputs from the AVIARY tool.

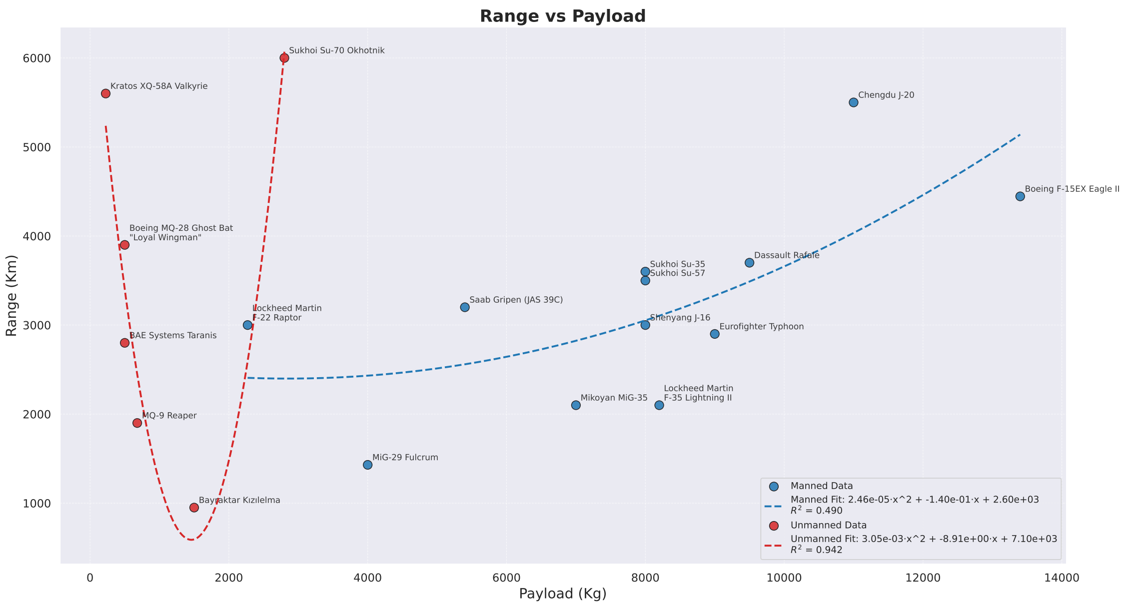



FIGURE 3.3: Range Vs Payload

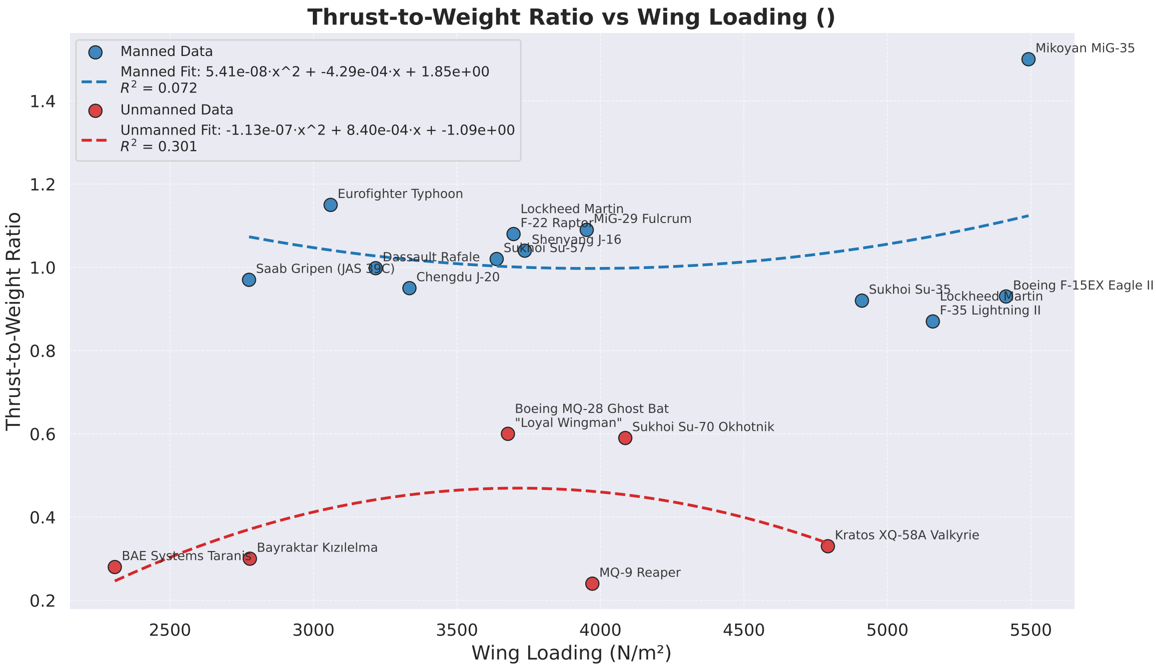



FIGURE 3.4: Thrust-to-Weight Vs Wing Loading

1. Payload vs Range
   The first comparison focuses on the trade-off between payload capacity and mission range. This relationship is fundamental in aircraft design, as it highlights how an increase in payload typically reduces the achievable range, and vice versa.
   As shown in the Figure 3.3, the data was separated into two groups: manned and unmanned aircraft. Each group was fitted with a second-order polynomial curve to represent the general trend.
   - For manned aircraft, the polynomial fit has the form:

     $$\text{Range} = 2.46 \times 10^{-5} \cdot \text{Payload}^2 - 0.140 \cdot \text{Payload} + 2600$$

     with an $R^2 = 0.490$, the value indicating a moderate correlation. This variability is expected, as manned aircraft often differ significantly in mission roles, payload configurations, and fuel planning strategies.

- In contrast, the unmanned aircraft data shows a much stronger and more predictable relationship:

$$\text{Range} = 3.05 \times 10^{-3} \cdot \text{Payload}^2 - 8.81 \cdot \text{Payload} + 7100$$

with a high value of $R^2 = 0.942$. This strong fit suggests that unmanned systems, often designed for specific roles like surveillance or precision strike, exhibit a more consistent payload-range behavior.

2. Thrust-to-Weight Ratio Vs Wing Loading
The second benchmark evaluates the relationship between thrust-to-weight ratio $(T/W)$ and wing loading (W/S). These parameters are critical in understanding an aircrafts maneuverability, climb performance, and takeoff/-landing characteristics.

- For manned aircraft, the polynomial fit is:

$$\text{T/W} = 5.41 \times 10^{-8} \cdot \text{W/S}^2 - 4.29 \times 10^{-4} \cdot \text{W/S} + 1.85$$

with the result of $R^2 = 0.072$, the value indicating significant spread in the data. This low correlation reflects the wide variety of mission profiles, aerodynamic layouts, and performance requirements among modern fighter jets.

- Unmanned aircraft exhibit a slightly stronger trend:

$$\text{T/W} = 1.13 \times 10^{-7} \cdot \text{W/S}^2 - 8.40 \times 10^{-4} \cdot \text{W/S} + 1.09$$

with $R^2 = 0.301$. While still showing variability, this result suggests a more standardized approach to propulsion and aerodynamic design in unmanned systems, likely due to their more specialized mission envelopes.

A comparative analysis of the selected four benchmark aircraft is shown in Table 3.2. This comparison highlights key performance metrics that are important as the baseline foundation for homeland defense interceptors design in this research.

TABLE 3.2: Aircraft Comparison Table

| Aircraft | Type | GTOW (lb) | Payload (lb) | T/W | W/S (lb/ft$^2$) |
|---|---|---|---|---|---|
| F-22 Raptor | Fighter | 64,840 | 5,004 | 1.08 | 77.22 |
| Saab Gripen (JAS 39C) | Fighter | 19,224 | 11,905 | 0.97 | 57.96 |
| Bayraktar Kzlelma | UCAV | 9,920 | 3,307 | 0.30 | 58.02 |
| MiG-29 Fulcrum | Fighter | 32,849 | 8,818 | 1.09 | 82.54 |

### 3.3.2 Mission Analysis

Table 3.3 shows the comparison of the three missions DCA Patrol, Point Defense Intercept, and Intercept/Escort highlights distinct operational priorities that directly influence the sizing strategy of the interceptor aircraft.

TABLE 3.3: Mission Requirement Comparison for Interceptor Aircraft

| Feature / Mission | DCA Patrol | Point Defense Intercept | Intercept / Escort |
|---|---|---|---|
| Mission Duration | Long (1+ hours patrol) | Short (quick reaction) | Medium to Long (escort profile) |
| Range | Medium | Short | Long |
| Fuel Requirement | High (loiter, 4 hours) | Low to Medium | Medium to High |
| Speed Requirement | Medium (cruise + dash) | High (fast intercept) | Medium to High |
| Weapon Load | Full AAM loadout | Interceptor missile load | Mixed AAM + external tanks |
| Design Priority | Endurance, awareness | Acceleration, climb rate | Flexibility, range, survivability |

By analyzing each mission's performance envelope, a multi mission-capable design can be derived, or a mission-optimized variant can be proposed based on which role is prioritized. The sizing method in AVIARY reflects these trade-offs through parametric input and trajectory-based mission analysis.

### 3.3.3 Baseline Analysis

- Study of existing interceptor aircraft such as the F-22 Raptor, Saab Gripen (JAS 39C), Bayraktar Kizilelma and MiG-29 Fulcrum.
- Benchmarking the Range, Payload, Thrust-to-Weight ratios and Wing Loading

### 3.3.4 Initial Sizing and Estimation

Using the insights from benchmarking and Raymer's analytical formulas and typical values, a first-pass sizing is completed. Parameters such as empty weight, fuel weight, and payload are integrated into the calculation of takeoff gross weight ($W_0$). Wing loading ($W/S$) and thrust-to-weight ratio ($T/W$) are then used to estimate wing area, wingspan, and required thrust.

Based on benchmark-informed inputs and preliminary equations, initial values are computed for:

- Takeoff Gross Weight ($W_0$)
- Wing Area (S)
- Thrust Required (T)
- Wingspan (b)
- Aspect Ratio (AR)
- Takeoff and Landing Max Lift Coefficient ($C_{Lmax}$)
- Takeoff $L/D$

These parameters serve as initial design foundation to feed into AVIARY's aircraft model input file.

### 3.3.5 Weight and Aerodynamics Estimation

1. Mission Weight Fractions
   Using Breguet formula which is given in Chapter 2 (Equation 2.12 and 2.11) and typical values to estimate the weight fraction for each segment as shown in Table 3.4.

   - **Cruise Out (300 nm)**

$$\begin{aligned} \frac{W_3}{W_2} &= \exp\left(-\frac{R \cdot c}{V \cdot \frac{L}{D}}\right) \\ &= \exp\left(-\frac{300 \cdot 0.8}{480 \cdot 8.7}\right) \\ &= 0.944 \end{aligned} \tag{3.1}$$

- **Cruise Back (400 nm)**

$$\begin{aligned} \frac{W_8}{W_7} &= \exp\left(-\frac{R \cdot c}{V \cdot \frac{L}{D}}\right) \\ &= \exp\left(-\frac{400 \cdot 0.8}{480 \cdot 8.7}\right) \\ &= 0.926 \end{aligned} \tag{3.2}$$

- **Escort (300 nm)**

$$\begin{aligned} \frac{W_8}{W_7} &= \exp\left(-\frac{R \cdot c}{V \cdot \frac{L}{D}}\right) \\ &= \exp\left(-\frac{300 \cdot 0.7}{270 \cdot 10}\right) \\ &= 0.925 \end{aligned} \tag{3.3}$$

- **Combat Air Patrol (Loiter)**

$$\begin{aligned} \frac{W_4}{W_3} &= \exp\left(-\frac{E \cdot c}{\frac{L}{D}}\right) \\ &= \exp\left(-\frac{4 \cdot 0.8}{10}\right) \\ &= 0.786 \end{aligned} \tag{3.4}$$

2. Total Mission Weight Fraction

   - **DCA Patrol Mission:**
     The total mission weight fraction is calculated as:

$$\begin{aligned} \left(\frac{W_{10}}{W_0}\right) &= 0.990 \times 0.985 \times 0.944 \times 0.786 \times 0.977 \\ &\quad \times 0.980 \times 0.990 \times 0.926 \times 0.995 \times 0.980 \\ &= 0.619 \end{aligned} \tag{3.5}$$

   - **Point Defense Intercept Mission:**

TABLE 3.4: Weight Fraction for each segments

| Segment | Weight Fraction |
|---|---|
| Warm-up and Takeoff | 0.990 |
| Climb | 0.985 |
| Cruise (300 nm) | 0.944 |
| CAP (Loiter) | 0.786 |
| Escort (300nm) | 0.925 |
| Dash | 0.977 |
| Combat maneuver | 0.980 |
| Climb (post-maneuver) | 0.990 |
| Cruise (400 nm) | 0.926 |
| Descent | 0.995 |
| Reserve | 0.980 |

The total mission weight fraction is calculated as:

$$\begin{aligned}\left(\frac{W_8}{W_0}\right) &= 0.990 \times 0.985 \times 0.977 \times 0.980 \times 0.990 \\ &\quad \times 0.950 \times 0.995 \times 0.980 \\ &= 0.856\end{aligned} \tag{3.6}$$

- **Intercept/Escort Mission:**
  The total mission weight fraction is calculated as:

$$\begin{aligned}\left(\frac{W_8}{W_0}\right) &= 0.990 \times 0.985 \times 0.977 \times 0.925 \times 0.990 \\ &\quad \times 0.970 \times 0.995 \times 0.980 \\ &= 0.825\end{aligned} \tag{3.7}$$

3. Fuel Fraction
   By using Raymer's equation given in Chapter 2 Equation 2.2 to obtain each mission fuel fraction as shown in Table 3.5.
4. Takeoff Weight ($W_0$)
   To estimate the takeoff weight using Raymer's formula given in Chapter 2 Equation 2.1, let's assume some parameters which are:

TABLE 3.5: Weight Fraction for each segments

| **Mission** | **Weight Fraction** | **Fuel Fraction** |
|---|---|---|
| DCA Patrol | 0.619 | 0.40 |
| Point Defense Intercept | 0.856 | 0.15 |
| Intercept/Escort | 0.825 | 0.18 |

- Payload = 802 lb
- Crew = 300 lbm
- Empty weight fraction = 0.57 (Using Nicolai's typical value **nicolaiFundamentalsAircr**
- Fuel weight fraction = 0.40 (DCA Patrol)

$$\begin{aligned} W_0 &= \frac{300 + 802}{1 - (0.61 + 0.36)} \\ &= 36,733 \text{ lbm } (16,661 \text{ Kg}) \end{aligned} \tag{3.8}$$

The result above shows for DCA Patrol mission Takeoff weight estimation which will be served as baseline sizing to be used in other mission profiles. Theoretically, Point Defense Interceptor and Escort Mission will have lighter Gross Takeoff weight that DCA Patrol Mission. As shown the result in Table 3.5, DCA Patrol has bigger fuel fraction than other two missions.

5. Fuel Weight

In this study, the estimation of fuel weight was carried out using a fuel fraction method, widely used in preliminary aircraft sizing. Rather than modeling fuel consumption continuously, this method uses empirically derived fuel weight fractions from Table 3.5 to estimate the total fuel required.

$$\begin{aligned} W_f &= W_0 \times \left(\frac{W_f}{W_0}\right) \\ &= 36,733 \times 0.40 \\ &= 14,693.2 \text{ lbm } (6,664.7 \text{ Kg}) \end{aligned} \tag{3.9}$$

This fuel weight serves as the baseline sizing for the DCA Patrol mission, with the assumption that other missions (Point Defense Intercept and Intercept/Escort) will result in slightly lower fuel requirements due to their lighter fuel fractions.

6. Lift-to-Drag Ratio

   The Lift-to-Drag ratio (L/D) is estimated based on the intended mission type, configuration, and benchmarking with existing aircraft of similar class. Rather than detailed aerodynamic modeling, typical values from historical data are used.

   For subsonic interceptor aircraft with moderate sweep and clean configuration, L/D values during cruise typically range between 8 and 12. In this study, an average cruise L/D value of 8.6 was assumed based on analogous platforms such as the Saab Gripen and MiG 29, as well as guidance from Raymer and Nicolai approach.

   For loiter or CAP (Combat Air Patrol) segments, the aircraft may operate at lower altitudes and optimized configurations, thus slightly increasing L/D. A value of 10.0 was assumed for the loiter phase.

   These assumptions are summarized below:

   TABLE 3.6: Estimated L/D

| Mission Phase | Estimated L/D |
|---|---|
| Cruise | 9.0 |
| Loiter (CAP) | 10.0 |
| Dash Segment | 7.0–8.0 |

   These values shown in Table 3.6 were used directly in the Breguet range and endurance equations to estimate mission fuel fractions.

7. $C_{L_{\max}}$ Takeoff and Landing

   In this study detailed airfoil data is not yet available, so typical values are estimated based on historical data. For a subsonic interceptor with a low-mounted wing and military-style high-lift devices, the following typical values were assumed:

   TABLE 3.7: Typical $C_{L_{\max}}$ Values for Takeoff and Landing

| Flight Phase | Configuration | Estimated $C_{L_{\max}}$ |
|---|---|---|
| Takeoff | Flaps/slats deployed | 1.7 - 2.0 |
| Landing | Full high-lift configuration | 2.2 - 2.6 |

The following values shown in Table 3.7 were based on Raymer **raymerAircraftDesignConc**
and Nicolai **nicolaiFundamentalsAircraftAirship2010**.

TABLE 3.8: Estimated $C_{L_{\max}}$ Values for Takeoff and Landing

| **Flight Phase** | **Estimated $C_{L_{\max}}$** |
|---|---|
| Takeoff | 1.9 |
| Landing | 2.4 |

Table 3.8 shows the estimated value of $C_{L_{\max}}$ that are used as baseline sizing value to be inputted to AVIARY.

8. Thrust-to-weight Ratio
   In this study, an initial T/W of 0.98 was selected for baseline sizing, considering the aircraft must achieve rapid climb, efficient patrol, and potential dash phases. This value was derived from historical benchmarking against reference aircraft such as the Saab Gripen and MiG 29 Fulcrum, and adjusted for subsonic operation.
   Typical values from Nicolai **nicolaiFundamentalsAircraftAirship2010** for subsonic to transonic interceptors are:
   - Combat configuration: $T/W$ = 0.80 to 1.10
   - Cruise/loiter sizing: $T/W$ = 0.30 to 0.5
9. Wing Loading
   In this study, a baseline wing loading of 82.54 lb/ft$^2$ was selected. This value was derived from benchmarking aircraft shown in Table **??** the Saab JAS 39C Gripen and similar subsonic-interceptor class aircraft.
10. Wing and Thrust Sizing
    To estimate the wing area and required thrust for the homeland defense interceptor aircraft, use the wing loading and thrust-to-weight ratio value from the benchmark plot that shown in Figure 3.4. Let's try using Mig-29 Fulcrum's wing loading and thrust-to-weight ratio value where $W/S$ = 82.54 lb/ft$^2$ and $T/W$ = 0.98. We will be using Raymer's common formula:

(a) Wing Area

$$\begin{aligned} S &= \frac{W_0}{W/S} \\ &= \frac{36,733}{82.54} \\ &= 445.03 \text{ ft}^2 \ (41.34 \text{ m}^2) \end{aligned} \tag{3.10}$$

(b) Required Thrust

$$\begin{aligned} T &= W_0 \times T/W \\ &= 26,733 \times 0.98 \\ &= 35,998 \text{ lbf} \ (160.12 \text{ kN}) \end{aligned} \tag{3.11}$$

Using the 2 engine configuration, with 17,999 lbf each.

### 3.3.6 Geometry Estimation

1. Wing Span and Aspect ratio
   For a subsonic interceptor, optimized for loiter and agility but not high-altitude efficiency, according to Raymer's the typical value of the aspect ratio for such aircraft is 3.5 - 4. Let's try using AR = 3.98. Using the fundamental equation from Raymer's book **raymerAircraftDesignConceptual2006a**:

$$\begin{aligned} AR &= \frac{b^2}{S} \\ b &= \sqrt{AR \times S} \\ b &= \sqrt{3.98 \times 445.03} \\ b &= 42.08 \text{ ft} \ (12.82 \text{ m}) \end{aligned} \tag{3.12}$$

2. Horizontal and Vertical Tail

   (a) Horizontal Tail Area

$$\begin{aligned} S_h &= \frac{V_h \times S \times c}{l_h} \\ &= \frac{0.9 \times 445.03 \times 10.72}{27.75} \\ &= 154.73 \text{ ft}^2 \ (14.37 \text{ m}^2) \end{aligned} \tag{3.13}$$

(b) Vertical Tail Area

$$\begin{aligned} S_v &= \frac{V_v \times S \times b}{l_v} \\ &= \frac{0.08 \times 445.03 \times 42.08}{27.75} \\ &= 53.98 \text{ ft}^2 \ (5.10 \text{ m}^2) \end{aligned} \tag{3.14}$$

### 3.3.7 Initial Parameters Used in AVIARY

This sizing step serves as the essential input layer in the full AVIARY modeling chain. The values calculated in this phase are inserted into the Aircraft Model CSV input file, where they define the baseline geometry, mass properties, and propulsion configuration. These inputs are subsequently used in mission simulation, constraint analysis, and performance evaluation throughout the remainder of the design process.

Table 3.9 presents the results of the initial weight and aerodynamic parameter estimation for the proposed homeland defense interceptor aircraft. These values were derived using conceptual sizing methods based on Raymer and Nikolai approach.

TABLE 3.9: List of Weight, Aerodynamics and Geometry initial parameters Estimation

| **Initial Parameters** | |
|---|---|
| **Parameter** | **Value** |
| $W_0$ (lb/kg) | 36,733 / 16,661 |
| $W_{\text{empty}}$ (lb/kg) | 20,937 / 9,496 |
| Fuel Weight (lb/kg) | 14,693 / 6,664 |
| Payload (lb/kg) | 802 / 363.78 |
| Wing Area (ft$^2$) | 445.03 / 41.34 |
| Wing Span (ft) | 42.08 / 12.83 |
| Aspect Ratio | 3.98 |
| Horizontal Tail Area (ft$^2$) | 154.73 |
| Vertical Tail Area (ft$^2$) | 53.98 |
| Thrust-to-Weight ($T/W$) | 0.98 |
| Wing Loading (lb/ft$^2$) | 82.54 |
| Thrust Needed (lbf/kN) | 35,998 / 160.12 kN |

These values were estimated using Raymer's empirical weight fraction methods and aerodynamic relationships, and implemented in OpenMDAO to automate parameter estimation. The takeoff gross weight ($W_0$) accounts for the complete operational mass of the aircraft at mission start, while the empty weight ($W_{\text{empty}}$) includes structural and onboard system components, excluding payload and fuel.

The fuel weight is derived from the mission profile and includes sufficient capacity for loiter, combat, dash, and return-to-base phases. The payload reflects the expected operational equipment and weapon systems for an interceptor role.

Aerodynamically, the aircraft was configured with a wing area of 445.03 $\text{ft}^2$ and a span of 42.08 ft, resulting in an aspect ratio of 3.98. This choice reflects typical supersonic aircraft characteristics, balancing low drag at high speed with structural compactness.

Performance parameters include a thrust-to-weight ratio of 0.98, which supports high acceleration and vertical performance, and a wing loading of 82.54 $\text{lb/ft}^2$, indicating emphasis on speed and penetration capability. The required thrust is estimated at 35,998 lbf, which informs the propulsion system selection and validates feasibility against existing engine models.

### 3.3.8 Engine Deck Selection

Utilized the engine deck model provided by AVIARY as shown in Appendix B B.7 to simulate thrust and specific fuel consumption (SFC) across various flight conditions. This pre-built engine model enabled rapid integration into the simulation framework, ensuring accurate representation of propulsion system performance for trade studies.

### 3.3.9 Energy Pack Weight Estimation

This stage focuses on the propulsion and fuel system sizing:

- Number of engines and their thrust rating.
- Engine mass based on thrust class.
- Fuel tank capacity requirements to support mission fuel loads and reserves.

Energy pack weight is critical for ensuring that the HDI can sustain long-endurance patrols and high-speed interceptions without exceeding structural limits.

### 3.3.10 Optimization with AVIARY

The entire aircraft model is simulated within the AVIARY framework. Inputs include:

- Weight breakdown
- Aerodynamics and geometry
- Propulsion data
- Mission phase definitions (climb, cruise, dash, loiter, combat)
- Performance evaluation in terms of trajectory result for each mission simulation.

AVIARY evaluates whether the aircraft can complete the mission within all constraints. If the mission is completed successfully, the design is considered feasible. If not, it progresses to the next step.

### 3.3.11 Constraint Check

After running simulations, a constraint evaluation is performed:

- Does the aircraft complete the mission within fuel and energy limits?
- Is the dash speed and endurance satisfied?
- Are altitudes requirements met?

If any constraint is not fulfilled, the design proceeds to resizing.

### 3.3.12 Resizing and Iteration

If performance requirements are not met, resizing occurs. This may involve:

- Adjusting wing area to improve lift and endurance
- Adjusting the altitude speed and duration of each segment to fulfill the missions.
- Increasing engine thrust for better climb and dash.

- Enlarging fuel tanks for extended range.

The updated configuration is re-simulated in AVIARY until all mission requirements are met.

### 3.3.13 Trade Studies and Analysis

Once a feasible design is achieved, additional trade studies can be performed to optimize:

- Aerodynamic performance (e.g., exploring different wing area, wing span and aspect ratios).
- Weight savings through structural material choices.
- Alternate propulsion options for improved fuel efficiency.

### 3.3.14 Preliminary Sizing Setup with AVIARY

The preliminary sizing process in this thesis utilizes the Level 2 on boarding structure of the AVIARY framework, as defined in the official documentation **Level2**. This setup provides a more flexible and customizable interface compared to the Level 1 approach, allowing for deeper integration of aircraft design parameters into the simulation workflow. All coding setup also available on my Gitlab Repository.

The steps taken in this setup to build and run the Level 2 mission model were as follows:

1. **Step 1: Aircraft Model in CSV file**
   The aircraft model used in this study is structured in a FLOPS legacy format, containing the essential input parameters required by the Aviary framework. This parameters include Gross Takeoff weight ($W_O$), wing area ($S$), thrust-to-weight ratio ($T/W$), and wing loading ($W/S$), payload, aspect ratio (AR) and other parameters were computed into Aviary Aircraft Model as shown in Appendix A A.1.
2. **Step 2: Insert Phase Info**
   Insert phase info that identifies the mission profile information of each mission (DCA Patrol, PDI and Escort) into the python script. In Appendix B

B.11,B.12 and B.13 shows the break down of each mission segments such as takeoff, climb, combat air patrol (CAP), dash, combat maneuver, escort, reserve and descent.

AVIARYs mission analysis framework does not explicitly identify or model distinct mission phases such as dash, acceleration, combat-air-patrol, combat, and escort as separate segments. Instead, these mission profiles are approximated or simplified by treating them as variations of cruise segments. This means that the detailed dynamic behaviors specific to each phasesuch as rapid acceleration during dash, maneuvering during combat, or loitering during air patrolare not individually captured. Rather, the performance characteristics (like speed, altitude, and fuel consumption) for these phases are represented using cruise segment parameters that approximate average or typical conditions.

This approach simplifies the mission modeling within AVIARY by using cruise segments as proxies, enabling the sizing and performance analysis to proceed without requiring complex transient or maneuver-specific modeling. However, it also means that the nuances and specific aerodynamic or propulsion demands of each distinct phase are generalized, which may affect the precision of performance predictions for highly dynamic mission profiles.

3. **Step 3: Add Aviary Run Mission arguments**

   In Aviary's Level 2 interface, the run-mission command provides a flexible way to execute mission analyses with customizable parameters. This command is typically used in a Python environment and allows users to specify various arguments to tailor the simulation to their specific needs as shown in Appendix B B.14. The optimization solver selected for this simulation is IPOPT. `IPOPT` is well-suited for large-scale nonlinear programming problems and provides robust convergence behavior in the presence of complex constraints, which are common in mission trajectory analyses. It supports sparse matrix operations and barrier methods, making it ideal for optimizing aircraft performance over multiple mission phases.

   The `COLLOCATION` scheme is chosen as the numerical method to solve the dynamic equations of the mission. This method discretizes the time domain into segments and approximates the state and control trajectories using polynomial basis functions. It is particularly advantageous for missions involving rapid changes in state variables, such as climb and descent. The collocation method offers greater numerical stability and solution accuracy compared to simpler schemes like shooting, especially when dealing with stiff or tightly coupled dynamic systems.

The objective type parameter is set to None, meaning AVIARY set objective to `Mission.Objectives.FUEL` which is Sizing. This configuration is intentionally used during preliminary simulation stages to perform a baseline mission analysis. It allows observation of how the aircraft behaves under nominal conditions using fixed mission profiles, without the influence of an optimization target.

The `aviary_history.db`, indicates the simulation retains a detailed log of state variables, control inputs, fuel usage, and phase transitions throughout the mission. This recorded data can later be visualized or used for post-processing analysis to evaluate aircraft performance, identify potential inefficiencies, and support design refinement.

In this simulation, no restart file is used. Setting restart-filename = None indicates that the simulation is executed from scratch rather than continuing from a previous run. This decision is appropriate when conducting initial runs or when modifying mission configurations, as it avoids inheriting any prior state that might bias the analysis.

Lastly, the maximum number of optimizer iterations is deliberately limited to 5. This low iteration count is chosen for debugging and validation purposes, allowing rapid feedback on the setup without committing to long run times.

4. **Step 4: Insert Aviary Analysis Problem**

   The `AviaryProblem` class in the AVIARY framework is used to define, configure, and execute aircraft mission simulations. The following ordered sequence of method calls represents the complete process of setting up and running an aircraft mission using AVIARY as shown in Appendix B B.15.

5. **Step 5: Add the Design Variables**

To perform parametric studies and potential optimization of the interceptor aircraft mission profile, several design variables were introduced using the add-design-var method from the AVIARY framework. These variables were selected based on their direct impact on aircraft performance, mission feasibility, and aerodynamic behavior. Each variable is defined with a specified range (lower, upper) and a reference value (ref) to guide scaling during optimization or sensitivity analysis as shown in Appendix B B.16. The variables and their purposes are described as follows:

- `av.Mission.Design.GROSS MASS`
  This variable represents the takeoff gross weight of the aircraft. It was allowed to vary between 32,000 and 37,000 pounds. This range reflects the expected operational envelope of a lightweight interceptor class aircraft. The reference value is set at 36,733 lb, based on preliminary weight estimation results. Gross mass significantly influences fuel consumption, required thrust, and overall aircraft size.
- `av.Aircraft.Wing.AREA`
  The wing platform area is a critical aerodynamic parameter that affects lift generation, wing loading, and stall speed. The area is allowed to vary between 400 and 500 ft$^2$, with a reference value of 445.03 ft$^2$, consistent with initial sizing outputs. This variable directly affects both takeoff and landing performance.
- `av.Aircraft.Wing.SPAN`
  The wingspan defines the lateral extent of the wing and contributes to aspect ratio, induced drag, and structural weight. The span is constrained between 43 and 50 feet. A reference span of 42.08 ft reflects the configuration from baseline sizing.
- `av.Mission.Design.RANGE`
  The design range is the mission-specific combat radius or operational range. Here, it is varied between 800 and 1100 nautical miles, with a nominal reference of 1000 NM. This range bracket supports scenarios from short-range interception to regional patrol missions.
- `av.Aircraft.Wing.ASPECT RATIO`
  The aspect ratio governs aerodynamic efficiency, especially during cruise.

A range of 3 to 5 is specified, with a baseline value of 3.98. This value represents a balance between structural feasibility and low induced drag, which is suitable for a supersonic-capable airframe.

- `av.Aircraft.HorizontalTail.AREA`
  The horizontal tail area influences longitudinal stability and control authority. It is constrained between 150 and 160 ft$^2$, with a nominal reference of 154.73 ft$^2$, based on tail volume coefficient estimation for a tailed configuration.
- `av.Aircraft.VerticalTail.AREA`
  The vertical tail area governs yaw stability and directional control. The area is allowed to vary from 52 to 55 ft$^2$, with a reference of 53.98 ft$^2$. This design space ensures adequate stability across maneuvering and high-angle-of-attack flight regimes.
- `av.Mission.Takeoff.LIFT OVER DRAG`
  This variable represents the aircrafts lift-to-drag ratio during the take-off phase. A higher value typically indicates better aerodynamic efficiency during ground roll and initial climb. The allowed range is 3 to 4, with a reference of 3.5. This variable indirectly influences required thrust and takeoff distance.
- `av.Mission.Takeoff.LIFT COEFFICIENT MAX`
  The maximum lift coefficient during takeoff affects stall speed and minimum takeoff distance. It is varied between 1.6 and 2.0, with a reference value of 1.8, representing moderate lift enhancement through leading-edge devices or flaps.
- `av.Mission.Landing.LIFT COEFFICIENT MAX`
  Similar to takeoff, this variable governs maximum lift during the landing phase. A higher $C_{L_{\max}}$ allows for slower approach speeds and shorter landing distances. The range is set from 1.8 to 2.5, with a nominal value of 2.2, assuming the presence of high-lift systems such as trailing-edge flaps.

6. **Step 6: Print Desired Variables**

After running the optimization simulation using AVIARY, the last step involves retrieving the computed parameters that define the aircrafts geometry, performance, and mission-specific metrics as shown in Appendix B B.17.

## 3.4 Limitations

- The aerodynamic and propulsion models rely on empirical approximations.
- Real-world operational factors such as atmospheric disturbances are not modeled.
- The scope is limited to preliminary design; detailed system-level analysis is excluded.
- Supersonic performance is not analyzed, as the design focuses exclusively on subsonic regimes.

# CHAPTER 4
# RESULTS AND DISCUSSIONS

## 4.1 Introduction

This chapter presents the results of the preliminary sizing of the homeland defense interceptor aircraft using AVIARY. The discussion emphasizes the critical design parameters, performance characteristics, and their alignment with the mission objectives. Key findings are analyzed in the context of design goals, ensuring clarity and relevance to the objectives of the study.

## 4.2 Aircraft Sizing Results

The refined parameters were tested using AVIARY, producing the following results for each mission:

TABLE 4.1: Sizing Results for Different Missions

| **Parameter** | **Initial** | **DCA Patrol** | **Point Defense** | **Intercept/Escort** |
|---|---|---|---|---|
| Gross Weight (lb) | 36,733 | 36,066 | 35,905 | 36,000 |
| Range (nmi) | 800 | 800 | 500 | 500 |
| Wing Area (ft$^2$) | 445.03 | 486.61 | 486.42 | 486.50 |
| Wing Span (ft) | 42.08 | 42 | 42 | 42 |
| Aspect ratio | 3.98 | 3.98 | 3.98 | 3.98 |
| Horizontal Tail Area (ft$^2$) | 153.46 | 153.46 | 153.46 | 153.46 |
| Vertical Tail Area (ft$^2$) | 53.98 | 53.98 | 53.98 | 53.98 |
| Thrust Required (lbf) | 35,998 | 35,500 | 35,500 | 35,500 |
| $C_{L_{Max}}$ Takeoff | 1.8 | 1.8 | 1.8 | 1.8 |
| $C_{L_{Max}}$ Landing | 2.5 | 2.5 | 2.49 | 2.5 |
| Take-off L/D | 3.55 | 3.55 | 3.55 | 3.55 |

Table 4.1 shows the changes of the HDI design from its initial sizing through three key mission profiles: DCA Patrol, Point Defense, and Intercept/Escort. Each

mission scenario introduces small yet meaningful refinements to the aircrafts configuration to optimize it for specific operational roles.

The initial gross weight of 36,733 lb was refined to 36,066 lb for the DCA Patrol mission and slightly reduced for Point Defense (35,905 lb) and Intercept/Escort (36,000 lb). These changes reflect on fuel differences across all three mission. The wing area increased from 445.03 $ft^2$ in the initial design to approximately 486.5 $ft^2$ across all three mission-specific configurations. A slight reduction in thrust required is seen from 35,998 lbf initially to 35,500 lbf in the final designs. Other parameters remain unchanged across all mission profiles.

## 4.3 Analysis

The configuration adjustments shows in Table 4.1 made during the mission-specific sizing process demonstrate a strategic refinement of the HDI design, leveraging the modular capability of the AVIARY tool to adapt the aircraft for varying operational demands without a complete redesign.

The increase in wing area across all missions reflects a deliberate attempt to improve the aerodynamic efficiency and maneuverability of the aircraft, particularly at lower speeds during takeoff and landing. This change resulted in a reduced wing loading, which benefits overall handling qualities and contributes to the aircrafts operational flexibility.

The gross weight adjustments were relatively minor but purposeful. For shorter-range missions like Point Defense and Intercept/Escort, fuel capacity and onboard systems were optimized to reduce unnecessary weight, resulting in improved climb rates and potentially lower fuel consumption. These reductions in weight directly influenced the thrust requirement, allowing for a drop of nearly 500 lbf while maintaining performance goals.

It is important to note that the aerodynamic coefficients and tail sizing remained unchanged throughout the mission profiles. This consistency suggests that the original aerodynamic assumptions were robust and that the control surface sizing was sufficient across all considered mission conditions.

Overall, these adjustments confirm that the initial design was well-calibrated, requiring only minor refinements to meet the distinct performance needs of various homeland defense scenarios. The ability to tailor the aircraft for each mission while preserving aerodynamic integrity and propulsion efficiency illustrates the flexibility and power of AVIARYs mission-based design framework, reinforcing its suitability for conceptual and preliminary aircraft design.

## 4.4 Comparison with Selected Aircraft

TABLE 4.2: Comparison of HDI Design and Selected Aircraft

| Parameter | HDI Design | F-22 Raptor | Gripen JAS 39C | Bayraktar Kzlelma | MiG-29 Fulcrum |
|---|---|---|---|---|---|
| Gross Weight (lb) | 36,066 | 64,860 | 19,224 | Not Available | 32,849 |
| Wing Area (ft$^2$) | 486.61 | 840 | 320 | Not Available | 409 |
| Wing Span (ft) | 42 | 44.5 | 27.6 | 32 | 37.3 |
| Aspect Ratio | 3.98 | 2.35 | 2.35 | Not Available | 3.4 |
| Thrust (lbf) | 35,500 | 70,000 | 18,100 | 9,900 | 36,660 |
| Range (nmi) | 800 | 1,839 | 810 | 500 | 890 |
| Thrust-to-Weight ($T/W$) | 0.98 | 1.08 | 0.97 | Not Available | 1.09 |
| Wing Loading (lb/ft$^2$) | 74.21 | 77.2 | 58 | Not Available | 83 |

Table 4.2 presents a comparative analysis between the Homeland Defense Interceptor (HDI) design and the selected aircraft, namely the F-22 Raptor, Gripen JAS 39C, Bayraktar Kzlelma, and MiG-29 Fulcrum. The comparison highlights the key sizing and performance parameters such as gross weight, wing area, thrust, range, and aerodynamic ratios.

The HDI exhibits a gross weight of 36,066 lb, placing it between the heavier F-22 (64,860 lb) and the lighter Gripen (19,224 lb). With a wing area of 486.61 ft$^2$ and a wingspan of 42 ft, the HDIs design is optimized for subsonic performance and low-to-moderate wing loading. The aspect ratio of 3.98 is notably higher than those of the F-22 (2.35) and MiG-29 (3.4), indicating enhanced aerodynamic efficiency and favorable loitering performance.

In terms of propulsion, the HDI requires 35,500 lbf of thrust, yielding a thrust-to-weight ratio of 0.98, which is similar to the Gripen (0.97) and just below the MiG-29 (1.09) and F-22 (1.08). This ratio supports rapid climb and acceleration performance, suitable for point-interception and patrol roles. The range of 800 nmi is sufficient for homeland defense missions, falling between the Gripen (810 nmi) and MiG-29 (890 nmi), though well below the F-22s extended 1,839 nmi reach.

The wing loading of 74.21 lb/ft$^2$ places the HDI in a moderate range lower than the MiG-29 (83 lb/ft$^2$) and F-22 (77.2 lb/ft$^2$), yet higher than the Gripen (58 lb/ft$^2$). This balance reflects a compromise between high-speed performance and maneuverability.

Figure 4.1 provides a performance envelope plot of various combat aircraft, plotting thrust-to-weight ratio ($T/W$) against wing loading (W/S) in SI units (N/m$^2$). The plot includes both manned fighters (e.g., F-22, F-35, J-20, MiG-35, Rafale, Typhoon) and unmanned combat air vehicles (UCAVs) (e.g., Bayraktar Kzlelma, MQ-28 Ghost Bat, Su-70 Okhotnik). Regression trends are fitted separately for manned and unmanned platforms.

The HDI design falls squarely within the performance envelope of modern manned multirole fighters, with a wing loading of approximately 3615 (N/m$^2$) and a thrust-to-weight ratio near 0.98. Its proximity to aircraft such as the Gripen JAS 39C and MiG-29 Fulcrum demonstrates that the HDI was sized for similar performance expectations: high responsiveness, efficient subsonic cruise, and moderate loiter capability.

The manned aircraft trendline exhibits a wide spread in values with a weak positive correlation, while unmanned aircraft tend to cluster in lower thrust and lower wing loading zones, optimized for endurance rather than speed or agility. The HDIs location away from the UCAV trend confirms that it was designed to fulfill interceptor-level mission agility and climb performance, not endurance-focused surveillance or loitering.

## 4.5 Aviary Mission Performance Analysis

The Aviary mission performance analysis provides a detailed look into the aircraft's behavior across different mission segments.

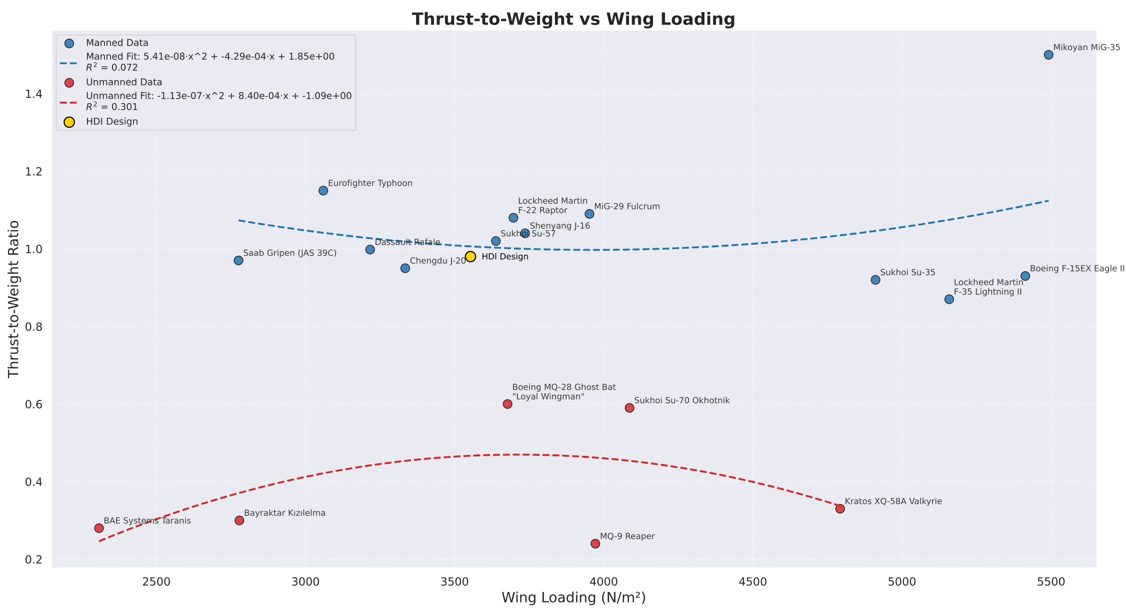



FIGURE 4.1: T/W Vs W/S of HDI Design

### 4.5.1 DCA Patrol Mission Analysis

The highest rate of mass loss occurs during the climb phase as shown in Figure 4.2, which is expected due to high engine thrust. In cruise segment show relatively stable mass with minimal fuel consumption, likely due to a simulation of dynamic maneuvering. The climb phase exhibits a high energy gain rate, demonstrating the aircrafts strong thrust-to-weight ratio. Cruise phases show low and stable energy rates, reflecting efficient level flight. Spikes in energy rate during certain cruise segments likely correspond to maneuvering or altitude changes. Thrust levels are high during climb. Thrust spikes in cruise segments indicate increased power demands for acceleration or maneuvers.

### 4.5.2 PDI Mission Analysis

Mass decreases from approximately 16,250 kg to 15,000 kg as shown in Figure 4.5. The rate of mass decrease is lower in cruise than climb, showing improved fuel efficiency at cruising altitude. Drag remains low and steady during cruise, indicating efficient aerodynamics. In Figure 4.6, a sharp drag increase in the cruise1 segment due to speed changes. Fuel consumption is lower in cruise than

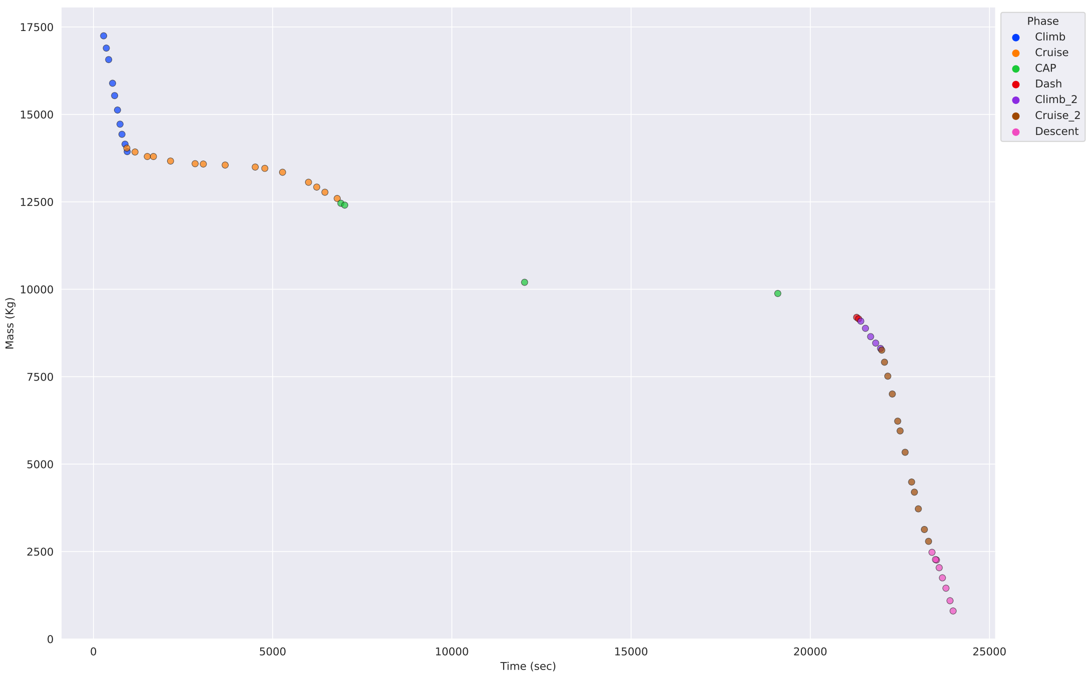



FIGURE 4.2: DCA Patrol: Mass over time

climb but remains relatively high to maintain altitude and speed. Increased negative fuel flow rate during descent1 may relate to thrust modulation for descent control as shown in Figure 4.7. Thrust varies; increases in cruise2 and descent2 segments are attributed to maneuvers and stability maintenance.

### 4.5.3 Escort Mission Analysis

The Escort segment stands out as the longest and most telling part of the mission, lasting from around 2,000 to 8,000 seconds as shown in Figure 4.8. Throughout this stretch, the aircraft's weight holds steady at roughly 14,000 kg. This consistency points to an efficient use of fuel, suggesting the aircraft is well-suited for extended loitering an essential trait for escort operations. After the Escort segment, there's a sharp drop in mass during the Climb2 phase , falling from 14,000 kg to about 11,500 kg between 8,500 and 9,500 seconds. This rapid decrease likely reflects a burst of fuel use for gaining altitude.

As shown in Figure 4.9 Climb 1 and Dash phases from 0 to approximately 2000 seconds, the drag force fluctuates, starting at higher values around 6-7 lbf

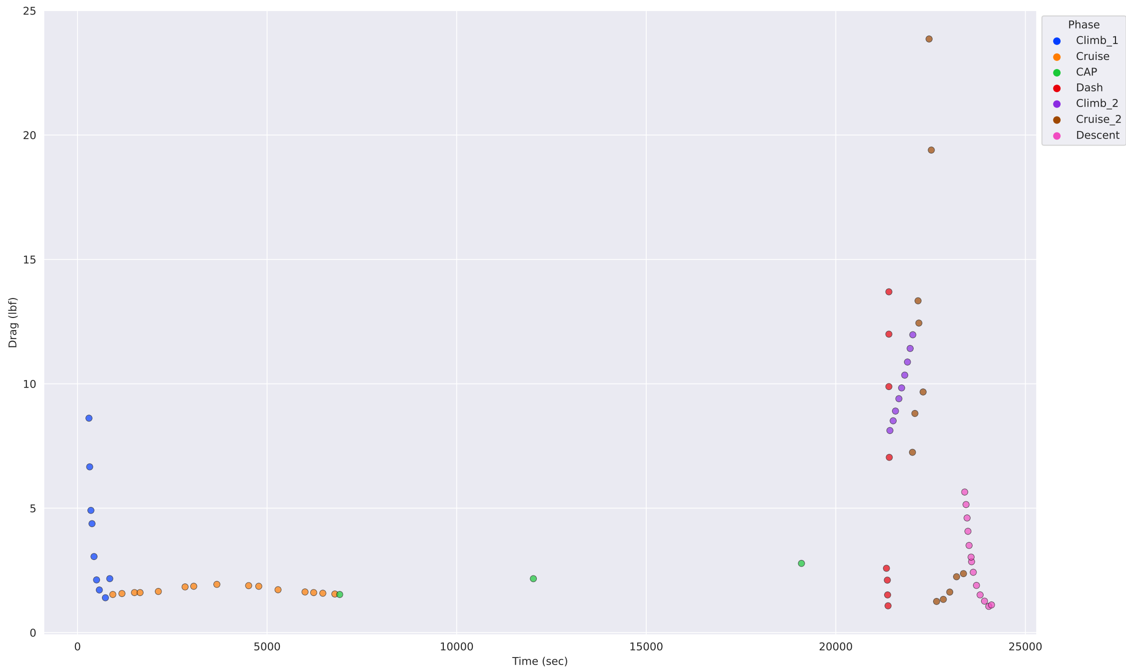



FIGURE 4.3: DCA Patrol: Drag over time

and then settling to lower levels approximately around 2 lbf. These fluctuations are expected during the dynamic nature of takeoff, initial climb, and acceleration. During escort phase is characterized by remarkably low and stable drag values, consistently hovering around 2 lbf in a duration from 2000 to 7500 seconds. This indicates that the aircraft is operating at an aerodynamically efficient condition, which is crucial for maximizing endurance during an escort mission.

In Climb 1 segment, approximately from 0 to 2000 seconds, the specific energy rate excess is initially high around 100 m/s and then decreases as the aircraft transitions out of the initial climb, stabilizing at lower values as shown in Figure 4.10. Then it show increase energy rate during Dash segment due to speed changes which increases. This reflects the initial energy expenditure for gaining altitude and speed. A dramatic increase in specific energy rate excess occurs during the Climb 2 and Cruise phases for approximately 9000 to 12000 seconds, reaching values well over 200 m/s. This significant rise indicates that the aircraft is undergoing a high-energy maneuver, such as a rapid climb to accelerate to a higher cruise speed.

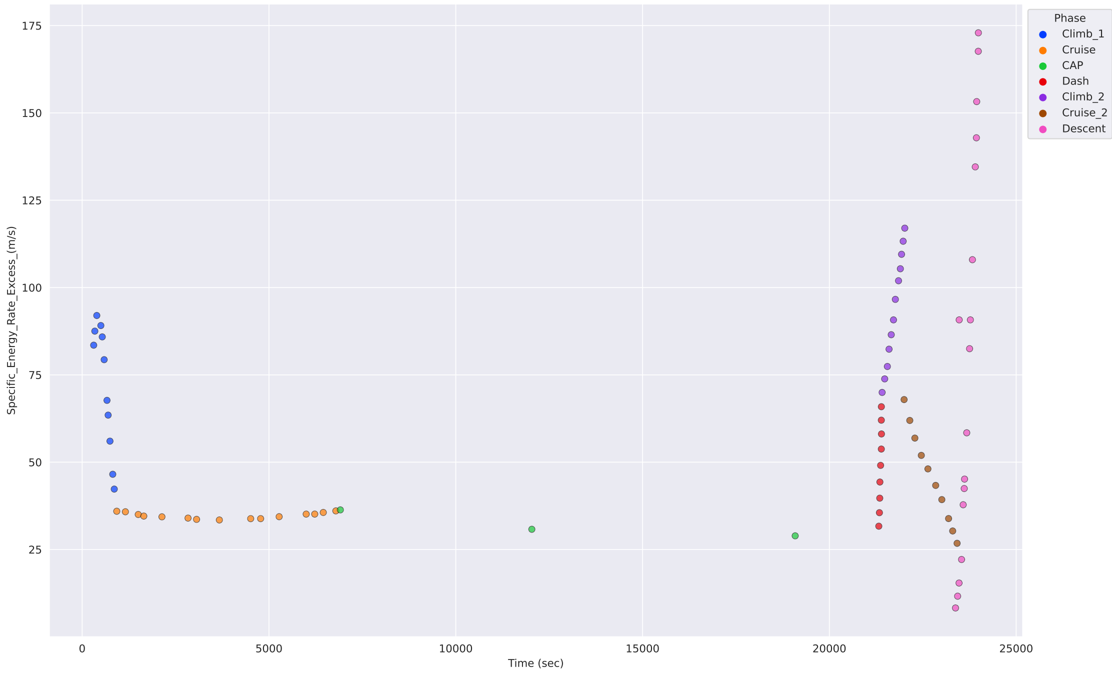



FIGURE 4.4: DCA Patrol: Specific Energy Rate Excess over time

## 4.6 Discussion of Results

The iterative sizing process using the AVIARY framework revealed several important insights into the optimal design and performance of the Homeland Defense Interceptor (HDI) aircraft. These results that shown in Table 4.1 directly support the research goals of identifying key design parameters and evaluating mission effectiveness.

To begin with, the optimization settled on a Gross Takeoff Weight (GTOW) of 36,066 lbm (16,329 kg). This figure represents a well-balanced compromise large enough to carry the required payload and fuel for extended missions, yet light enough to maintain the agility and responsiveness expected of an interceptor. The chosen GTOW places the HDI in the mid-size unmanned aerial vehicle category, aligning with its intended role as a capable yet efficient asset for national defense.

The study also confirmed several important performance and geometric characteristics. Notably, a wing area of 486.61 $ft^2$ (45.17 $m^2$) was identified as ideal. This size offers the lift needed for both long-endurance patrols and rapid interception maneuvers. Combined with a thrust-to-weight ratio of approximately

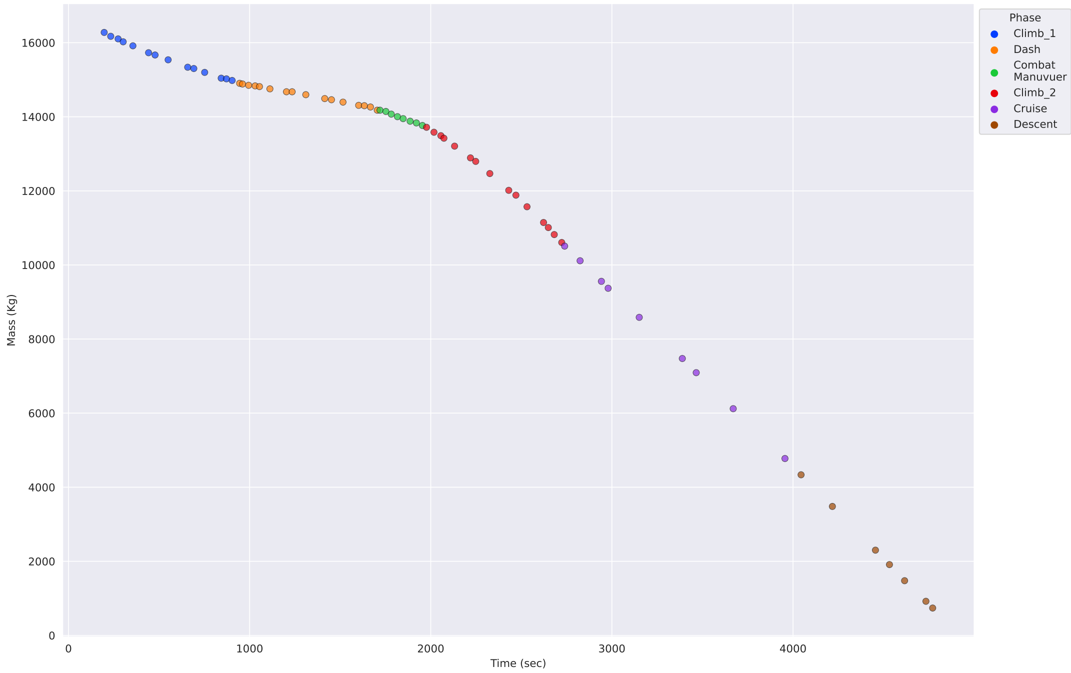



FIGURE 4.5: PDI: Mass over time

0.98, the aircraft demonstrates strong acceleration and climb performance key traits for quick-response missions.

The DCA Patrol mission profile emerged as the most influential factor in shaping the aircrafts overall design. The long endurance and loiter time required for this mission heavily influenced fuel capacity and aerodynamic efficiency.

With an optimized gross takeoff weight (GTOW) of around 36,066 lbm (16,359 Kg), the HDI falls into a distinct category. It is significantly lighter than larger fifth-generation stealth fighters like the F-22 Raptor, which has an empty weight of approximately 64,460 lbm (29,238.6 Kg) and an even higher GTOW. Compared to multirole fighters such as the MiG-29, with an empty weight of 24,251 lbs (11,000 Kg), the HDI also stands apart designed not for multirole versatility, but for focused interceptor performance. Its overall mass aligns more closely with aircraft like the Gripen JAS 39C, which has an empty weight near 15,000 lbs. However, the HDI's higher GTOW reflects the advantages of an unmanned design where it doesn't have a cockpit, pilot safety systems, and other human-related mass and its emphasis on extended endurance and payload capacity.

In relation to unmanned counterparts like the Bayraktar Kzlelma, which has


FIGURE 4.6: PDI: Drag over time plot

a GTOW around 13,000 lbm (5,987.4 Kg), the HDI is nearly double in size. This suggests a platform designed for more demanding missions: longer range, larger payloads, and faster response times. These characteristics support the HDIs intended role in defending wide territorial areas, rather than engaging in limited tactical strikes.

The HDIs thrust-to-weight ratio of 0.98 sits just below typical values for modern manned fighters, which often range from 1.0 to 1.2. While slightly lower, it remains within a range that ensures fast acceleration and effective climb rates both essential for interception tasks. The wing loading of 74.21 lb/ft$^2$ places the HDI in a moderate range lower than the MiG-29 (83 lb/ft$^2$) and F-22 (77.2 lb/ft$^2$), yet higher than the Gripen (58 lb/ft$^2$). Its wing area of 486 ft$^2$ provides a practical compromise between efficient cruise capability (needed for patrol endurance) and the agility required for responsive interception, distinguishing it from fighter designs that prioritize close-in, high-G maneuverability.

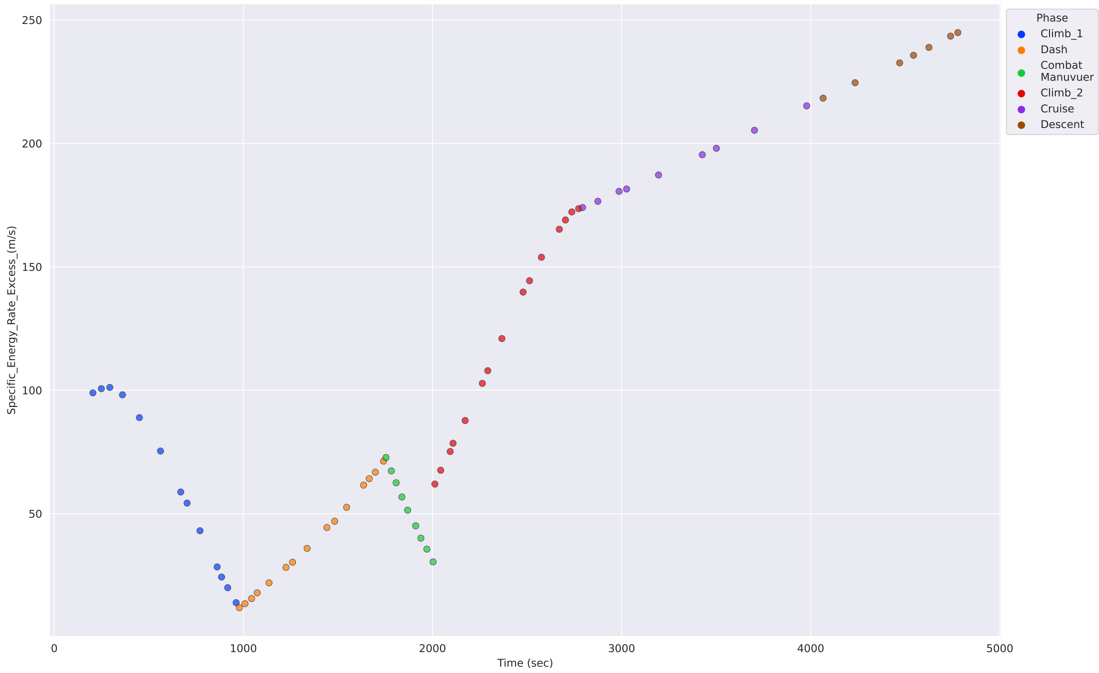



FIGURE 4.7: PDI: Specific Energy Rate Excess over time

### 4.6.1 Design Limitations and Challenges

Discussion on design constraints, such as:

- A key trade-off was between geometry and supersonic performance; the Mach 0.9 limit was chosen because the limitation on building the engine deck in AVIARY.
- Potential deviations from ideal parameters due to iterative assumptions.
- Balancing fuel requirements for high-speed dash and extended loiter was a significant challenge, influencing fuel capacity and wing sizing.

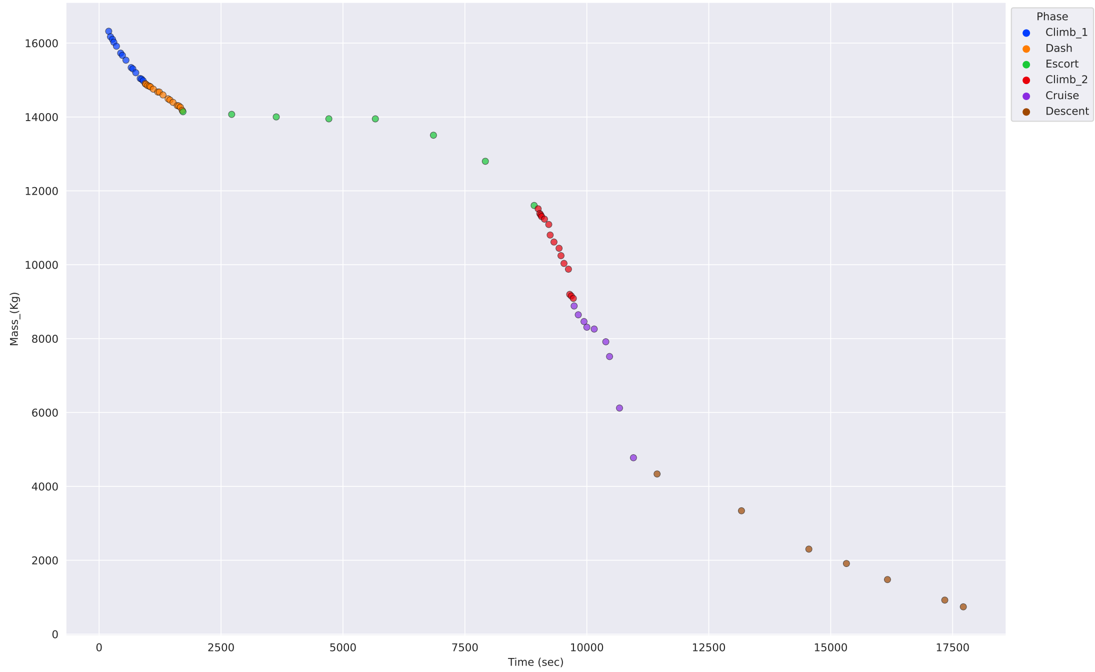



FIGURE 4.8: Escort: Mass over time

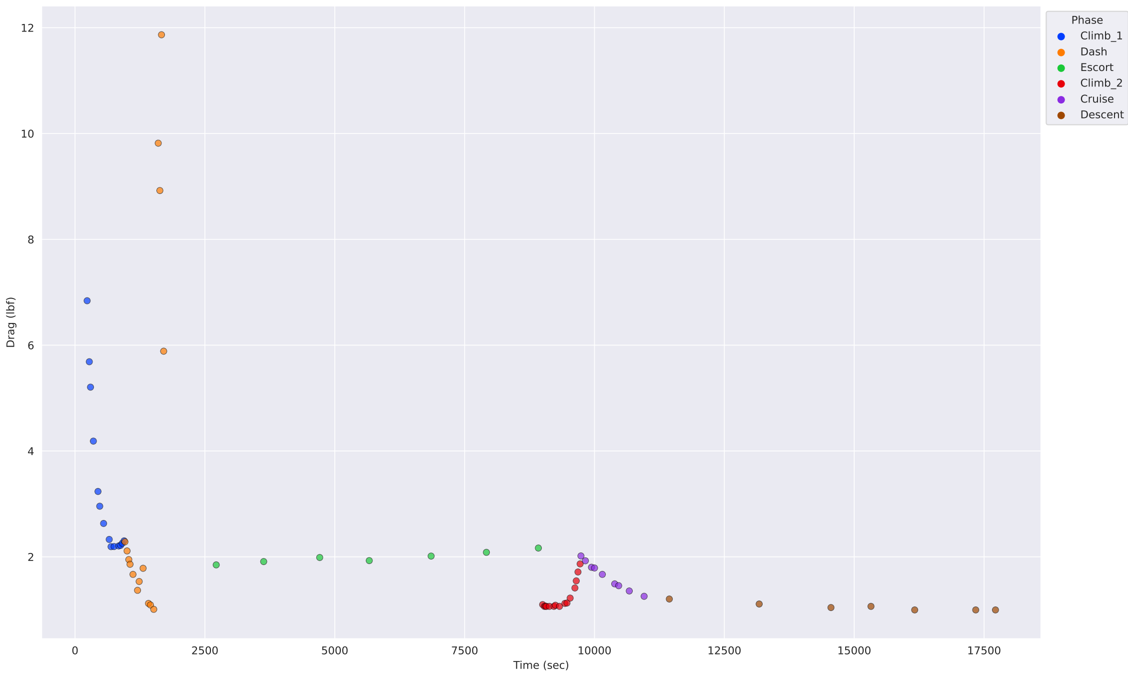



FIGURE 4.9: Escort: Drag over time

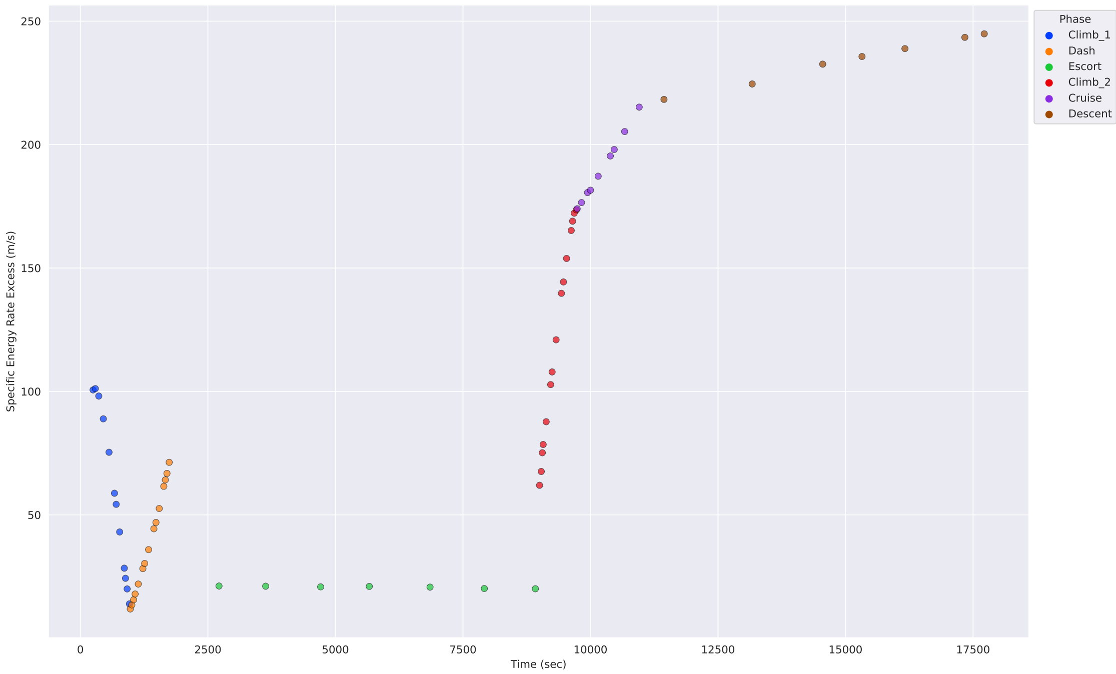



FIGURE 4.10: Escort: Energy Rate over time

# CHAPTER 5
# SUMMARY, CONCLUSION, RECOMMENDATION

## 5.1 Summary

This research set out to accomplish the preliminary sizing of an Homeland Defense Interceptor (HDI) aircraft, leveraging the powerful, open-source AVIARY framework. The study was driven by the growing demand for a new generation of interceptors aircraft capable of effectively countering advanced aerial threats.

To achieve this, AVIARY's integrated parametric environment, built upon OpenMDAO, was employed to conduct a detailed mission-specific sizing analysis. This analysis spanned three critical mission profiles outlined in the 2025 AIAA Undergraduate Design Team Request for Proposal (RFP): Defensive Counter-Air (DCA) Patrol, Point Defense Interception, and Intercept/Escort. It is important to note that due to the current capabilities of AVIARY, this design focused exclusively on subsonic flight regimes, thus excluding any supersonic aerodynamic effects from the analysis. From the comprehensive RFP requirements, the design was primarily driven by the intercept radius and the Defensive Counter-Air (DCA) Patrol mission's endurance capability. Initial design estimates were calibrated with robust benchmarking data drawn from well-established fighter aircraft, including the F-22 Raptor, Saab JAS 39C Gripen, MiG-29 Fulcrum, and Bayraktar Kzlelma. Through an iterative process involving simulation and constraint analysis within AVIARY, key parameters such as takeoff gross weight, fuel weight, wing area, thrust-to-weight ratio, and mission fuel fractions were meticulously determined. Notably, Raymers weight fraction estimation method, integrated within OpenMDAO, proved instrumental during the initial sizing phase.

A striking consistency in the HDI design's performance across all simulated mission profiles became evident, particularly in its impressive ability to fulfill both loiter and dash requirements during the demanding DCA mission. Among the specific discoveries:

- An optimal Gross Take-Off Weight (GTOW) of 36,066 lbm (16,359 Kg) was pinpointed. This weight proved to be a sweet spot, effectively balancing the aircraft's payload capacity, range, and maneuverability.
- Crucial geometric parameters, including a wing area of 486 $ft^2$ (45.17 $m^2$) and an Aspect Ratio of 3.98, designed to ensure ample lift for both patrol and interception duties.
- A total Thrust of 35,000 lbf (155.69 kN), resulting in a Thrust-to-Weight ratio of 0.98, ensuring potent acceleration and climb performance.
- A Wing Loading of Wing Loading of 72.04 lb/$ft^2$ (351.73 kg/$m^2$), supporting efficient flight across the mission profiles.
- HDI Design gross weight, thrust-to-weight ratio, wing loading, and range place it within the operational performance envelope of lightweight multirole fighters such as the Gripen JAS 39C and MiG-29 Fulcrum, while remaining aerodynamically efficient with a relatively high aspect ratio. The HDIs positioning in the Thrust-to-Weight vs Wing Loading plot validates its agility and responsiveness, which are essential for quick-intercept and defensive patrol missions.
- Delving into mission performance, the highest rate of mass loss was observed during the climb phase, a direct result of the high engine thrust demanded. Conversely, cruise segments demonstrated remarkably stable mass with minimal fuel consumption. The climb phase also showcased a vigorous energy gain rate, underlining the aircraft's potent thrust-to-weight ratio, while drag remained consistently low and steady during cruise, indicative of highly efficient aerodynamics.

This study marks a significant contribution to the field of preliminary aircraft design. It vividly demonstrates the practical application of AVIARY as a cutting-edge, open-source, and truly multidisciplinary tool for conceptual sizing. Furthermore, it offers a replicable and scalable methodology for sizing aircraft under demanding mission constraints, enabling designers to rapidly evaluate crucial trade offs concerning weight, range, thrust, and mission endurance.

## 5.2 Conclusion

In conclusion, this research successfully delivered a preliminary sizing model for a Homeland Defense Interceptor aircraft, thoughtfully developed using the AVIARY framework. This work effectively addresses the ambitious objectives set forth in the AIAA 2025 RFP. The study firmly establishes the feasibility of designing an unmanned interceptor fully capable of executing a diverse array of homeland defense missions, encompassing Defensive Counter-Air Patrol, Point Defense Interception, and Intercept/Escort as the result shown in Chapter 4 in Table 4.1. The iterative design process facilitated by AVIARY allowed for the precise determination of critical design parameters that align with the required performance boundaries. The consistent performance of the HDI design across various mission profiles, particularly its ability to meet demanding loiter and dash requirements, stands as strong validation for AVIARY's effectiveness as a rapid, multidisciplinary tool for conceptual aircraft design. It is important to reiterate that this design and analysis were conducted within the subsonic flight regime, primarily driven by the intercept radius and the DCA Patrol mission's endurance requirements, acknowledging the current limitations of the AVIARY tool in modeling supersonic aerodynamics. Ultimately, this effort provides a robust and practical methodological framework for future conceptual design endeavors, especially within the exciting domain of unmanned combat aircraft.

## 5.3 Recommendation

Building upon the insights gained from this preliminary sizing study, several avenues for future research and development emerge that could significantly enhance the HDI aircraft concept:

- Supersonic Flight Regime Integration:
  A critical next step involves extending the design and analysis to include supersonic flight capabilities. This would necessitate exploring alternative or augmented multidisciplinary design optimization (MDO) frameworks capable of robustly modeling supersonic aerodynamics, propulsion integration for supersonic performance, and associated structural considerations.

- Comprehensive RFP Compliance:
  While this study focused on key driving requirements, future work should aim to incorporate and satisfy all remaining RFP specifications, ensuring a more complete and compliant design solution. This includes a detailed analysis of all performance parameters, payload requirements, and operational constraints.
- Advanced Optimization and Higher-Fidelity Analysis:
  The preliminary sizing could be further refined through more advanced MDO techniques. Incorporating higher-fidelity analysis tools, such as Computational Fluid Dynamics (CFD) for detailed aerodynamic analysis, Finite Element Analysis (FEA) for structural integrity, and more sophisticated propulsion system modeling, would provide greater precision in the design parameters.
- Trade-off Studies and Sensitivity Analysis:
  Conducting extensive trade-off studies on various design variables (wing sweep, aspect ratio, engine cycle) and performing sensitivity analyses on key input parameters would provide a deeper understanding of design drivers and performance envelopes.
- Enhanced AVIARY Potential Analysis
  Enhance the AVIARY aircraft model to support detailed static and dynamic stability analysis, including the integration of control surface sizing, moments of inertia estimation, and trim analysis across multiple flight regimes. integrate aerodynamic databases (CFD or DATCOM) with AVIARYs framework to capture nonlinear aerodynamic effects more accurately, particularly for post-stall behavior or high-angle-of-attack scenarios.

These recommendations aim to guide subsequent design iterations toward a more refined, comprehensive, and ultimately deployable Homeland Defense Interceptor aircraft.

# Appendices

# APPENDIX A
# AIRCRAFT MODEL FILE

TABLE A.1: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 1 | aircraft:air-conditioning:mass-scaler | 0.8 | Unitless |
| 2 | aircraft:anti-icing:mass-scaler | 0.2 | Unitless |
| 3 | aircraft:apu:mass-scaler | 0.5 | Unitless |
| 4 | aircraft:avionics:mass-scaler | 1.2 | Unitless |
| 5 | aircraft:canard:area | 0 | $ft^2$ |
| 6 | aircraft:canard:aspect-ratio | 0 | Unitless |
| 7 | aircraft:canard:thickness-to-chord | 0 | Unitless |
| 8 | aircraft:crew-and-payload:baggage-mass-per-passenger | 0 | lbm |
| 9 | aircraft:crew-and-payload:cargo-container-mass-scaler | 0.5 | Unitless |
| 10 | aircraft:crew-and-payload:flight-crew-mass-scaler | 0 | Unitless |
| 11 | aircraft:crew-and-payload:mass-per-passenger | 300 | lbm |
| 12 | aircraft:crew-and-payload:misc-cargo | 0 | lbm |
| 13 | aircraft:crew-and-payload:non-flight-crew-mass-scaler | 0 | Unitless |
| 14 | aircraft:crew-and-payload:num-business-class | 0 | Unitless |
| 15 | aircraft:crew-and-payload:num-first-class | 0 | Unitless |
| 16 | aircraft:crew-and-payload:num-flight-attendants | 1 | Unitless |

TABLE A.2: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 17 | aircraft:crew-and-payload:num-flight-crew | 0 | Unitless |
| 18 | aircraft:crew-and-payload:num-galley-crew | 0 | Unitless |
| 19 | aircraft:crew-and-payload:num-non-flight-crew | 0 | Unitless |
| 20 | aircraft:crew-and-payload:num-passengers | 0 | Unitless |
| 21 | aircraft:crew-and-payload:num-tourist-class | 0 | Unitless |
| 22 | aircraft:crew-and-payload:passenger-service-mass-scaler | 0 | Unitless |
| 23 | aircraft:crew-and-payload:wing-cargo | 0 | lbm |
| 24 | aircraft:crew-and-payload:total-payload-mass | 802 | lbm |
| 25 | aircraft:design:base-area | 0 | $ft^2$ |
| 26 | aircraft:design:empty-mass | 21,407 | lbm |
| 27 | aircraft:design:empty-mass-margin-scaler | 1.1 | Unitless |
| 28 | aircraft:design:lift-dependent-drag-coeff-factor | 0.03 | Unitless |
| 29 | aircraft:design:touchdown-mass | 25,000 | lbm |
| 30 | aircraft:design:reserve-fuel-additional | 1,102.3 | lbm |
| 31 | aircraft:design:subsonic-drag-coeff-factor | 0.015 | Unitless |
| 32 | aircraft:design:supersonic-drag-coeff-factor | 1.2 | Unitless |

TABLE A.3: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 33 | aircraft:design:use-alt-mass | True | Unitless |
| 34 | aircraft:design:zero-lift-drag-coeff-factor | 0.012 | Unitless |
| 35 | aircraft:electrical:mass-scaler | 0.03 | Unitless |
| 36 | aircraft:engine:additional-mass-fraction | 0.05 | Unitless |
| 37 | aircraft:engine:constant-fuel-consumption | 0.9 | lbm/h |
| 38 | aircraft:engine:data-file | turbofan-28k.deck | Unitless |
| 39 | aircraft:engine:flight-idle-thrust-fraction | 0.1 | Unitless |
| 40 | aircraft:engine:flight-idle-max-fraction | 0.2 | Unitless |
| 41 | aircraft:engine:flight-idle-min-fraction | 0.05 | Unitless |
| 42 | aircraft:engine:fuel-flow-scaler-constant-term | 1 | Unitless |
| 43 | aircraft:engine:fuel-flow-scaler-linear-term | 1 | Unitless |
| 44 | aircraft:engine:generate-flight-idle | True | Unitless |
| 45 | aircraft:engine:geopotential-alt | False | Unitless |
| 46 | aircraft:engine:ignore-negative-thrust | True | Unitless |
| 47 | aircraft:engine:interpolation-method | slinear | Unitless |
| 48 | aircraft:engine:mass-scaler | 1 | Unitless |

TABLE A.4: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 49 | aircraft:engine:mass | 5,000 | lbm |
| 50 | aircraft:engine:num-engines | 2 | Unitless |
| 51 | aircraft:engine:num-fuselage-engines | 2 | Unitless |
| 52 | aircraft:engine:num-wing-engines | 0 | Unitless |
| 53 | aircraft:engine:reference-mass | 5,000 | lbm |
| 54 | aircraft:engine:reference-sls-thrust | 18,000 | lbf |
| 55 | aircraft:engine:scale-mass | True | Unitless |
| 56 | aircraft:engine:scale-performance | True | Unitless |
| 57 | aircraft:engine:scaled-sls-thrust | 18,000 | lbf |
| 58 | aircraft:engine:subsonic-fuel-flow-scaler | 0.9 | Unitless |
| 59 | aircraft:engine:supersonic-fuel-flow-scaler | 1.2 | Unitless |
| 60 | aircraft:engine:thrust-reversers-mass-scaler | 0 | Unitless |
| 61 | aircraft:engine:wing-locations | [0.0] | Unitless |
| 62 | aircraft:fins:area | 58.6 | $ft^2$ |
| 63 | aircraft:fins:mass-scaler | 1.2 | Unitless |
| 64 | aircraft:fins:mass | 661 | lbm |

TABLE A.5: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 65 | aircraft:fins:num-fins | 2 | Unitless |
| 66 | aircraft:fins:taper-ratio | 0.5 | Unitless |
| 67 | aircraft:fuel:auxiliary-fuel-capacity | 0 | lbm |
| 68 | aircraft:fuel:density-ratio | 0.8 | Unitless |
| 69 | aircraft:fuel:fuel-system–mass-scaler | 1 | Unitless |
| 70 | aircraft:fuel:fuselage-fuel-capacity | 3,000 | lbm |
| 71 | aircraft:fuel:num-tanks | 5 | Unitless |
| 72 | aircraft:fuel:total-capacity | 15,000 | lbm |
| 73 | aircraft:fuel:unusable-fuel-mass-scaler | 0.02 | Unitless |
| 74 | aircraft:furnishings:mass-scaler | 0 | Unitless |
| 75 | aircraft:fuselage:length | 50 | ft |
| 76 | aircraft:fuselage:mass-scaler | 1 | Unitless |
| 77 | aircraft:fuselage:max-height | 8.2 | ft |
| 78 | aircraft:fuselage:max-width | 5.2 | ft |
| 79 | aircraft:fuselage:military-cargo-floor | False | Unitless |
| 80 | aircraft:fuselage:num-fuselages | 1 | Unitless |

TABLE A.6: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 81 | aircraft:fuselage:passenger-compartment-length | 0 | ft |
| 82 | aircraft:fuselage:planform-area | 200 | $ft^2$ |
| 83 | aircraft:fuselage:wetted-area-scaler | 1 | Unitless |
| 84 | aircraft:fuselage:wetted-area | 753 | $ft^2$ |
| 85 | aircraft:horizontal-tail:area | 154.73 | $ft^2$ |
| 86 | aircraft:horizontal-tail:aspect-ratio | 4 | Unitless |
| 87 | aircraft:horizontal-tail:mass-scaler | 1 | Unitless |
| 88 | aircraft:horizontal-tail:taper-ratio | 0.4 | lbm |
| 89 | aircraft:horizontal-tail:thickness-to-chord | 0.1 | Unitless |
| 90 | aircraft:horizontal-tail:vertical-tail-fraction | 0.3 | Unitless |
| 91 | aircraft:horizontal-tail:wetted-area-scaler | 1 | ft |
| 92 | aircraft:horizontal-tail:wetted-area | 172 | $ft^2$ |
| 93 | aircraft:hydraulics:mass-scaler | 1 | ft |
| 94 | aircraft:hydraulics:system-pressure | 720,000 | $lbf/ft^2$ |
| 95 | aircraft:instruments:mass-scaler | 1 | Unitless |
| 96 | aircraft:landing-gear:carrier-based | False | Unitless |

TABLE A.7: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 97 | aircraft:landing-gear:main-gear-mass-scaler | 1 | Unitless |
| 98 | aircraft:landing-gear:main-gear-oleo-length | 23.6 | inch |
| 99 | aircraft:landing-gear:nose-gear-mass-scaler | 1 | Unitless |
| 100 | aircraft:landing-gear:nose-gear-oleo-length | 15.75 | inch |
| 101 | aircraft:nacelle:avg-diameter | 3,3 | ft |
| 102 | aircraft:nacelle:avg-length | 16.5 | ft |
| 103 | aircraft:nacelle:count-factor | 1 | Unitless |
| 104 | aircraft:nacelle:mass-scaler | 1 | Unitless |
| 105 | aircraft:nacelle:wetted-area-scaler | 1 | Unitless |
| 106 | aircraft:paint:mass-per-unit-area | 14.23 | $/lbm/ft^2$ |
| 107 | aircraft:propulsion:engine-oil-mass-scaler | 1 | Unitless |
| 108 | aircraft:propulsion:misc-mass-scaler | 1 | Unitless |
| 109 | aircraft:vertical-tail:area | 53.98 | $ft^2$ |
| 110 | aircraft:vertical-tail:aspect-ratio | 1.5 | Unitless |
| 111 | aircraft:vertical-tail:mass-scaler | 1 | Unitless |
| 112 | aircraft:vertical-tail:num-tails | 2 | Unitless |

TABLE A.8: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 113 | aircraft:vertical-tail:taper-ratio | 0.5 | Unitless |
| 114 | aircraft:vertical-tail:thickness-to-chord | 0.1 | Unitless |
| 115 | aircraft:vertical-tail:wetted-area-scaler | 1.1 | Unitless |
| 116 | aircraft:vertical-tail:wetted-area | 97.6 | $ft^2$ |
| 117 | aircraft:wing:aeroelastic-tailoring-factor | 1 | Unitless |
| 118 | aircraft:wing:airfoil-technology | 2 | Unitless |
| 119 | aircraft:wing:area | 445.03 | $ft^2$ |
| 120 | aircraft:wing:aspect-ratio | 3.98 | Unitless |
| 121 | aircraft:wing:bending-factor | 1 | Unitless |
| 122 | aircraft:wing:bending-mass-scaler | 1 | Unitless |
| 123 | aircraft:wing:chord-per-semispan | 4 * 0.23 * 0.084 | Unitless |
| 124 | aircraft:wing:composite-fraction | 0.5 | Unitless |
| 125 | aircraft:wing:control-surface-area | 107.6 | $ft^2$ |
| 126 | aircraft:wing:control-surface-area-ratio | 0.2 | Unitless |
| 127 | aircraft:wing:glove-and-bat | 0 | $ft^2$ |
| 128 | aircraft:wing:input-station-dist | 0 * 0.2759 * 0.9367 | Unitless |

TABLE A.9: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 129 | aircraft:wing:load-distribution-control | 0.8 | Unitless |
| 130 | aircraft:wing:load-fraction | 0.5 | Unitless |
| 131 | aircraft:wing:load-path-sweep-dist | 1.5 * 22 | deg |
| 132 | aircraft:wing:mass-scaler | 1 | Unitless |
| 133 | aircraft:wing:max-camber-at-70-semispan | 0.02 | Unitless |
| 134 | aircraft:wing:misc-mass-scaler | 1 | Unitless |
| 135 | aircraft:wing:num-integration-stations | 5 | Unitless |
| 136 | aircraft:wing:shear-control-mass-scaler | 1 | Unitless |
| 137 | aircraft:wing:span-efficiency-reduction | True | Unitless |
| 138 | aircraft:wing:span | 44 | ft |
| 139 | aircraft:wing:strut-bracing-factor | 0 | Unitless |
| 140 | aircraft:wing:surface-ctrl-mass-scaler | 1 | Unitless |
| 141 | aircraft:wing:sweep | 50 | deg |
| 142 | aircraft:wing:taper-ratio | 0.4 | Unitless |
| 143 | aircraft:wing:thickness-to-chord-dist | 0.1 * 0.115 * 0.104 | Unitless |
| 144 | aircraft:wing:thickness-to-chord | 0.08 | Unitless |

TABLE A.10: Interceptor Aircraft Model Input CSV File

| No. | Parameter | Value | Unit |
|---|---|---|---|
| 145 | aircraft:wing:ultimate-load-factor | 9 | Unitless |
| 146 | aircraft:wing:var-sweep-mass-penalty | 0 | Unitless |
| 147 | aircraft:wing:wetted-area-scaler | 1 | Unitless |
| 148 | aircraft:wing:wetted-area | 960 | $ft^2$ |
| 149 | mission:constraints:max-mach | 1.6 | Unitless |
| 150 | mission:design:cruise-altitude | 35,000 | ft |
| 151 | mission:design:gross-mass | 36,773 | lbm |
| 152 | mission:design:range | 800 | Nm |
| 153 | mission:cruise:range | 500 | Nm |
| 154 | mission:design:thrust-takeoff-per-eng | 36,000 | lbf |
| 155 | mission:landing:lift-coefficient-max | 2.2 | Unitless |
| 156 | mission:summary:cruise-mach | 0.9 | Unitless |
| 157 | mission:summary:fuel-flow-scaler | 1 | Unitless |
| 158 | mission:takeoff:fuel-simple | 1,102 | lbm |
| 159 | mission:takeoff:lift-coefficient-max | 2 | Unitless |
| 160 | mission:takeoff:lift-over-drag | 10 | Unitless |
| 161 | settings:equations-of-motion | Height-Energy | Unitless |
| 162 | settings:mass-method | FLOPS | Unitless |

# APPENDIX B
# PYTHON CODES

## B.1 Level 2 Example

```python
from copy import deepcopy

import aviary.api as av

# inputs that run_aviary() requires
aircraft_data = 'models/test_aircraft/aircraft_for_bench_GwGm.csv'
optimizer = 'IPOPT'
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'aviary_history.db'
restart_filename = None
max_iter = 0
phase_info = deepcopy(av.default_2DOF_phase_info)

# Build problem
prob = av.AviaryProblem(analysis_scheme)

# Load aircraft and options data from user
# Allow for user overrides here
prob.load_inputs(aircraft_data, phase_info)

# Preprocess inputs
prob.check_and_preprocess_inputs()

# adds a pre-mission group (propulsion, geometry, aerodynamics, and mass)
prob.add_pre_mission_systems()

```

```python
# adds a sequence of core mission phases.
prob.add_phases()

# adds a landing phase
prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

# adds an optimizer to the driver
prob.add_driver(optimizer, max_iter=max_iter)

# adds relevant design variables
prob.add_design_variables()

# Load optimization problem formulation
# Detail which variables the optimizer can control
prob.add_objective(objective_type=objective_type)

# setup the problem
prob.setup()

# set initial guesses of states and controls variables
prob.set_initial_guesses()

# run the problem we just set up
prob.run_aviary_problem(record_filename, restart_filename=
    restart_filename)
```

## B.2 Default Phase Info: FlOPS

```python
from aviary.subsystems.propulsion.propulsion_builder import
    CorePropulsionBuilder
from aviary.subsystems.geometry.geometry_builder import
    CoreGeometryBuilder
from aviary.subsystems.mass.mass_builder import CoreMassBuilder
from aviary.subsystems.aerodynamics.aerodynamics_builder import
    CoreAerodynamicsBuilder
from aviary.variable_info.variable_meta_data import _MetaData as
    BaseMetaData
```

```
from aviary.variable_info.variables import Dynamic, Mission
from aviary.variable_info.enums import LegacyCode

FLOPS = LegacyCode.FLOPS

prop = CorePropulsionBuilder('core_propulsion', BaseMetaData)
mass = CoreMassBuilder('core_mass', BaseMetaData, FLOPS)
aero = CoreAerodynamicsBuilder('core_aerodynamics', BaseMetaData, FLOPS)
geom = CoreGeometryBuilder('core_geometry', BaseMetaData, FLOPS)

default_premission_subsystems = [prop, geom, mass, aero]
default_mission_subsystems = [aero, prop]


phase_info = {
    "pre_mission": {"include_takeoff": False, "optimize_mass": True},
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.2, "unitless"),
            "final_mach": (0.72, "unitless"),
            "mach_bounds": ((0.18, 0.74), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (32000.0, "ft"),
            "altitude_bounds": ((0.0, 34000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((0.0, 0.0), "min"),
            "duration_bounds": ((64.0, 192.0), "min"),
            "add_initial_mass_constraint": False,
```

```
        },
    },
    "cruise": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.72, "unitless"),
            "final_mach": (0.72, "unitless"),
            "mach_bounds": ((0.7, 0.74), "unitless"),
            "initial_altitude": (32000.0, "ft"),
            "final_altitude": (34000.0, "ft"),
            "altitude_bounds": ((23000.0, 38000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((64.0, 192.0), "min"),
            "duration_bounds": ((56.5, 169.5), "min"),
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.72, "unitless"),
            "final_mach": (0.36, "unitless"),
            "mach_bounds": ((0.34, 0.74), "unitless"),
            "initial_altitude": (34000.0, "ft"),
            "final_altitude": (500.0, "ft"),
            "altitude_bounds": ((0.0, 38000.0), "ft"),
```

```
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((120.5, 361.5), "min"),
            "duration_bounds": ((29.0, 87.0), "min"),
        },
    },
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (1906., "nmi"),
    },
}


def phase_info_parameterization(phase_info, post_mission_info, aviary_inputs):
    """
    Modify the values in the phase_info dictionary to accommodate different values
    for the following mission design inputs: cruise altitude, cruise Mach number,
    cruise range, design gross mass.

    Parameters
    ----------
    phase_info : dict
        Dictionary of phase settings for a mission profile
    aviary_inputs : <AviaryValues>
        Object containing values and units for all aviary inputs and options

    Returns
    -------
    dict
        Modified phase_info that has been changed to match the new mission
        parameters
    """
```

```

    alt_cruise = aviary_inputs.get_val(Mission.Design.CRUISE_ALTITUDE, units='ft')
    mach_cruise = aviary_inputs.get_val(Mission.Summary.CRUISE_MACH)

    # Range
    old_range_cruise, range_units = post_mission_info['target_range']
    range_cruise = aviary_inputs.get_val(Mission.Design.RANGE, units=range_units)
    if range_cruise != old_range_cruise:
        new_val = post_mission_info['target_range'][0] * range_cruise / old_range_cruise
        post_mission_info['target_range'] = (new_val, range_units)

    # Altitude
    old_alt_cruise = 32000.
    if alt_cruise != old_alt_cruise:
        phase_info['climb']['user_options']['final_altitude'] = (alt_cruise, 'ft')
        phase_info['cruise']['user_options']['initial_altitude'] = (alt_cruise, 'ft')
        phase_info['cruise']['user_options']['final_altitude'] = (alt_cruise, 'ft')
        phase_info['descent']['user_options']['initial_altitude'] = (alt_cruise, 'ft')

    # Mach
    old_mach_cruise = 0.72
    if mach_cruise != old_mach_cruise:
        phase_info['climb']['user_options']['final_mach'] = (mach_cruise, "unitless")
        phase_info['cruise']['user_options']['initial_mach'] = (mach_cruise, "unitless")
        phase_info['cruise']['user_options']['final_mach'] = (mach_cruise, "unitless")
        phase_info['descent']['user_options']['initial_mach'] = (mach_cruise, "unitless")

```

```
    return phase_info, post_mission_info
```

## B.3 Default Phase Info: GASP

```
from aviary.variable_info.enums import SpeedType
from aviary.variable_info.variables import Mission

# defaults for 2DOF based phases
mission_distance = 3675

phase_info = {
    'groundroll': {
        'user_options': {
            'num_segments': 1,
            'order': 3,
            'connect_initial_mass': False,
            'fix_initial': True,
            'fix_initial_mass': False,
            'duration_ref': (50.0, 's'),
            'duration_bounds': ((1.0, 100.0), 's'),
            'velocity_lower': (0, 'kn'),
            'velocity_upper': (1000, 'kn'),
            'velocity_ref': (150, 'kn'),
            'mass_lower': (0, 'lbm'),
            'mass_upper': (None, 'lbm'),
            'mass_ref': (150_000, 'lbm'),
            'mass_defect_ref': (150_000, 'lbm'),
            'distance_lower': (0, 'ft'),
            'distance_upper': (10.0e3, 'ft'),
            'distance_ref': (3000, 'ft'),
            'distance_defect_ref': (3000, 'ft'),
        },
        'initial_guesses': {
            'time': ([0.0, 40.0], 's'),
            'velocity': ([0.066, 143.1], 'kn'),
            'distance': ([0.0, 1000.0], 'ft'),
            'throttle': ([0.956, 0.956], 'unitless'),
        },
    },
    'rotation': {
```

```
        'user_options': {
            'num_segments': 1,
            'order': 3,
            'fix_initial': False,
            'duration_bounds': ((1, 100), 's'),
            'duration_ref': (50.0, 's'),
            'velocity_lower': (0, 'kn'),
            'velocity_upper': (1000, 'kn'),
            'velocity_ref': (100, 'kn'),
            'velocity_ref0': (0, 'kn'),
            'mass_lower': (0, 'lbm'),
            'mass_upper': (None, 'lbm'),
            'mass_ref': (150_000, 'lbm'),
            'mass_defect_ref': (150_000, 'lbm'),
            'distance_lower': (0, 'ft'),
            'distance_upper': (10_000, 'ft'),
            'distance_ref': (5000, 'ft'),
            'distance_defect_ref': (5000, 'ft'),
            'angle_lower': (0.0, 'rad'),
            'angle_upper': (5.0, 'rad'),
            'angle_ref': (5.0, 'rad'),
            'angle_defect_ref': (5.0, 'rad'),
            'normal_ref': (10000, 'lbf'),
        },
        'initial_guesses': {
            'time': ([40.0, 5.0], 's'),
            'angle_of_attack': ([0.0, 2.5], 'deg'),
            'velocity': ([143, 150.0], 'kn'),
            'distance': ([3680.37217765, 4000], 'ft'),
            'throttle': ([0.956, 0.956], 'unitless'),
        },
    },
    'ascent': {
        'user_options': {
            'num_segments': 4,
            'order': 3,
            'fix_initial': False,
            'velocity_lower': (0, 'kn'),
            'velocity_upper': (700, 'kn'),
            'velocity_ref': (200, 'kn'),
```

```
                'velocity_ref0': (0, 'kn'),
                'mass_lower': (0, 'lbm'),
                'mass_upper': (None, 'lbm'),
                'mass_ref': (150_000, 'lbm'),
                'mass_defect_ref': (150_000, 'lbm'),
                'distance_lower': (0, 'ft'),
                'distance_upper': (15_000, 'ft'),
                'distance_ref': (1e4, 'ft'),
                'distance_defect_ref': (1e4, 'ft'),
                'alt_lower': (0, 'ft'),
                'alt_upper': (700, 'ft'),
                'alt_ref': (1000, 'ft'),
                'alt_defect_ref': (1000, 'ft'),
                'final_altitude': (500, 'ft'),
                'alt_constraint_ref': (500, 'ft'),
                'angle_lower': (-10, 'rad'),
                'angle_upper': (20, 'rad'),
                'angle_ref': (57.2958, 'deg'),
                'angle_defect_ref': (57.2958, 'deg'),
                'pitch_constraint_lower': (0.0, 'deg'),
                'pitch_constraint_upper': (15.0, 'deg'),
                'pitch_constraint_ref': (1.0, 'deg'),
            },
            'initial_guesses': {
                'time': ([45.0, 25.0], 's'),
                'flight_path_angle': ([0.0, 8.0], 'deg'),
                'angle_of_attack': ([2.5, 1.5], 'deg'),
                'velocity': ([150.0, 185.0], 'kn'),
                'distance': ([4.0e3, 10.0e3], 'ft'),
                'altitude': ([0.0, 500.0], 'ft'),
                'tau_gear': (0.2, 'unitless'),
                'tau_flaps': (0.9, 'unitless'),
                'throttle': ([0.956, 0.956], 'unitless'),
            },
        },
        'accel': {
            'user_options': {
                'num_segments': 1,
                'order': 3,
                'fix_initial': False,
```

```
            'alt': (500, 'ft'),
            'EAS_constraint_eq': (250, 'kn'),
            'duration_bounds': ((1, 200), 's'),
            'duration_ref': (1000, 's'),
            'velocity_lower': (150, 'kn'),
            'velocity_upper': (270, 'kn'),
            'velocity_ref': (250, 'kn'),
            'velocity_ref0': (150, 'kn'),
            'mass_lower': (0, 'lbm'),
            'mass_upper': (None, 'lbm'),
            'mass_ref': (150_000, 'lbm'),
            'mass_defect_ref': (150_000, 'lbm'),
            'distance_lower': (0, 'NM'),
            'distance_upper': (150, 'NM'),
            'distance_ref': (5, 'NM'),
            'distance_defect_ref': (5, 'NM'),
        },
        'initial_guesses': {
            'time': ([70.0, 13.0], 's'),
            'velocity': ([185.0, 250.0], 'kn'),
            'distance': ([10.0e3, 20.0e3], 'ft'),
            'throttle': ([0.956, 0.956], 'unitless'),
        },
    },
    'climb1': {
        'user_options': {
            'num_segments': 1,
            'order': 3,
            'fix_initial': False,
            'EAS_target': (250, 'kn'),
            'mach_cruise': 0.8,
            'target_mach': False,
            'final_altitude': (10.0e3, 'ft'),
            'duration_bounds': ((30, 300), 's'),
            'duration_ref': (1000, 's'),
            'alt_lower': (400, 'ft'),
            'alt_upper': (11_000, 'ft'),
            'alt_ref': (10.0e3, 'ft'),
            'mass_lower': (0, 'lbm'),
            'mass_upper': (None, 'lbm'),
```

```
                'mass_ref': (150_000, 'lbm'),
                'mass_defect_ref': (150_000, 'lbm'),
                'distance_lower': (0, 'NM'),
                'distance_upper': (500, 'NM'),
                'distance_ref': (10, 'NM'),
                'distance_ref0': (0, 'NM'),
            },
            'initial_guesses': {
                'time': ([1.0, 2.0], 'min'),
                'distance': ([20.0e3, 100.0e3], 'ft'),
                'altitude': ([500.0, 10.0e3], 'ft'),
                'throttle': ([0.956, 0.956], 'unitless'),
            },
        },
        'climb2': {
            'user_options': {
                'num_segments': 3,
                'order': 3,
                'fix_initial': False,
                'EAS_target': (270, 'kn'),
                'mach_cruise': 0.8,
                'target_mach': True,
                'final_altitude': (37.5e3, 'ft'),
                'required_available_climb_rate': (0.1, 'ft/min'),
                'duration_bounds': ((200, 17_000), 's'),
                'duration_ref': (5000, 's'),
                'alt_lower': (9000, 'ft'),
                'alt_upper': (40000, 'ft'),
                'alt_ref': (30000, 'ft'),
                'alt_ref0': (0, 'ft'),
                'mass_lower': (0, 'lbm'),
                'mass_upper': (None, 'lbm'),
                'mass_ref': (150_000, 'lbm'),
                'mass_defect_ref': (150_000, 'lbm'),
                'distance_lower': (10, 'NM'),
                'distance_upper': (1000, 'NM'),
                'distance_ref': (500, 'NM'),
                'distance_ref0': (0, 'NM'),
                'distance_defect_ref': (500, 'NM'),
            },
```

```
            'initial_guesses': {
                'time': ([216.0, 1300.0], 's'),
                'distance': ([100.0e3, 200.0e3], 'ft'),
                'altitude': ([10.0e3, 37.5e3], 'ft'),
                'throttle': ([0.956, 0.956], 'unitless'),
            },
        },
        'cruise': {
            'user_options': {
                'alt_cruise': (37.5e3, 'ft'),
                'mach_cruise': 0.8,
            },
            'initial_guesses': {
                # [Initial mass, delta mass] for special cruise phase.
                'mass': ([171481.0, -35000], 'lbm'),
                'initial_distance': (200.0e3, 'ft'),
                'initial_time': (1516.0, 's'),
                'altitude': (37.5e3, 'ft'),
                'mach': (0.8, 'unitless'),
            },
        },
        'desc1': {
            'user_options': {
                'num_segments': 3,
                'order': 3,
                'fix_initial': False,
                'input_initial': False,
                'EAS_limit': (350, 'kn'),
                'mach_cruise': 0.8,
                'input_speed_type': SpeedType.MACH,
                'final_altitude': (10.0e3, 'ft'),
                'duration_bounds': ((300.0, 900.0), 's'),
                'duration_ref': (1000, 's'),
                'alt_lower': (1000, 'ft'),
                'alt_upper': (40_000, 'ft'),
                'alt_ref': (30_000, 'ft'),
                'alt_ref0': (0, 'ft'),
                'alt_constraint_ref': (10000, 'ft'),
                'mass_lower': (0, 'lbm'),
                'mass_upper': (None, 'lbm'),
```

```
            'mass_ref': (140_000, 'lbm'),
            'mass_ref0': (0, 'lbm'),
            'mass_defect_ref': (140_000, 'lbm'),
            'distance_lower': (3000.0, 'NM'),
            'distance_upper': (5000.0, 'NM'),
            'distance_ref': (mission_distance, 'NM'),
            'distance_ref0': (0, 'NM'),
            'distance_defect_ref': (100, 'NM'),
        },
        'initial_guesses': {
            'mass': (136000.0, 'lbm'),
            'altitude': ([37.5e3, 10.0e3], 'ft'),
            'throttle': ([0.0, 0.0], 'unitless'),
            'distance': ([0.92 * mission_distance, 0.96 *
mission_distance], 'NM'),
            'time': ([28000.0, 500.0], 's'),
        },
    },
    'desc2': {
        'user_options': {
            'num_segments': 1,
            'order': 7,
            'fix_initial': False,
            'input_initial': False,
            'EAS_limit': (250, 'kn'),
            'mach_cruise': 0.80,
            'input_speed_type': SpeedType.EAS,
            'final_altitude': (1000, 'ft'),
            'duration_bounds': ((100.0, 5000), 's'),
            'duration_ref': (500, 's'),
            'alt_lower': (500, 'ft'),
            'alt_upper': (11_000, 'ft'),
            'alt_ref': (10.0e3, 'ft'),
            'alt_ref0': (1000, 'ft'),
            'alt_constraint_ref': (1000, 'ft'),
            'mass_lower': (0, 'lbm'),
            'mass_upper': (None, 'lbm'),
            'mass_ref': (150_000, 'lbm'),
            'mass_defect_ref': (150_000, 'lbm'),
            'distance_lower': (0, 'NM'),
```

```
                'distance_upper': (5000, 'NM'),
                'distance_ref': (3500, 'NM'),
                'distance_defect_ref': (100, 'NM'),
            },
            'initial_guesses': {
                'mass': (136000.0, 'lbm'),
                'altitude': ([10.0e3, 1.0e3], 'ft'),
                'throttle': ([0.0, 0.0], 'unitless'),
                'distance': ([0.96 * mission_distance, mission_distance], 'NM'),
                'time': ([28500.0, 500.0], 's'),
            },
        },
}


def phase_info_parameterization(phase_info, post_mission_info, aviary_inputs):
    """
    Modify the values in the phase_info dictionary to accomodate different values
    for the following mission design inputs: cruise altitude, cruise Mach number,
    cruise range, design gross mass.

    Parameters
    ----------
    phase_info : dict
        Dictionary of phase settings for a mission profile
    post_mission_info : dict
        Dictionary of phase settings for a post mission profile
    aviary_inputs : <AviaryValues>
        Object containing values and units for all aviary inputs and options

    Returns
    -------
    dict
        Modified phase_info that has been changed to match the new mission
```

```python
        parameters
    """
    range_cruise = aviary_inputs.get_val(Mission.Design.RANGE, units='NM')
    alt_cruise = aviary_inputs.get_val(Mission.Design.CRUISE_ALTITUDE, units='ft')
    gross_mass = aviary_inputs.get_val(Mission.Design.GROSS_MASS, units='lbm')
    mach_cruise = aviary_inputs.get_val(Mission.Design.MACH)

    # Range
    old_range_cruise = phase_info['desc2']['initial_guesses']['distance'][0][1]
    range_scale = 1.0
    if range_cruise != old_range_cruise:
        phase_info['desc1']['initial_guesses']['distance'] = (
            [0.92 * range_cruise, 0.96 * range_cruise],
            'NM',
        )
        phase_info['desc2']['initial_guesses']['distance'] = (
            [0.96 * range_cruise, range_cruise],
            'NM',
        )
        range_scale = range_cruise / old_range_cruise

    # Altitude
    old_alt_cruise = phase_info['climb2']['user_options']['final_altitude'][0]
    if alt_cruise != old_alt_cruise:
        phase_info['climb2']['user_options']['final_altitude'] = (alt_cruise, 'ft')
        phase_info['climb2']['initial_guesses']['altitude'] = ([10.0e3, alt_cruise], 'ft')
        phase_info['cruise']['initial_guesses']['altitude'] = (alt_cruise, 'ft')
        phase_info['desc1']['initial_guesses']['altitude'] = ([alt_cruise, 10.0e3], 'ft')

        # TODO - Could adjust time guesses/bounds in climb2 and desc2.
```

```

    # Mass
    old_gross_mass = 175400.0
    if gross_mass != old_gross_mass:
        # Note, this requires that the guess for gross mass is pretty close to the
        # compute mass.

        fuel_used = 35000 * range_scale
        phase_info['groundroll']['initial_guesses']['mass'] = ([gross_mass, gross_mass], 'lbm')
        phase_info['rotation']['initial_guesses']['mass'] = ([gross_mass, gross_mass], 'lbm')
        phase_info['accel']['initial_guesses']['mass'] = ([gross_mass, gross_mass], 'lbm')
        phase_info['ascent']['initial_guesses']['mass'] = ([gross_mass, gross_mass], 'lbm')

        phase_info['cruise']['initial_guesses']['mass'] = ([gross_mass, -fuel_used], 'lbm')

        end_mass = gross_mass - fuel_used
        phase_info['desc1']['initial_guesses']['mass'] = (end_mass, 'lbm')
        phase_info['desc2']['initial_guesses']['mass'] = (end_mass, 'lbm')

    # Mach
    old_mach_cruise = phase_info['cruise']['initial_guesses']['mach'][0]
    if mach_cruise != old_mach_cruise:
        phase_info['cruise']['initial_guesses']['mach'] = (mach_cruise, 'unitless')

    return phase_info, post_mission_info
```

## B.4 Outputted Phase Info

```
phase_info = {
```

```
    "pre_mission": {"include_takeoff": False, "optimize_mass": True},
    "climb_1": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "polynomial_control_order": 1,
            "use_polynomial_control": True,
            "num_segments": 9,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.2, "unitless"),
            "final_mach": (0.72, "unitless"),
            "mach_bounds": ((0.18000000000000002, 0.74), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (1500.0, "ft"),
            "altitude_bounds": ((0.0, 2000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((0.0, 0.0), "min"),
            "duration_bounds": ((10.5, 31.5), "min"),
        },
        "initial_guesses": {"times": ([0, 21], "min")},
    },
    "climb_2": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "polynomial_control_order": 1,
            "use_polynomial_control": True,
            "num_segments": 9,
            "order": 3,
            "solve_for_distance": False,
```

```
            "initial_mach": (0.72, "unitless"),
            "final_mach": (0.72, "unitless"),
            "mach_bounds": ((0.7, 0.74), "unitless"),
            "initial_altitude": (1500.0, "ft"),
            "final_altitude": (27000.0, "ft"),
            "altitude_bounds": ((1000.0, 27500.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((10.5, 31.5), "min"),
            "duration_bounds": ((15.0, 45.0), "min"),
        },
        "initial_guesses": {"times": ([21, 30], "min")},
    },
    "climb_3": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "polynomial_control_order": 1,
            "use_polynomial_control": True,
            "num_segments": 9,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.72, "unitless"),
            "final_mach": (0.72, "unitless"),
            "mach_bounds": ((0.7, 0.74), "unitless"),
            "initial_altitude": (27000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((26500.0, 35500.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((25.5, 76.5), "min"),
            "duration_bounds": ((6.0, 18.0), "min"),
        },
        "initial_guesses": {"times": ([51, 12], "min")},
```

```
    },
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (443, "nmi"),
    },
}
```

## B.5 Add External Subsystem

```
from copy import deepcopy

import aviary.api as av
from aviary.api import Aircraft
from aviary.examples.external_subsystems.simple_weight.
    simple_weight_builder import (
    WingWeightBuilder,
)

# Max iterations set to 1 to reduce runtime of example
max_iter = 1
phase_info = deepcopy(av.default_height_energy_phase_info)
# Here we just add the simple weight system to only the pre-
    mission
phase_info['pre_mission']['external_subsystems'] = [
    WingWeightBuilder(name='wing_external')]

prob = av.AviaryProblem()

# Load aircraft and options data from user
# Allow for user overrides here
prob.load_inputs('models/test_aircraft/aircraft_for_bench_FwFm.csv
    ', phase_info)

# Have checks for clashing user inputs
# Raise warnings or errors depending on how clashing the issues
    are
prob.check_and_preprocess_inputs()

prob.add_pre_mission_systems()
```

```

prob.add_phases()

prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

prob.add_driver('SLSQP', max_iter)

prob.add_design_variables()

prob.add_objective()

prob.setup()

prob.set_initial_guesses()

prob.run_aviary_problem(suppress_solver_print=True)

print('Engine Mass', prob.get_val(av.Aircraft.Engine.MASS))
print('Wing Mass', prob.get_val(av.Aircraft.Wing.MASS))
print('Horizontal Tail Mass', prob.get_val(av.Aircraft.
    HorizontalTail.MASS))
```

## B.6 Level 3 Example: N3CC Aircraft

```
'''
NOTES:
Includes:
Takeoff, Climb, Cruise, Descent, Landing
Computed Aero
N3CC data
'''
import unittest

import dymos as dm
import numpy as np
import openmdao.api as om
import scipy.constants as _units
```

```python
from openmdao.utils.assert_utils import assert_near_equal
from openmdao.utils.testing_utils import use_tempdirs
from openmdao.utils.testing_utils import require_pyoptsparse

from aviary.mission.flops_based.phases.build_landing import Landing
from aviary.mission.flops_based.phases.build_takeoff import Takeoff
from aviary.utils.functions import set_aviary_initial_values
from aviary.utils.test_utils.assert_utils import warn_timeseries_near_equal
from aviary.validation_cases.validation_tests import get_flops_inputs
from aviary.variable_info.functions import setup_trajectory_params
from aviary.utils.aviary_values import AviaryValues
from aviary.variable_info.variables_in import VariablesIn
from aviary.mission.flops_based.phases.energy_phase import EnergyPhase

from aviary.variable_info.variables import Aircraft, Dynamic, Mission
from aviary.subsystems.premission import CorePreMission
from aviary.interface.default_phase_info.height_energy import default_premission_subsystems, default_mission_subsystems
from aviary.utils.preprocessors import preprocess_crewpayload

try:
    import pyoptsparse
except ImportError:
    pyoptsparse = None


# benchmark for simple sizing problem on the N3CC
def run_trajectory(sim=True):
    prob = om.Problem(model=om.Group())
    if pyoptsparse:
        driver = prob.driver = om.pyOptSparseDriver()
        driver.options["optimizer"] = "SNOPT"
        # driver.declare_coloring()  # currently disabled pending resolve of issue 2507
```

```python
    if driver.options["optimizer"] == "SNOPT":
        driver.opt_settings["Major iterations limit"] = 45
        driver.opt_settings["Major optimality tolerance"] = 1e-4
        driver.opt_settings["Major feasibility tolerance"] = 1e-6
        driver.opt_settings["iSumm"] = 6
    elif driver.options["optimizer"] == "IPOPT":
        driver.opt_settings["max_iter"] = 100
        driver.opt_settings["tol"] = 1e-3
        driver.opt_settings['print_level'] = 4

else:
    driver = prob.driver = om.ScipyOptimizeDriver()
    opt_settings = prob.driver.opt_settings

    driver.options['optimizer'] = 'SLSQP'
    opt_settings['maxiter'] = 100
    opt_settings['ftol'] = 5.0e-3
    opt_settings['eps'] = 1e-2

##########################################
# Aircraft Input Variables and Options   #
##########################################

aviary_inputs = get_flops_inputs('N3CC')

aviary_inputs.set_val(Mission.Landing.LIFT_COEFFICIENT_MAX,
                      2.4, units="unitless")
aviary_inputs.set_val(Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                      2.0, units="unitless")
aviary_inputs.set_val(
    Mission.Takeoff.ROLLING_FRICTION_COEFFICIENT,
    val=.0175, units="unitless")

takeoff_fuel_burned = 577  # lbm TODO: where should this get connected from?
takeoff_thrust_per_eng = 24555.5  # lbf TODO: where should this get connected from?
takeoff_L_over_D = 17.35  # TODO: should this come from aero?
```

```

    aviary_inputs.set_val(Mission.Takeoff.FUEL_SIMPLE,
                          takeoff_fuel_burned, units='lbm')
    aviary_inputs.set_val(Mission.Takeoff.LIFT_OVER_DRAG,
                          takeoff_L_over_D, units="unitless")
    aviary_inputs.set_val(Mission.Design.THRUST_TAKEOFF_PER_ENG,
                          takeoff_thrust_per_eng, units='lbf')

    alt_airport = 0  # ft

    alt_i_climb = 0*_units.foot  # m
    alt_f_climb = 35000.0*_units.foot  # m
    mass_i_climb = 131000*_units.lb  # kg
    mass_f_climb = 126000*_units.lb  # kg
    # initial mach set to lower value so it can intersect with
   takeoff end mach
    # mach_i_climb = 0.3
    mach_i_climb = 0.2
    mach_f_climb = 0.79
    distance_i_climb = 0*_units.nautical_mile  # m
    distance_f_climb = 160.3*_units.nautical_mile  # m
    t_i_climb = 2 * _units.minute  # sec
    t_f_climb = 26.20*_units.minute  # sec
    t_duration_climb = t_f_climb - t_i_climb

    alt_i_cruise = 35000*_units.foot  # m
    alt_f_cruise = 35000*_units.foot  # m
    alt_min_cruise = 35000*_units.foot  # m
    alt_max_cruise = 35000*_units.foot  # m
    mass_i_cruise = 126000*_units.lb  # kg
    mass_f_cruise = 102000*_units.lb  # kg
    cruise_mach = 0.79
    distance_i_cruise = 160.3*_units.nautical_mile  # m
    distance_f_cruise = 3243.9*_units.nautical_mile  # m
    t_i_cruise = 26.20*_units.minute  # sec
    t_f_cruise = 432.38*_units.minute  # sec
    t_duration_cruise = t_f_cruise - t_i_cruise

    alt_i_descent = 35000*_units.foot
    # final altitude set to 35 to ensure landing is feasible point
```

```
    # alt_f_descent = 0*_units.foot
    alt_f_descent = 35*_units.foot
    mach_i_descent = 0.79
    mach_f_descent = 0.3
    mass_i_descent = 102000*_units.pound
    mass_f_descent = 101000*_units.pound
    distance_i_descent = 3243.9*_units.nautical_mile
    distance_f_descent = 3378.7*_units.nautical_mile
    t_i_descent = 432.38*_units.minute
    t_f_descent = 461.62*_units.minute
    t_duration_descent = t_f_descent - t_i_descent

    ##########################
    # Design Variables       #
    ##########################

    # Nudge it a bit off the correct answer to verify that the optimize takes us there.
    aviary_inputs.set_val(Mission.Design.GROSS_MASS, 135000.0, units='lbm')

    prob.model.add_design_var(Mission.Design.GROSS_MASS, units='lbm',
                              lower=100000.0, upper=200000.0, ref=135000)

    takeoff_options = Takeoff(
        airport_altitude=alt_airport,  # ft
        # no units
        num_engines=aviary_inputs.get_val(Aircraft.Engine.NUM_ENGINES)
    )

    ##################
    # Define Phases  #
    ##################
    num_segments_climb = 6
    num_segments_cruise = 1
    num_segments_descent = 5

```

```
    climb_seg_ends, _ = dm.utils.lgl.lgl(num_segments_climb + 1)
    descent_seg_ends, _ = dm.utils.lgl.lgl(num_segments_descent + 1)

    transcription_climb = dm.Radau(
        num_segments=num_segments_climb, order=3, compressed=True,
        segment_ends=climb_seg_ends)
    transcription_cruise = dm.Radau(
        num_segments=num_segments_cruise, order=3, compressed=True)
    transcription_descent = dm.Radau(
        num_segments=num_segments_descent, order=3, compressed=True,
        segment_ends=descent_seg_ends)

    climb_options = EnergyPhase(
        'test_climb',
        user_options=AviaryValues({
            'initial_altitude': (alt_i_climb, 'm'),
            'final_altitude': (alt_f_climb, 'm'),
            'initial_mach': (mach_i_climb, 'unitless'),
            'final_mach': (mach_f_climb, 'unitless'),
            'fix_initial': (False, 'unitless'),
            'input_initial': (True, 'unitless'),
            'use_polynomial_control': (False, 'unitless'),
        }),
        core_subsystems=default_mission_subsystems,
        subsystem_options={'core_aerodynamics': {'method': 'computed'}},
        transcription=transcription_climb,
    )

    cruise_options = EnergyPhase(
        'test_cruise',
        user_options=AviaryValues({
            'initial_altitude': (alt_min_cruise, 'm'),
            'final_altitude': (alt_max_cruise, 'm'),
            'initial_mach': (cruise_mach, 'unitless'),
            'final_mach': (cruise_mach, 'unitless'),
            'required_available_climb_rate': (300, 'ft/min'),
```

```
            'fix_initial': (False, 'unitless'),
        }),
        core_subsystems=default_mission_subsystems,
        subsystem_options={'core_aerodynamics': {'method': 'computed'}},
        transcription=transcription_cruise,
    )

    descent_options = EnergyPhase(
        'test_descent',
        user_options=AviaryValues({
            'final_altitude': (alt_f_descent, 'm'),
            'initial_altitude': (alt_i_descent, 'm'),
            'initial_mach': (mach_i_descent, 'unitless'),
            'final_mach': (mach_f_descent, 'unitless'),
            'fix_initial': (False, 'unitless'),
            'use_polynomial_control': (False, 'unitless'),
        }),
        core_subsystems=default_mission_subsystems,
        subsystem_options={'core_aerodynamics': {'method': 'computed'}},
        transcription=transcription_descent,
    )

    landing_options = Landing(
        ref_wing_area=aviary_inputs.get_val(Aircraft.Wing.AREA, units='ft**2'),
        Cl_max_ldg=aviary_inputs.get_val(
            Mission.Landing.LIFT_COEFFICIENT_MAX)  # no units
    )

    preprocess_crewpayload(aviary_inputs)

    # Upstream static analysis for aero
    prob.model.add_subsystem(
        'pre_mission',
        CorePreMission(aviary_options=aviary_inputs,
                       subsystems=default_premission_subsystems),
        promotes_inputs=['aircraft:*', 'mission:*'],
        promotes_outputs=['aircraft:*', 'mission:*'])
```

```

    # directly connect phases (strong_couple = True), or use linkage constraints (weak
    # coupling / strong_couple=False)
    strong_couple = False

    takeoff = takeoff_options.build_phase(False)

    climb = climb_options.build_phase(aviary_options=aviary_inputs)

    cruise = cruise_options.build_phase(aviary_options=aviary_inputs)

    descent = descent_options.build_phase(aviary_options=aviary_inputs)

    landing = landing_options.build_phase(False)

    prob.model.add_subsystem(
        'takeoff', takeoff, promotes_inputs=['aircraft:*', 'mission:*'],
        promotes_outputs=['mission:*'])

    traj = prob.model.add_subsystem('traj', dm.Trajectory())

    # if fix_initial is false, can we always set input_initial to be true for
    # necessary states, and then ignore if we use a linkage?
    climb.set_time_options(fix_initial=True, fix_duration=False, units='s',
                           duration_bounds=(t_duration_climb*0.5, t_duration_climb*2),
                           duration_ref=t_duration_climb)
    cruise.set_time_options(fix_initial=False, fix_duration=False, units='s',
                            duration_bounds=(t_duration_cruise*0.5, t_duration_cruise*2),
                            duration_ref=t_duration_cruise,
```

```
                             initial_bounds=(t_duration_climb*0.5,
t_duration_climb*2))
    descent.set_time_options(
        fix_initial=False, fix_duration=False, units='s',
        duration_bounds=(t_duration_descent*0.5,
t_duration_descent*2),
        duration_ref=t_duration_descent,
        initial_bounds=(
            (t_duration_cruise + t_duration_climb)*0.5,
            (t_duration_cruise + t_duration_climb)*2))

    traj.add_phase('climb', climb)

    traj.add_phase('cruise', cruise)

    traj.add_phase('descent', descent)

    prob.model.add_subsystem(
        'landing', landing, promotes_inputs=['aircraft:*', '
mission:*'],
        promotes_outputs=['mission:*'])

    traj.link_phases(["climb", "cruise", "descent"], [
                     "time", Dynamic.Mission.MASS, Dynamic.Mission
.DISTANCE], connected=strong_couple)

    traj = setup_trajectory_params(prob.model, traj, aviary_inputs
)

    ##################################
    # Connect in Takeoff and Landing #
    ##################################
    prob.model.connect(Mission.Takeoff.FINAL_MASS,
                       'traj.climb.initial_states:mass')
    prob.model.connect(Mission.Takeoff.GROUND_DISTANCE,
                       'traj.climb.initial_states:distance')

    prob.model.connect('traj.descent.states:mass',
                       Mission.Landing.TOUCHDOWN_MASS, src_indices
=[-1])
```

```
    # TODO: approach velocity should likely be connected
    prob.model.connect('traj.descent.control_values:altitude', Mission.Landing.INITIAL_ALTITUDE,
                       src_indices=[-1])

    ##########################
    # Constraints            #
    ##########################

    ecomp = om.ExecComp('fuel_burned = initial_mass - descent_mass_final',
                        initial_mass={'units': 'lbm', 'shape': 1},
                        descent_mass_final={'units': 'lbm', 'shape': 1},
                        fuel_burned={'units': 'lbm', 'shape': 1})

    prob.model.add_subsystem('fuel_burn', ecomp,
                             promotes_inputs=[
                                 ('initial_mass', Mission.Design.GROSS_MASS)],
                             promotes_outputs=['fuel_burned'])

    prob.model.connect("traj.descent.states:mass",
                       "fuel_burn.descent_mass_final", src_indices=[-1])

    # TODO: need to add some sort of check that this value is less than the fuel capacity
    # TODO: need to update this with actual FLOPS value, this gives unrealistic
    # appearance of accuracy
    # TODO: the overall_fuel variable is the burned fuel plus the reserve, but should
    # also include the unused fuel, and the hierarchy variable name should be more clear
    ecomp = om.ExecComp('overall_fuel = fuel_burned + fuel_reserve',
                        fuel_burned={'units': 'lbm', 'shape': 1},
                        fuel_reserve={'units': 'lbm', 'val': 2173.},
```

```
                             overall_fuel={'units': 'lbm'})
    prob.model.add_subsystem('fuel_calc', ecomp,
                             promotes_inputs=['fuel_burned'],
                             promotes_outputs=['overall_fuel'])

    ecomp = om.ExecComp(
        'mass_resid = operating_empty_mass + overall_fuel + payload_mass '
        '- initial_mass',
        operating_empty_mass={'units': 'lbm'},
        overall_fuel={'units': 'lbm'},
        payload_mass={'units': 'lbm'},
        initial_mass={'units': 'lbm'},
        mass_resid={'units': 'lbm'})

    prob.model.add_subsystem(
        'mass_constraint', ecomp,
        promotes_inputs=[
            ('operating_empty_mass', Aircraft.Design.OPERATING_MASS),
            'overall_fuel',
            ('payload_mass', Aircraft.CrewPayload.TOTAL_PAYLOAD_MASS),
            ('initial_mass', Mission.Design.GROSS_MASS)],
        promotes_outputs=['mass_resid'])

    prob.model.add_constraint('mass_resid', equals=0.0, ref=1.0)

    ##########################
    # Add Objective Function #
    ##########################

    # This is an example of a overall mission objective
    # create a compound objective that minimizes climb time and maximizes final mass
    # we are maxing final mass b/c we don't have an independent value for fuel_mass yet
    # we are going to normalize these (making each of the sub-objectives approx = 1 )
    # TODO: change the scaling on climb_duration
```

```
    prob.model.add_subsystem(
        "regularization",
        om.ExecComp(
            "reg_objective = fuel_mass/1500",
            reg_objective=0.0,
            fuel_mass={"units": "lbm", "shape": 1},
        ),
        promotes_outputs=['reg_objective']
    )
    # connect the final mass from cruise into the objective
    prob.model.connect(Mission.Design.FUEL_MASS, "regularization.
fuel_mass")

    prob.model.add_objective('reg_objective', ref=1)

    # Set initial default values for all LEAPS aircraft variables.
    set_aviary_initial_values(prob.model, aviary_inputs)

    prob.model.add_subsystem(
        'input_sink',
        VariablesIn(aviary_options=aviary_inputs),
        promotes_inputs=['*'],
        promotes_outputs=['*']
    )

    prob.setup(force_alloc_complex=True)

    ###########################################
    # Intial Settings for States and Controls #
    ###########################################

    prob.set_val('traj.climb.t_initial', t_i_climb, units='s')
    prob.set_val('traj.climb.t_duration', t_duration_climb, units=
's')

    prob.set_val('traj.climb.controls:altitude', climb.interp(
        Dynamic.Mission.ALTITUDE, ys=[alt_i_climb, alt_f_climb]),
units='m')
    prob.set_val(
        'traj.climb.controls:mach', climb.interp(
```

```
            Dynamic.Mission.MACH, ys=[mach_i_climb, mach_f_climb]), units='unitless')
    prob.set_val('traj.climb.states:mass', climb.interp(
        Dynamic.Mission.MASS, ys=[mass_i_climb, mass_f_climb]), units='kg')
    prob.set_val('traj.climb.states:distance', climb.interp(
        Dynamic.Mission.DISTANCE, ys=[distance_i_climb, distance_f_climb]), units='m')

    prob.set_val('traj.cruise.t_initial', t_i_cruise, units='s')
    prob.set_val('traj.cruise.t_duration', t_duration_cruise, units='s')

    prob.set_val('traj.cruise.polynomial_controls:altitude', cruise.interp(
        Dynamic.Mission.ALTITUDE, ys=[alt_i_cruise, alt_f_cruise]), units='m')
    prob.set_val(
        'traj.cruise.polynomial_controls:mach', cruise.interp(
            Dynamic.Mission.MACH, ys=[cruise_mach, cruise_mach]), units='unitless')
    prob.set_val('traj.cruise.states:mass', cruise.interp(
        Dynamic.Mission.MASS, ys=[mass_i_cruise, mass_f_cruise]), units='kg')
    prob.set_val('traj.cruise.states:distance', cruise.interp(
        Dynamic.Mission.DISTANCE, ys=[distance_i_cruise, distance_f_cruise]), units='m')

    prob.set_val('traj.descent.t_initial', t_i_descent, units='s')
    prob.set_val('traj.descent.t_duration', t_duration_descent, units='s')

    prob.set_val('traj.descent.controls:altitude', descent.interp(
        Dynamic.Mission.ALTITUDE, ys=[alt_i_descent, alt_f_descent]), units='m')
    prob.set_val(
        'traj.descent.controls:mach', descent.interp(
            Dynamic.Mission.MACH, ys=[mach_i_descent, mach_f_descent]), units='unitless')
    prob.set_val('traj.descent.states:mass', descent.interp(
```

```
        Dynamic.Mission.MASS, ys=[mass_i_descent, mass_f_descent]), units='kg')
    prob.set_val('traj.descent.states:distance', descent.interp(
        Dynamic.Mission.DISTANCE, ys=[distance_i_descent, distance_f_descent]), units='m')

    # Turn off solver printing so that the SNOPT output is readable.
    prob.set_solver_print(level=0)

    dm.run_problem(prob, simulate=sim, make_plots=False, simulate_kwargs={
                   'times_per_seg': 100, 'atol': 1e-9, 'rtol': 1e-9},
                   solution_record_file='N3CC_sizing.db')
    # prob.run_model()
    # z=prob.check_totals(method='cs', step=2e-40, compact_print=False)
    # exit()
    prob.record("final")
    prob.cleanup()

    return prob


@use_tempdirs
class ProblemPhaseTestCase(unittest.TestCase):

    @require_pyoptsparse(optimizer="SNOPT")
    def bench_test_sizing_N3CC(self):

        prob = run_trajectory(sim=False)

        times_climb = prob.get_val('traj.climb.timeseries.time', units='s')
        altitudes_climb = prob.get_val(
            'traj.climb.timeseries.altitude', units='m')
        masses_climb = prob.get_val('traj.climb.timeseries.mass', units='kg')
```

```
    distances_climb = prob.get_val('traj.climb.timeseries.distance', units='m')
    velocities_climb = prob.get_val(
        'traj.climb.timeseries.velocity', units='m/s')
    thrusts_climb = prob.get_val('traj.climb.timeseries.thrust_net_total', units='N')
    times_cruise = prob.get_val('traj.cruise.timeseries.time', units='s')
    altitudes_cruise = prob.get_val(
        'traj.cruise.timeseries.altitude', units='m')
    masses_cruise = prob.get_val('traj.cruise.timeseries.mass', units='kg')
    distances_cruise = prob.get_val('traj.cruise.timeseries.distance', units='m')
    velocities_cruise = prob.get_val(
        'traj.cruise.timeseries.velocity', units='m/s')
    thrusts_cruise = prob.get_val(
        'traj.cruise.timeseries.thrust_net_total', units='N')
    times_descent = prob.get_val('traj.descent.timeseries.time', units='s')
    altitudes_descent = prob.get_val(
        'traj.descent.timeseries.altitude', units='m')
    masses_descent = prob.get_val('traj.descent.timeseries.mass', units='kg')
    distances_descent = prob.get_val('traj.descent.timeseries.distance', units='m')
    velocities_descent = prob.get_val(
        'traj.descent.timeseries.velocity', units='m/s')
    thrusts_descent = prob.get_val(
        'traj.descent.timeseries.thrust_net_total', units='N')

    expected_times_s_climb = [[120.], [163.76325828], [224.14762249], [243.2590673],
                              [243.2590673], [336.41088085], [464.94137099],
                              [505.62083166], [505.62083166], [626.47618404],
                              [793.2319034], [846.00951843], [846.00951843],
```

```
                                    [966.8648708], [1133.62059016],
    [1186.39820519],
                                    [1186.39820519],
    [1279.55001873], [1408.08050887],
                                    [1448.75996955],
    [1448.75996955], [1492.52322782],
                                    [1552.90759204],
    [1572.01903685]]
         expected_altitudes_m_climb = [[10.668], [0.],
    [835.43072799], [1237.19027083],
                                        [1237.19027083],
    [3154.43458807], [5478.53141388],
                                        [6035.03506592],
    [6035.03506592], [7369.93558792],
                                        [8611.93242084],
    [8931.907682], [8931.907682],
                                        [9555.56028912],
    [10196.52547754], [10341.80261401],
                                        [10341.80261401],
    [10526.85288051], [10653.9273369],
                                        [10668.], [10668.],
    [10668.], [10664.89939949], [10668.]]
         expected_masses_kg_climb = [[58331.64520977],
    [58289.14326023], [58210.1552262],
                                      [58183.99841695],
    [58183.99841695], [58064.21371439],
                                      [57928.02594919],
    [57893.85802859], [57893.85802859],
                                      [57803.5753272],
    [57701.17154376], [57671.70469428],
                                      [57671.70469428],
    [57609.99954455], [57532.67046087],
                                      [57509.28816794],
    [57509.28816794], [57468.36184014],
                                      [57411.39726843],
    [57392.97463799], [57392.97463799],
                                      [57372.80054331],
    [57344.30023996], [57335.11578186]]
         expected_distances_m_climb = [[1459.66454213],
    [6138.11750563], [15269.84014039],
```

```
                                    [18334.78691191],
    [18334.78691191], [34224.35398213],
                                    [57056.15756331],
    [64259.82264908], [64259.82264908],
                                    [85983.44219477],
    [116781.82372802], [
                                        126787.42506431],
                                    [126787.42506431],
    [149770.216723], [181655.376959],
                                    [191785.63505195],
    [191785.63505195], [
                                        210054.32022093],
                                    [236459.08240245],
    [245218.76016711], [
                                        245218.76016711],
                                    [254896.51199279],
    [268679.73823007], [273140.04867091]]
        expected_velocities_ms_climb = [[77.19291754],
    [132.35228283], [162.28279625],
                                      [166.11250634],
    [166.11250634], [176.30796789],
                                      [178.55049791],
    [178.67862048], [178.67862048],
                                      [181.75616771],
    [188.37687907], [189.78671051],
                                      [189.78671051],
    [190.67789763], [191.79702196],
                                      [193.51306342],
    [193.51306342], [199.28763405],
                                      [212.62699415],
    [218.00379223], [218.00379223],
                                      [224.21350812],
    [232.18822869], [234.25795132]]
        expected_thrusts_N_climb = [[82387.90445676],
    [105138.56964482], [113204.81413537],
                                  [110631.69881503],
    [110631.69881503], [97595.73505557],
                                  [74672.35033284],
    [68787.54639919], [68787.54639919],
```

```
                                      [56537.02658008], [46486.74146611], [43988.99250831],
                                      [43988.99250831], [39814.93772493], [36454.43040965],
                                      [35877.85248136], [35877.85248136], [35222.20684522],
                                      [34829.78305946], [34857.84665827], [34857.84665827],
                                      [34988.40432301], [35243.28462529], [35286.14075168]]

     expected_times_s_cruise = [[1572.01903685], [10224.85688577],
                                [22164.00704809], [25942.70725365]]
     expected_altitudes_m_cruise = [[10668.], [10668.],
                                    [10668.], [10668.]]
     expected_masses_kg_cruise = [[57335.11578186], [53895.3524649],
                                  [49306.34176818], [47887.72131688]]
     expected_distances_m_cruise = [[273140.04867091], [2300136.11626779],
                                    [5096976.9738027], [5982167.54261146]]
     expected_velocities_ms_cruise = [[234.25795132], [234.25795132],
                                      [234.25795132], [234.25795132]]
     expected_thrusts_N_cruise = [[28998.46944214], [28027.44677784],
                                  [26853.54343662], [26522.10071819]]

     expected_times_s_descent = [[25942.70725365], [26017.49838255], [26120.69487534],
                                 [26153.35621248], [26153.35621248], [26306.10095576],
                                 [26516.85752584], [26583.56125968], [26583.56125968],
```

```
                                   [26765.15964678],
    [27015.72835621], [27095.03249603],
                                   [27095.03249603],
    [27247.77723931], [27458.5338094],
                                   [27525.23754324],
    [27525.23754324], [27600.02867214],
                                   [27703.22516493],
    [27735.88650207]]
        expected_altitudes_m_descent = [[10668.], [10668.],
    [10065.85989164],
                                       [9813.33438203],
    [9813.33438203], [8725.84180063],
                                       [7399.4358998],
    [6995.36734403], [6995.36734403],
                                       [5880.98032937],
    [4300.0193441], [3787.54388607],
                                       [3787.54388607],
    [2768.24427675], [1328.48487414],
                                       [872.88952055],
    [872.88952055], [393.61709386],
                                       [10.668], [10.668]]
        expected_masses_kg_descent = [[47887.72131688],
    [47887.72131688], [47887.72131688],
                                     [47887.72131688],
    [47887.72131688], [47887.72131688],
                                     [47887.72131688],
    [47887.72131688], [47887.72131688],
                                     [47887.87994829],
    [47886.14050369], [47884.40804261],
                                     [47884.40804261],
    [47872.68009732], [47849.34258173],
                                     [47842.09391697],
    [47842.09391697], [47833.150133],
                                     [47820.60083267],
    [47816.69389115]]
        expected_distances_m_descent = [[5982167.54261146],
    [5998182.9986382], [6017437.15474761],
                                       [6023441.80485498],
    [6023441.80485498], [
            6050450.80904601],
```

```
            [6085202.09693457], [6095472.55993089], [
            6095472.55993089],
            [6122427.64793619], [6158176.53029461], [
            6169433.26612421],
            [6169433.26612421], [6191073.79296936], [
            6220839.65410345],
            [6230195.89769052], [6230195.89769052], [
            6240573.15704622],
            [6253740.15118502], [6257352.4]]
        expected_velocities_ms_descent = [[234.25795132],
    [197.64415171], [182.5029101],
                                          [181.15994177],
    [181.15994177], [172.42254637],
                                          [156.92424445],
    [152.68023428], [152.68023428],
                                          [145.15327267],
    [141.83129659], [141.72294853],
                                          [141.72294853],
    [141.82174008], [140.7384589],
                                          [139.65952688],
    [139.65952688], [136.46721905],
                                          [115.68627038],
    [102.07377559]]
        expected_thrusts_N_descent = [[0.], [8.11038373e-13],
    [4.89024663e-13],
                                      [-3.80747087e-14], [0.],
    [1.10831527e-12],
                                      [7.77983996e-13],
    [-6.64041439e-16], [0.],
                                      [0.], [-5.7108371e-14],
    [-8.28280737e-14],
                                      [0.], [1.14382967e-13],
    [5.36023712e-14],
                                      [-3.04972888e-14], [0.],
    [-1.0682523e-13],
                                      [-6.30544276e-14],
    [5.37050667e-15]]
        expected_times_s_climb = np.array(expected_times_s_climb)
        expected_altitudes_m_climb = np.array(
    expected_altitudes_m_climb)
```

```
        expected_masses_kg_climb = np.array(
    expected_masses_kg_climb)
        expected_distances_m_climb = np.array(
    expected_distances_m_climb)
        expected_velocities_ms_climb = np.array(
    expected_velocities_ms_climb)
        expected_thrusts_N_climb = np.array(
    expected_thrusts_N_climb)

        expected_times_s_cruise = np.array(expected_times_s_cruise
    )
        expected_altitudes_m_cruise = np.array(
    expected_altitudes_m_cruise)
        expected_masses_kg_cruise = np.array(
    expected_masses_kg_cruise)
        expected_distances_m_cruise = np.array(
    expected_distances_m_cruise)
        expected_velocities_ms_cruise = np.array(
    expected_velocities_ms_cruise)
        expected_thrusts_N_cruise = np.array(
    expected_thrusts_N_cruise)

        expected_times_s_descent = np.array(
    expected_times_s_descent)
        expected_altitudes_m_descent = np.array(
    expected_altitudes_m_descent)
        expected_masses_kg_descent = np.array(
    expected_masses_kg_descent)
        expected_distances_m_descent = np.array(
    expected_distances_m_descent)
        expected_velocities_ms_descent = np.array(
    expected_velocities_ms_descent)
        expected_thrusts_N_descent = np.array(
    expected_thrusts_N_descent)

        # Check Objective and other key variables to a reasonably
    tight tolerance.

        # TODO: update truth values once everyone is using latest
    Dymos
```

```
        rtol = 5e-2

        # Mass at the end of Descent
        assert_near_equal(masses_descent[-1],
                          expected_masses_kg_descent[-1],
 tolerance=rtol)

        # Range at the end of Descent
        assert_near_equal(distances_descent[-1],
                          expected_distances_m_descent[-1],
 tolerance=rtol)

        # Flight time
        assert_near_equal(times_descent[-1],
                          expected_times_s_descent[-1], tolerance=
 rtol)

        # Check mission values.

        # NOTE rtol = 0.05 = 5% different from truth (first
 timeseries)
        #      atol = 2 = no more than +/-2 meter/second/kg
 difference between values
        #      atol_altitude - 30 ft. There is occasional time-
 shifting with the N3CC
        #                    model during climb and descent so
 we need a looser
        #                    absolute tolerance for the points
 near the ground.
        rtol = .05
        atol = 2.0
        atol_altitude = 30.0

        # FLIGHT PATH
        # CLIMB
        warn_timeseries_near_equal(
            times_climb, altitudes_climb, expected_times_s_climb,
            expected_altitudes_m_climb, abs_tolerance=
 atol_altitude, rel_tolerance=rtol)
        warn_timeseries_near_equal(
```

```
            times_climb, masses_climb, expected_times_s_climb,
            expected_masses_kg_climb, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_climb, distances_climb, expected_times_s_climb,
            expected_distances_m_climb, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_climb, velocities_climb, expected_times_s_climb,
            expected_velocities_ms_climb, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_climb, thrusts_climb, expected_times_s_climb,
            expected_thrusts_N_climb, abs_tolerance=atol,
rel_tolerance=rtol)

        # CRUISE
        warn_timeseries_near_equal(
            times_cruise, altitudes_cruise,
expected_times_s_cruise,
            expected_altitudes_m_cruise, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_cruise, masses_cruise, expected_times_s_cruise,
            expected_masses_kg_cruise, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_cruise, distances_cruise,
expected_times_s_cruise,
            expected_distances_m_cruise, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_cruise, velocities_cruise,
expected_times_s_cruise,
            expected_velocities_ms_cruise, abs_tolerance=atol,
rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_cruise, thrusts_cruise, expected_times_s_cruise,
            expected_thrusts_N_cruise, abs_tolerance=atol,
rel_tolerance=rtol)
```

```

        # DESCENT
        warn_timeseries_near_equal(
            times_descent, altitudes_descent, expected_times_s_descent,
            expected_altitudes_m_descent, abs_tolerance=atol_altitude,
            rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_descent, masses_descent, expected_times_s_descent,
            expected_masses_kg_descent, abs_tolerance=atol, rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_descent, distances_descent, expected_times_s_descent,
            expected_distances_m_descent, abs_tolerance=atol, rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_descent, velocities_descent, expected_times_s_descent,
            expected_velocities_ms_descent, abs_tolerance=atol, rel_tolerance=rtol)
        warn_timeseries_near_equal(
            times_descent, thrusts_descent, expected_times_s_descent,
            expected_thrusts_N_descent, abs_tolerance=atol, rel_tolerance=rtol)


if __name__ == '__main__':
    z = ProblemPhaseTestCase()
    z.bench_test_sizing_N3CC()
```

## B.7 Engine Deck file

```
# FLOPS-derived engine deck converted from turbofan_28_ENGDEK

```

```
Mach_Number (unitless), Altitude (ft), Throttle (unitless), Gross_Thrust (lbf), Ram_Drag (lbf), Fuel_Flow (lb/h), NOx_Rate (lb/h)
                    0.0,             0.0,                    21.0,        1446.4,             0.0,             842.2,            4.7876
                    0.0,             0.0,                    22.0,        2314.3,             0.0,             976.0,            5.1926
                    0.0,             0.0,                    24.0,        4049.9,             0.0,            1281.0,            6.6532
                    0.0,             0.0,                    26.0,        5785.6,             0.0,            1659.9,            8.2234
                    0.0,             0.0,                    30.0,        9642.7,             0.0,            2621.6,           12.0946
                    0.0,             0.0,                    34.0,       13499.8,             0.0,            3673.3,           15.0321
                    0.0,             0.0,                    38.0,       17356.9,             0.0,            4795.2,           17.5877
                    0.0,             0.0,                    42.0,       21214.0,             0.0,            5979.1,           19.8547
                    0.0,             0.0,                    46.0,       25071.1,             0.0,            7258.1,           33.3976
                    0.0,             0.0,                    48.0,       26999.7,             0.0,            7932.6,           49.2185
                    0.0,             0.0,                    50.0,       28928.1,             0.0,            8662.3,           61.9894
                    0.0,          2000.0,                    21.0,        1446.4,             0.0,             781.8,            5.2521
                    0.0,          2000.0,                    22.0,        2314.2,             0.0,             914.2,            5.1431
                    0.0,          2000.0,                    24.0,        4049.9,             0.0,            1220.2,            6.7284
                    0.0,          2000.0,                    26.0,        5785.6,             0.0,            1608.2,            8.4472
                    0.0,          2000.0,                    30.0,        9642.7,             0.0,            2572.5,           12.5205
                    0.0,          2000.0,                    34.0,       13499.7,             0.0,            3624.6,           15.6007
                    0.0,          2000.0,                    38.0,       17356.8,             0.0,            4748.2,           18.2407
```

```
                   0.0,        2000.0,                  42.0,
        21213.8,         0.0,          5953.1,           20.6168
                   0.0,        2000.0,                  46.0,
        25070.8,         0.0,          7274.3,           46.5192
                   0.0,        2000.0,                  48.0,
        26999.7,         0.0,          8005.4,           61.6004
                   0.0,        2000.0,                  50.0,
        28928.0,         0.0,          8769.0,           72.5104
                   0.0,        5000.0,                  21.0,
         1338.1,         0.0,           706.8,            6.8008
                   0.0,        5000.0,                  22.0,
         2140.9,         0.0,           828.3,              5.03
                   0.0,        5000.0,                  24.0,
         3746.7,         0.0,          1111.0,            6.2319
                   0.0,        5000.0,                  26.0,
         5352.3,         0.0,          1469.9,            7.9885
                   0.0,        5000.0,                  30.0,
         8920.6,         0.0,          2355.4,           12.0275
                   0.0,        5000.0,                  34.0,
        12488.9,         0.0,          3323.4,           15.1312
                   0.0,        5000.0,                  38.0,
        16057.1,         0.0,          4353.1,           17.7742
                   0.0,        5000.0,                  42.0,
        19625.3,         0.0,          5473.5,           20.1814
                   0.0,        5000.0,                  46.0,
        23193.5,         0.0,          6708.4,           39.1314
                   0.0,        5000.0,                  48.0,
        24977.9,         0.0,          7391.9,           57.3145
                   0.0,        5000.0,                  50.0,
        26761.8,         0.0,          8104.6,           69.2582
                   0.1,           0.0,                  21.0,
         2503.9,      1057.5,           905.8,            4.7726
                   0.1,           0.0,                  22.0,
         3575.1,      1260.9,          1088.1,            5.7209
                   0.1,           0.0,                  24.0,
         5626.3,      1576.4,          1525.4,            7.6905
                   0.1,           0.0,                  26.0,
         7615.0,      1829.5,          2001.1,            9.4545
                   0.1,           0.0,                  30.0,
        11922.9,      2280.3,          3129.8,           13.5965
```

```
                    0.1,            0.0,                  34.0,
        16145.2,          2645.4,            4326.3,            16.5878
                    0.1,            0.0,                  38.0,
        20316.5,          2959.7,            5587.6,            19.1578
                    0.1,            0.0,                  42.0,
        24452.2,          3238.4,            6932.0,            26.8701
                    0.1,            0.0,                  46.0,
        28560.1,          3489.2,            8389.7,            57.6579
                    0.1,            0.0,                  48.0,
        30603.9,          3604.5,            9194.3,            69.2477
                    0.1,            0.0,                  50.0,
        32642.3,          3714.3,           10032.8,            77.4403
                    0.1,         2000.0,                  21.0,
         2330.3,           984.5,             824.4,              4.924
                    0.1,         2000.0,                  22.0,
         3327.1,          1173.8,             991.2,             5.5336
                    0.1,         2000.0,                  24.0,
         5235.7,          1467.4,            1391.5,             7.5234
                    0.1,         2000.0,                  26.0,
         7086.3,          1702.9,            1826.8,             9.2983
                    0.1,         2000.0,                  30.0,
        11094.9,          2122.6,            2859.5,            13.4284
                    0.1,         2000.0,                  34.0,
        15023.5,          2462.3,            3960.7,            16.4467
                    0.1,         2000.0,                  38.0,
        18904.9,          2754.8,            5117.1,            19.0209
                    0.1,         2000.0,                  42.0,
        22752.9,          3013.8,            6361.1,            24.7835
                    0.1,         2000.0,                  46.0,
        26574.2,          3246.4,            7721.3,            56.6411
                    0.1,         2000.0,                  48.0,
        28476.4,          3353.7,            8462.9,            68.5093
                    0.1,         2000.0,                  50.0,
        30372.6,          3455.8,            9235.9,            77.0405
                    0.1,         5000.0,                  21.0,
         2084.6,           880.6,             738.0,             6.5816
                    0.1,         5000.0,                  22.0,
         2976.4,          1050.0,             884.8,             4.9156
                    0.1,         5000.0,                  24.0,
         4684.1,          1312.8,            1237.6,             6.8861
```

```
                    0.1,        5000.0,                      26.0,
        6339.8,         1523.6,            1621.7,              8.6447
                    0.1,        5000.0,                      30.0,
        9926.1,         1899.3,            2532.7,             12.6499
                    0.1,        5000.0,                      34.0,
       13440.9,         2203.3,            3503.2,             15.6314
                    0.1,        5000.0,                      38.0,
       16913.5,         2465.1,            4524.4,             18.1789
                    0.1,        5000.0,                      42.0,
       20356.6,         2696.9,            5621.1,             20.4797
                    0.1,        5000.0,                      46.0,
       23775.6,         2905.2,            6820.1,             42.0756
                    0.1,        5000.0,                      48.0,
       25476.8,         3001.2,            7474.2,             58.5348
                    0.1,        5000.0,                      50.0,
       27173.2,         3092.6,            8156.2,             69.8119
                    0.1,       10000.0,                      21.0,
        1721.9,          727.2,             612.6,              9.402
                    0.1,       10000.0,                      22.0,
        2458.8,          867.2,             730.7,              7.2271
                    0.1,       10000.0,                      24.0,
        3869.8,         1084.5,            1014.8,              5.7309
                    0.1,       10000.0,                      26.0,
        5237.7,         1258.8,            1324.4,              7.4445
                    0.1,       10000.0,                      30.0,
        8200.9,         1569.4,            2059.1,             10.9423
                    0.1,       10000.0,                      34.0,
       11104.9,         1820.8,            2841.8,             14.0989
                    0.1,       10000.0,                      38.0,
       13974.0,         2037.3,            3665.3,             16.5692
                    0.1,       10000.0,                      42.0,
       16818.6,         2229.1,            4548.6,             18.7998
                    0.1,       10000.0,                      46.0,
       19643.6,         2401.3,            5515.6,             21.9454
                    0.1,       10000.0,                      48.0,
       21049.0,         2480.6,            6042.3,             33.2727
                    0.1,       10000.0,                      50.0,
       22450.7,         2556.3,            6592.6,             50.9804
                    0.2,           0.0,                      21.0,
        3984.0,         2660.5,             927.8,              4.88
```

```
        0.2,        0.0,              22.0,
   5132.8,     3015.2,     1158.3,      6.0773
        0.2,        0.0,              24.0,
   7290.5,     3584.8,     1655.7,      8.2264
        0.2,        0.0,              26.0,
   9344.4,     4050.6,     2168.2,      10.048
        0.2,        0.0,              30.0,
  13714.4,     4891.5,     3335.6,     14.1832
        0.2,        0.0,              34.0,
  17928.1,     5575.9,     4564.4,     17.1365
        0.2,        0.0,              38.0,
  22048.5,     6167.1,     5839.6,     19.6518
        0.2,        0.0,              42.0,
  26104.0,     6693.4,     7176.2,      31.254
        0.2,        0.0,              46.0,
  30105.5,     7165.7,     8625.1,     60.8012
        0.2,        0.0,              48.0,
  32088.4,     7383.6,     9408.5,     71.4541
        0.2,        0.0,              50.0,
  34059.1,     7590.1,    10231.7,      78.754
        0.2,     2000.0,              21.0,
   3705.8,     2475.9,      844.5,      4.8026
        0.2,     2000.0,              22.0,
   4773.5,     2805.8,     1055.1,      5.8928
        0.2,     2000.0,              24.0,
   6779.2,     3335.6,     1509.9,      8.0591
        0.2,     2000.0,              26.0,
   8688.2,     3768.8,     1978.1,      9.8731
        0.2,     2000.0,              30.0,
  12750.0,     4551.0,     3045.6,     14.0129
        0.2,     2000.0,              34.0,
  16665.9,     5187.3,     4175.2,     16.9893
        0.2,     2000.0,              38.0,
  20495.5,     5737.3,     5341.9,     19.5044
        0.2,     2000.0,              42.0,
  24264.1,     6226.1,     6575.1,     28.1727
        0.2,     2000.0,              46.0,
  27981.8,     6663.9,     7924.2,     59.5213
        0.2,     2000.0,              48.0,
  29823.9,     6866.3,     8646.0,     70.5775
```

```
                    0.2,        2000.0,                     50.0,
        31655.3,          7058.3,            9403.9,            78.2786
                    0.2,        5000.0,                     21.0,
         3315.0,          2214.7,             755.5,              6.435
                    0.2,        5000.0,                     22.0,
         4270.4,          2510.0,             941.0,              5.268
                    0.2,        5000.0,                     24.0,
         6064.9,          2984.3,            1342.0,             7.4149
                    0.2,        5000.0,                     26.0,
         7773.0,          3372.1,            1755.4,             9.2124
                    0.2,        5000.0,                     30.0,
        11407.0,          4072.2,            2696.9,            13.2256
                    0.2,        5000.0,                     34.0,
        14910.5,          4641.7,            3693.1,            16.1681
                    0.2,        5000.0,                     38.0,
        18336.7,          5134.0,            4722.6,            18.6548
                    0.2,        5000.0,                     42.0,
        21708.3,          5571.5,            5810.0,            20.9007
                    0.2,        5000.0,                     46.0,
        25034.4,          5963.4,            6999.7,             46.548
                    0.2,        5000.0,                     48.0,
        26682.2,          6144.6,            7635.7,            60.7973
                    0.2,        5000.0,                     50.0,
        28321.1,          6316.5,            8304.7,            71.4843
                    0.2,       10000.0,                     21.0,
         2738.0,          1829.0,             626.4,             9.2804
                    0.2,       10000.0,                     22.0,
         3527.6,          2073.3,             775.5,             6.4797
                    0.2,       10000.0,                     24.0,
         5010.5,          2465.5,            1098.5,             6.2413
                    0.2,       10000.0,                     26.0,
         6421.8,          2786.1,            1431.7,             7.9908
                    0.2,       10000.0,                     30.0,
         9424.6,          3365.1,            2191.4,            11.7561
                    0.2,       10000.0,                     34.0,
        12319.4,          3836.1,            2994.6,            14.6148
                    0.2,       10000.0,                     38.0,
        15150.2,          4243.1,            3825.2,            17.0282
                    0.2,       10000.0,                     42.0,
        17936.1,          4604.8,            4701.2,            19.1798
```

```
                 0.2,     10000.0,              46.0,
    20683.4,       4929.0,        5659.9,       23.6308
                 0.2,     10000.0,              48.0,
    22045.5,       5078.8,        6173.0,       36.4956
                 0.2,     10000.0,              50.0,
    23399.5,       5221.0,        6712.2,       53.4919
                0.25,         0.0,              21.0,
     4982.9,       3711.0,         967.7,        5.1347
                0.25,         0.0,              22.0,
     6155.9,       4120.9,        1221.5,        6.4157
                0.25,         0.0,              24.0,
     8353.0,       4791.8,        1741.2,        8.5818
                0.25,         0.0,              26.0,
    10434.6,       5347.2,        2267.6,       10.7481
                0.25,         0.0,              30.0,
    14837.1,       6358.1,        3458.3,       14.5379
                0.25,         0.0,              34.0,
    19055.8,       7185.2,        4700.4,       17.4532
                0.25,         0.0,              38.0,
    23164.2,       7901.7,        5979.8,       19.9304
                0.25,         0.0,              42.0,
    27194.4,       8540.4,        7318.7,       34.2788
                0.25,         0.0,              46.0,
    31158.1,       9112.7,        8772.0,       62.9527
                0.25,         0.0,              48.0,
    33119.2,       9377.6,        9547.4,       72.8167
                0.25,         0.0,              50.0,
    35064.8,       9627.7,       10368.8,         79.61
                0.25,      2000.0,              21.0,
     4635.4,       3453.6,         881.2,        4.9403
                0.25,      2000.0,              22.0,
     5725.7,       3834.8,        1113.2,        6.2358
                0.25,      2000.0,              24.0,
     7767.9,       4458.7,        1587.7,        8.4148
                0.25,      2000.0,              26.0,
     9702.8,       4975.4,        2068.9,       10.3133
                0.25,      2000.0,              30.0,
    13794.8,       5915.8,        3157.8,       14.3694
                0.25,      2000.0,              34.0,
    17715.3,       6684.5,        4299.7,       17.3076
```

```
                  0.25,        2000.0,                 38.0,
        21533.5,         7351.2,        5470.6,          19.7854
                  0.25,        2000.0,                 42.0,
        25278.1,         7944.4,        6705.3,          30.7493
                  0.25,        2000.0,                 46.0,
        28960.8,         8474.8,        8057.1,           61.504
                  0.25,        2000.0,                 48.0,
        30782.4,         8720.7,        8773.3,          71.9783
                  0.25,        2000.0,                 50.0,
        32590.3,         8953.2,        9529.4,          79.1712
                  0.25,        5000.0,                 21.0,
         4146.8,         3089.4,         787.7,           5.8398
                  0.25,        5000.0,                 22.0,
         5122.4,         3430.7,         992.0,           5.6074
                  0.25,        5000.0,                 24.0,
         6949.9,         3989.3,        1411.0,           7.7697
                  0.25,        5000.0,                 26.0,
         8681.0,         4451.7,        1835.5,           9.5548
                  0.25,        5000.0,                 30.0,
        12342.2,         5293.4,        2795.6,          13.5758
                  0.25,        5000.0,                 34.0,
        15850.1,         5981.6,        3803.0,          16.4816
                  0.25,        5000.0,                 38.0,
        19266.4,         6578.4,        4836.6,          18.9321
                  0.25,        5000.0,                 42.0,
        22617.1,         7109.4,        5925.2,          21.9946
                  0.25,        5000.0,                 46.0,
        25911.5,         7584.2,        7117.3,           49.402
                  0.25,        5000.0,                 48.0,
        27541.0,         7804.3,        7748.5,          62.5764
                  0.25,        5000.0,                 50.0,
        29158.9,         8012.2,        8415.7,          72.7322
                  0.25,       10000.0,                 21.0,
         3424.9,         2551.5,         652.1,           8.6939
                  0.25,       10000.0,                 22.0,
         4231.3,         2833.9,         816.4,           5.6904
                  0.25,       10000.0,                 24.0,
         5741.3,         3295.8,        1153.9,           6.5846
                  0.25,       10000.0,                 26.0,
         7171.8,         3678.2,        1496.8,           8.3269
```

```
                   0.25,       10000.0,                    30.0,
        10196.9,         4374.2,            2270.9,          12.0944
                   0.25,       10000.0,                    34.0,
        13095.1,         4943.3,            3083.3,           14.918
                   0.25,       10000.0,                    38.0,
        15917.5,         5436.7,            3916.9,           17.295
                   0.25,       10000.0,                    42.0,
        18685.9,         5875.8,            4794.1,          19.4143
                   0.25,       10000.0,                    46.0,
        21407.5,         6268.5,            5754.5,          25.0166
                   0.25,       10000.0,                    48.0,
        22754.0,         6450.6,            6263.1,          38.8691
                   0.25,       10000.0,                    50.0,
        24090.7,         6622.7,            6801.4,           55.311
                    0.3,           0.0,                    21.0,
         6173.4,         4945.6,            1022.8,           5.4835
                    0.3,           0.0,                    22.0,
         7364.1,         5399.7,            1289.1,           6.7761
                    0.3,           0.0,                    24.0,
         9594.3,         6156.5,            1828.9,            8.945
                    0.3,           0.0,                    26.0,
        11702.0,         6790.9,            2370.2,          11.3695
                    0.3,           0.0,                    30.0,
        16141.2,         7956.0,            3586.7,          14.9079
                    0.3,           0.0,                    34.0,
        20374.2,         8914.9,            4845.9,          17.7911
                    0.3,           0.0,                    38.0,
        24482.0,         9748.6,            6136.4,          20.2415
                    0.3,           0.0,                    42.0,
        28498.9,        10491.9,            7487.2,          38.1853
                    0.3,           0.0,                    46.0,
        32440.2,        11158.6,            8946.6,          65.3539
                    0.3,           0.0,                    48.0,
        34385.6,        11467.0,            9722.6,          74.4559
                    0.3,           0.0,                    50.0,
        36314.3,        11758.7,           10541.1,          80.6075
                    0.3,        2000.0,                    21.0,
         5743.4,         4602.6,             931.5,           5.2941
                    0.3,        2000.0,                    22.0,
         6850.3,         5024.9,            1175.0,           6.6001
```

```
                    0.3,         2000.0,                     24.0,
          8923.2,          5728.8,              1668.1,            8.7823
                    0.3,         2000.0,                     26.0,
         10882.3,          6318.8,              2161.9,           11.0244
                    0.3,         2000.0,                     30.0,
         15008.5,          7402.8,              3275.4,            14.742
                    0.3,         2000.0,                     34.0,
         18942.1,          8294.0,              4433.1,           17.6473
                    0.3,         2000.0,                     38.0,
         22759.9,          9069.6,              5614.1,           20.0977
                    0.3,         2000.0,                     42.0,
         26492.4,          9759.9,              6860.5,           34.3047
                    0.3,         2000.0,                     46.0,
         30152.8,         10377.9,              8215.0,           63.8559
                    0.3,         2000.0,                     48.0,
         31959.4,         10663.8,              8934.3,           73.6791
                    0.3,         2000.0,                     50.0,
         33752.0,         10934.9,              9687.3,           80.2171
                    0.3,         5000.0,                     21.0,
          5137.9,          4117.3,               832.0,            5.0092
                    0.3,         5000.0,                     22.0,
          6128.4,          4495.5,              1046.5,            5.9682
                    0.3,         5000.0,                     24.0,
          7983.1,          5125.5,              1481.4,            8.1293
                    0.3,         5000.0,                     26.0,
          9735.9,          5653.8,              1917.6,            9.9046
                    0.3,         5000.0,                     30.0,
         13427.5,          6623.9,              2899.2,           13.9423
                    0.3,         5000.0,                     34.0,
         16946.8,          7421.7,              3920.4,           16.8161
                    0.3,         5000.0,                     38.0,
         20362.5,          8115.9,              4962.2,           19.2379
                    0.3,         5000.0,                     42.0,
         23701.7,          8733.8,              6061.4,           23.5209
                    0.3,         5000.0,                     46.0,
         26976.2,          9286.8,              7256.0,           52.5299
                    0.3,         5000.0,                     48.0,
         28592.9,          9543.0,              7890.1,           64.7543
                    0.3,         5000.0,                     50.0,
         30196.6,          9785.7,              8554.4,           74.2115
```

```
                    0.3,    10000.0,        21.0,       4243.8,    3400.8,     687.6,     7.8677
                    0.3,    10000.0,        22.0,       5062.5,    3713.6,     860.4,      4.908
                    0.3,    10000.0,        24.0,       6595.3,    4234.8,    1211.0,     6.9375
                    0.3,    10000.0,        26.0,       8043.8,    4671.6,    1563.1,     8.6667
                    0.3,    10000.0,        30.0,      11094.2,    5473.9,    2354.8,    12.4504
                    0.3,    10000.0,        34.0,      14002.0,    6133.6,    3177.8,    15.2398
                    0.3,    10000.0,        38.0,      16824.1,    6707.5,    4018.5,    17.5905
                    0.3,    10000.0,        42.0,      19583.3,    7218.6,    4904.6,    19.6927
                    0.3,    10000.0,        46.0,      22288.8,    7676.1,    5867.0,    26.9467
                    0.3,    10000.0,        48.0,      23624.2,    7887.9,    6378.4,    42.0826
                    0.3,    10000.0,        50.0,      24949.6,    8088.6,    6913.7,     57.381
                    0.3,    15000.0,        21.0,       3479.6,    2788.1,     567.1,     10.753
                    0.3,    15000.0,        22.0,       4151.5,    3045.2,     705.0,     7.7596
                    0.3,    15000.0,        24.0,       5409.3,    3473.3,     985.3,     5.6485
                    0.3,    15000.0,        26.0,       6597.7,    3831.9,    1267.2,      7.317
                    0.3,    15000.0,        30.0,       9100.3,    4490.8,    1900.3,    10.2364
                    0.3,    15000.0,        34.0,      11486.0,    5032.6,    2558.6,    13.4803
                    0.3,    15000.0,        38.0,      13801.0,    5503.8,    3231.5,    15.7326
                    0.3,    15000.0,        42.0,      16064.5,    5923.6,    3939.7,    17.7439
                    0.3,    15000.0,        46.0,      18284.0,    6299.3,    4708.7,    19.6831
```

```
                    0.3,      15000.0,                    48.0,
         19379.7,          6473.2,           5117.8,          21.3254
                    0.3,      15000.0,                    50.0,
         20466.5,          6637.9,           5546.1,          30.4095
                   0.35,          0.0,                    21.0,
          7566.3,          6374.0,           1096.3,           5.9318
                   0.35,          0.0,                    22.0,
          8770.7,          6863.1,           1379.4,           7.2407
                   0.35,          0.0,                    24.0,
         11031.6,          7693.3,           1932.5,           9.3656
                   0.35,          0.0,                    26.0,
         13168.6,          8399.6,           2472.9,          11.7648
                   0.35,          0.0,                    30.0,
         17653.6,          9705.4,           3720.3,          15.2909
                   0.35,          0.0,                    34.0,
         21914.4,         10786.9,           5002.2,          18.1525
                   0.35,          0.0,                    38.0,
         26037.3,         11730.5,           6310.8,          20.5849
                   0.35,          0.0,                    42.0,
         30057.1,         12571.0,           7691.3,          43.1208
                   0.35,          0.0,                    46.0,
         33993.9,         13328.4,           9155.7,          67.9905
                   0.35,          0.0,                    48.0,
         35932.2,         13677.3,           9939.7,          76.3155
                   0.35,          0.0,                    50.0,
         37853.7,         14009.0,          10755.8,           81.718
                   0.35,       2000.0,                    21.0,
          7039.8,          5932.0,            999.1,           5.7509
                   0.35,       2000.0,                    22.0,
          8159.4,          6386.9,           1257.8,           7.0693
                   0.35,       2000.0,                    24.0,
         10260.8,          7159.0,           1763.3,           9.2076
                   0.35,       2000.0,                    26.0,
         12247.3,          7816.1,           2255.9,          11.5847
                   0.35,       2000.0,                    30.0,
         16416.2,          9030.9,           3397.3,          15.1257
                   0.35,       2000.0,                    34.0,
         20375.6,         10036.0,           4575.8,          18.0083
                   0.35,       2000.0,                    38.0,
         24207.3,         10913.7,           5773.4,          20.4408
```

```
                    0.35,        2000.0,                  42.0,
        27941.9,        11694.1,          7047.4,           39.005
                    0.35,        2000.0,                  46.0,
        31598.2,        12396.5,          8403.2,          66.4441
                    0.35,        2000.0,                  48.0,
        33398.3,        12719.4,          9133.8,          75.6132
                    0.35,        2000.0,                  50.0,
        35183.5,        13027.6,          9884.0,          81.3882
                    0.35,        5000.0,                  21.0,
         6297.7,         5306.8,           891.4,           5.1268
                    0.35,        5000.0,                  22.0,
         7299.6,         5714.1,          1119.5,           6.4339
                    0.35,        5000.0,                  24.0,
         9179.9,         6405.3,          1565.3,           8.5499
                    0.35,        5000.0,                  26.0,
        10957.1,         6993.4,          2000.3,          10.2625
                    0.35,        5000.0,                  30.0,
        14686.9,         8080.8,          3006.9,          14.3212
                    0.35,        5000.0,                  34.0,
        18229.2,         8980.4,          4046.1,          17.1706
                    0.35,        5000.0,                  38.0,
        21657.4,         9766.2,          5102.9,          19.5757
                    0.35,        5000.0,                  42.0,
        24998.3,        10464.7,          6226.3,          25.9255
                    0.35,        5000.0,                  46.0,
        28269.4,        11093.1,          7421.8,          55.8105
                    0.35,        5000.0,                  48.0,
        29879.7,        11382.5,          8065.8,          67.2746
                    0.35,        5000.0,                  50.0,
        31477.1,        11658.5,          8727.8,          75.9179
                    0.35,       10000.0,                  21.0,
         5202.0,         4383.5,           735.2,            6.795
                    0.35,       10000.0,                  22.0,
         6030.2,         4720.6,           919.0,            5.282
                    0.35,       10000.0,                  24.0,
         7584.1,         5292.2,          1278.6,            7.346
                    0.35,       10000.0,                  26.0,
         9052.7,         5778.6,          1629.9,           9.0072
                    0.35,       10000.0,                  30.0,
        12134.6,         6677.8,          2441.6,           12.816
```

```
                   0.35,      10000.0,                34.0,
        15061.4,        7421.8,         3279.3,          15.5823
                   0.35,      10000.0,                38.0,
        17893.8,        8071.6,         4132.1,          17.9157
                   0.35,      10000.0,                42.0,
        20654.1,        8649.1,         5037.3,          20.0235
                   0.35,      10000.0,                46.0,
        23357.0,        9169.2,         6000.8,          29.5981
                   0.35,      10000.0,                48.0,
        24688.2,        9408.4,         6520.4,          45.9418
                   0.35,      10000.0,                50.0,
        26007.2,        9636.6,         7053.3,          59.5544
                   0.35,      15000.0,                21.0,
         4265.5,        3594.1,          604.8,           9.6899
                   0.35,      15000.0,                22.0,
         4945.3,        3871.2,          751.6,           6.6744
                   0.35,      15000.0,                24.0,
         6220.4,        4340.7,         1039.2,           6.0427
                   0.35,      15000.0,                26.0,
         7425.3,        4740.1,         1320.6,           7.6429
                   0.35,      15000.0,                30.0,
         9953.9,        5478.5,         1969.8,          10.8708
                   0.35,      15000.0,                34.0,
        12355.2,        6089.6,         2639.9,          13.8065
                   0.35,      15000.0,                38.0,
        14678.8,        6623.1,         3322.2,          16.0412
                   0.35,      15000.0,                42.0,
        16943.3,        7097.5,         4045.6,          18.0572
                   0.35,      15000.0,                46.0,
        19160.5,        7524.6,         4816.5,          19.9608
                   0.35,      15000.0,                48.0,
        20251.7,        7720.9,         5231.4,          22.6619
                   0.35,      15000.0,                50.0,
        21334.5,        7908.4,         5657.5,          33.1924
                    0.4,       2000.0,                21.0,
         8528.1,        7446.9,         1080.6,           6.2764
                    0.4,       2000.0,                22.0,
         9658.0,        7928.2,         1353.5,           7.5936
                    0.4,       2000.0,                24.0,
        11784.7,        8757.4,         1878.4,           9.7055
```

```
                    0.4,        2000.0,                 26.0,
         13797.7,       9473.0,           2382.4,          12.1006
                    0.4,        2000.0,                 30.0,
         18018.9,      10811.1,           3529.2,          15.5376
                    0.4,        2000.0,                 34.0,
         22012.9,      11922.0,           4732.9,          18.4036
                    0.4,        2000.0,                 38.0,
         25869.2,      12895.0,           5956.0,          20.8298
                    0.4,        2000.0,                 42.0,
         29617.1,      13759.9,           7260.5,          44.5575
                    0.4,        2000.0,                 46.0,
         33282.9,      14542.5,           8623.9,          69.2745
                    0.4,        2000.0,                 48.0,
         35081.9,      14900.1,           9368.5,          77.6817
                    0.4,        2000.0,                 50.0,
         36867.3,      15243.9,          10120.0,          82.6292
                    0.4,        5000.0,                 21.0,
          7629.4,       6662.3,            963.1,           5.6477
                    0.4,        5000.0,                 22.0,
          8640.5,       7093.1,           1203.9,           6.9524
                    0.4,        5000.0,                 24.0,
         10543.4,       7835.5,           1666.8,           9.0427
                    0.4,        5000.0,                 26.0,
         12344.4,       8476.0,           2111.2,          11.1881
                    0.4,        5000.0,                 30.0,
         16121.1,       9673.7,           3123.3,          14.7282
                    0.4,        5000.0,                 34.0,
         19694.5,      10668.0,           4184.8,          17.5597
                    0.4,        5000.0,                 38.0,
         23144.4,      11539.2,           5263.3,          19.9572
                    0.4,        5000.0,                 42.0,
         26497.4,      12313.3,           6414.0,           29.514
                    0.4,        5000.0,                 46.0,
         29777.0,      13014.0,           7616.7,          59.0834
                    0.4,        5000.0,                 48.0,
         31386.7,      13333.9,           8272.8,           70.033
                    0.4,        5000.0,                 50.0,
         32983.8,      13641.9,           8936.1,          77.8099
                    0.4,       10000.0,                 21.0,
          6302.4,       5503.6,            792.8,           5.5515
```

```
                      0.4,       10000.0,                 22.0,
       7138.3,         5860.2,             987.2,             5.7899
                      0.4,       10000.0,                 24.0,
       8710.6,         6473.9,            1360.1,             7.8236
                      0.4,       10000.0,                 26.0,
      10198.9,         7003.7,            1718.7,             9.4457
                      0.4,       10000.0,                 30.0,
      13319.7,         7994.3,            2535.4,            13.2086
                      0.4,       10000.0,                 34.0,
      16272.3,         8816.6,            3391.4,            15.9573
                      0.4,       10000.0,                 38.0,
      19122.7,         9537.0,            4262.0,            18.2845
                      0.4,       10000.0,                 42.0,
      21893.2,        10177.2,            5188.5,            20.3954
                      0.4,       10000.0,                 46.0,
      24602.9,        10756.8,            6158.4,            33.2437
                      0.4,       10000.0,                 48.0,
      25932.9,        11021.4,            6687.2,            50.2607
                      0.4,       10000.0,                 50.0,
      27252.3,        11276.0,            7221.2,            62.3136
                      0.4,       15000.0,                 21.0,
       5168.0,         4512.9,             650.8,             8.4556
                      0.4,       15000.0,                 22.0,
       5854.1,         4806.0,             806.0,             5.4657
                      0.4,       15000.0,                 24.0,
       7144.6,         5310.3,            1104.8,              6.507
                      0.4,       15000.0,                 26.0,
       8365.6,         5745.3,            1391.4,              8.064
                      0.4,       15000.0,                 30.0,
      10925.8,         6558.6,            2044.8,            11.5264
                      0.4,       15000.0,                 34.0,
      13348.1,         7234.0,            2729.6,            14.1632
                      0.4,       15000.0,                 38.0,
      15686.6,         7825.6,            3426.3,            16.3919
                      0.4,       15000.0,                 42.0,
      17959.2,         8351.4,            4166.5,            18.4104
                      0.4,       15000.0,                 46.0,
      20182.2,         8827.3,            4942.1,            20.2805
                      0.4,       15000.0,                 48.0,
      21272.7,         9044.6,            5364.6,            24.6069
```

```
                    0.4,     15000.0,                50.0,
        22355.5,          9253.8,          5792.4,         36.8914
                    0.4,     20000.0,                21.0,
         4204.2,          3671.1,           533.0,         11.3443
                    0.4,     20000.0,                22.0,
         4763.2,          3910.2,           655.9,          8.3987
                    0.4,     20000.0,                24.0,
         5814.2,          4321.4,           892.7,          5.1322
                    0.4,     20000.0,                26.0,
         6808.5,          4676.0,          1119.9,          6.6187
                    0.4,     20000.0,                30.0,
         8893.2,          5339.0,          1638.7,          9.2928
                    0.4,     20000.0,                34.0,
        10865.4,          5889.5,          2181.2,         12.2577
                    0.4,     20000.0,                38.0,
        12769.1,          6371.6,          2734.1,         14.3697
                    0.4,     20000.0,                42.0,
        14619.4,          6800.2,          3321.2,         16.2797
                    0.4,     20000.0,                46.0,
        16429.0,          7188.1,          3935.2,         18.0475
                    0.4,     20000.0,                48.0,
        17317.4,          7365.4,          4270.7,         18.9554
                    0.4,     20000.0,                50.0,
        18198.2,          7535.7,          4608.9,         19.8172
                   0.45,      5000.0,                21.0,
         9134.3,          8186.4,          1052.9,          6.2594
                   0.45,      5000.0,                22.0,
        10153.5,          8637.0,          1297.4,          7.5048
                   0.45,      5000.0,                24.0,
        12077.2,          9423.1,          1777.2,          9.5637
                   0.45,      5000.0,                26.0,
        13900.8,         10109.4,          2237.9,         11.8951
                   0.45,      5000.0,                30.0,
        17726.1,         11407.1,          3263.8,         15.2102
                   0.45,      5000.0,                34.0,
        21336.8,         12490.1,          4347.5,         18.0093
                   0.45,      5000.0,                38.0,
        24813.0,         13438.6,          5462.2,         20.4201
                   0.45,      5000.0,                42.0,
        28188.8,         14287.0,          6626.4,         34.5899
```

```
                   0.45,        5000.0,                    46.0,
        31486.0,         15056.3,            7841.9,           62.5039
                   0.45,        5000.0,                    48.0,
        33098.2,         15405.0,            8512.7,           72.9747
                   0.45,        5000.0,                    50.0,
        34700.2,         15743.1,            9179.8,           79.8154
                   0.45,       10000.0,                    21.0,
         7545.9,          6763.0,             865.0,            5.1128
                   0.45,       10000.0,                    22.0,
         8388.6,          7135.9,            1062.6,            6.3291
                   0.45,       10000.0,                    24.0,
         9978.2,          7786.1,            1449.5,            8.3334
                   0.45,       10000.0,                    26.0,
        11485.1,          8353.6,            1820.8,            9.9346
                   0.45,       10000.0,                    30.0,
        14645.9,          9426.7,            2648.1,           13.6721
                   0.45,       10000.0,                    34.0,
        17629.4,         10322.5,            3522.4,           16.3907
                   0.45,       10000.0,                    38.0,
        20501.5,         11106.9,            4421.5,           18.7295
                   0.45,       10000.0,                    42.0,
        23290.9,         11808.7,            5360.6,           20.8118
                   0.45,       10000.0,                    46.0,
        26014.7,         12444.7,            6339.6,           38.0177
                   0.45,       10000.0,                    48.0,
        27346.9,         12733.1,            6880.0,            54.809
                   0.45,       10000.0,                    50.0,
        28670.6,         13012.9,            7417.8,           65.5452
                   0.45,       15000.0,                    21.0,
         6188.1,          5546.1,             708.2,            7.0161
                   0.45,       15000.0,                    22.0,
         6879.8,          5852.6,             866.3,            5.0623
                   0.45,       15000.0,                    24.0,
         8184.3,          6386.7,            1176.1,            6.9948
                   0.45,       15000.0,                    26.0,
         9420.7,          6852.7,            1473.1,             8.533
                   0.45,       15000.0,                    30.0,
        12014.0,          7734.1,            2135.7,           11.9737
                   0.45,       15000.0,                    34.0,
        14461.5,          8469.7,            2834.4,           14.5758
```

```
                  0.45,      15000.0,                38.0,
        16817.6,        9113.8,        3553.9,        16.8157
                  0.45,      15000.0,                42.0,
        19105.8,        9690.3,        4304.2,        18.8064
                  0.45,      15000.0,                46.0,
        21340.5,       10212.6,        5087.2,        20.6468
                  0.45,      15000.0,                48.0,
        22433.2,       10449.4,        5519.7,        27.6021
                  0.45,      15000.0,                50.0,
        23518.8,       10679.1,        5949.6,        41.4721
                  0.45,      20000.0,                21.0,
         5034.5,        4512.0,         578.2,         9.9201
                  0.45,      20000.0,                22.0,
         5598.0,        4762.1,         703.5,         7.1405
                  0.45,      20000.0,                24.0,
         6660.5,        5197.7,         949.2,         5.5997
                  0.45,      20000.0,                26.0,
         7667.4,        5577.6,        1184.8,         7.0642
                  0.45,      20000.0,                30.0,
         9779.0,        6296.0,        1710.7,         9.6704
                  0.45,      20000.0,                34.0,
        11771.9,        6895.7,        2264.6,        12.6477
                  0.45,      20000.0,                38.0,
        13690.0,        7420.5,        2835.3,        14.7686
                  0.45,      20000.0,                42.0,
        15553.0,        7890.4,        3430.4,        16.6516
                  0.45,      20000.0,                46.0,
        17372.0,        8316.2,        4051.2,        18.3923
                  0.45,      20000.0,                48.0,
        18261.9,        8509.5,        4393.4,         19.298
                  0.45,      20000.0,                50.0,
        19145.4,        8696.4,        4733.9,        20.4806
                  0.45,      25000.0,                21.0,
         4060.5,        3638.9,         471.3,        12.8242
                  0.45,      25000.0,                22.0,
         4515.9,        3841.4,         569.7,        10.0682
                  0.45,      25000.0,                24.0,
         5374.3,        4193.9,         762.8,         5.6593
                  0.45,      25000.0,                26.0,
         6187.3,        4501.0,         947.5,         5.5964
```

```
            0.45,    25000.0,            30.0,
  7892.2,        5081.8,        1360.8,        8.0513
            0.45,    25000.0,            34.0,
  9501.3,        5566.7,        1795.6,       10.5022
            0.45,    25000.0,            38.0,
 11049.7,        5991.0,        2244.5,       12.7089
            0.45,    25000.0,            42.0,
 12553.9,        6370.9,        2711.4,       14.4744
            0.45,    25000.0,            46.0,
 14022.2,        6715.2,        3198.7,       16.1089
            0.45,    25000.0,            48.0,
 14740.5,        6871.3,        3466.7,       16.9602
            0.45,    25000.0,            50.0,
 15453.9,        7022.7,        3734.8,       17.7627
             0.5,    10000.0,            21.0,
  8935.3,        8163.6,         961.2,        5.8415
             0.5,    10000.0,            22.0,
  9784.6,        8549.9,        1163.5,        6.9976
             0.5,    10000.0,            24.0,
 11396.5,        9235.8,        1543.3,        8.8511
             0.5,    10000.0,            26.0,
 12924.7,        9838.0,        1929.2,       10.5792
             0.5,    10000.0,            30.0,
 16130.8,       10986.4,        2783.0,       14.2064
             0.5,    10000.0,            34.0,
 19153.1,       11950.9,        3679.1,       16.8955
             0.5,    10000.0,            38.0,
 22058.2,       12798.3,        4602.1,       19.2194
             0.5,    10000.0,            42.0,
 24877.3,       13559.7,        5557.3,       21.8787
             0.5,    10000.0,            46.0,
 27624.5,       14249.1,        6554.3,       44.2213
             0.5,    10000.0,            48.0,
 28965.7,       14561.4,        7104.9,       59.1261
             0.5,    10000.0,            50.0,
 30298.2,       14865.0,        7650.4,       69.0476
             0.5,    15000.0,            21.0,
  7328.1,        6695.3,         785.0,        5.2669
             0.5,    15000.0,            22.0,
  8025.5,        7012.9,         947.3,        5.7106
```

```
                     0.5,      15000.0,                 24.0,
     9348.0,              7576.1,          1250.9,            7.4914
                     0.5,      15000.0,                 26.0,
    10602.1,              8070.9,          1560.0,             9.017
                     0.5,      15000.0,                 30.0,
    13232.8,              9014.0,          2243.0,           12.4829
                     0.5,      15000.0,                 34.0,
    15712.4,              9806.1,          2959.9,           15.0577
                     0.5,      15000.0,                 38.0,
    18095.8,             10502.0,          3698.5,           17.2835
                     0.5,      15000.0,                 42.0,
    20408.6,             11127.3,          4462.3,           19.2512
                     0.5,      15000.0,                 46.0,
    22662.5,             11693.7,          5260.4,           22.1856
                     0.5,      15000.0,                 48.0,
    23762.9,             11950.1,          5699.8,            31.922
                     0.5,      15000.0,                 50.0,
    24855.4,             12199.1,          6135.5,            46.874
                     0.5,      20000.0,                 21.0,
     5962.3,              5447.4,           638.9,            8.1949
                     0.5,      20000.0,                 22.0,
     6530.4,              5706.6,           767.5,            5.5864
                     0.5,      20000.0,                 24.0,
     7607.5,              6165.9,          1008.8,            6.0754
                     0.5,      20000.0,                 26.0,
     8628.5,              6569.1,          1253.7,            7.5239
                     0.5,      20000.0,                 30.0,
    10770.3,              7337.8,          1795.3,           10.1237
                     0.5,      20000.0,                 34.0,
    12788.9,              7983.4,          2363.9,           13.1017
                     0.5,      20000.0,                 38.0,
    14729.0,              8550.6,          2949.4,           15.2078
                     0.5,      20000.0,                 42.0,
    16611.6,              9060.2,          3555.1,           17.0676
                     0.5,      20000.0,                 46.0,
    18446.0,              9521.6,          4187.2,           18.7919
                     0.5,      20000.0,                 48.0,
    19341.8,              9731.0,          4535.0,            19.685
                     0.5,      20000.0,                 50.0,
    20231.5,              9934.1,          4880.0,           22.2565
```

```
                    0.5,        25000.0,                    21.0,          4809.4,         4393.9,           518.7,          11.1067
                    0.5,        25000.0,                    22.0,          5268.5,         4603.8,           619.7,           8.5392
                    0.5,        25000.0,                    24.0,          6138.4,         4975.2,           809.4,           4.6585
                    0.5,        25000.0,                    26.0,          6963.0,         5301.2,          1001.5,           6.0297
                    0.5,        25000.0,                    30.0,          8692.6,         5923.0,          1426.9,           8.4555
                    0.5,        25000.0,                    34.0,         10322.4,         6444.9,          1873.7,          11.1348
                    0.5,        25000.0,                    38.0,         11888.8,         6903.5,          2334.1,          13.1197
                    0.5,        25000.0,                    42.0,         13408.7,         7315.6,          2809.6,          14.8632
                    0.5,        25000.0,                    46.0,         14889.7,         7688.8,          3305.8,          16.4821
                    0.5,        25000.0,                    48.0,         15612.7,         7857.9,          3578.7,          17.3225
                    0.5,        25000.0,                    50.0,         16331.0,         8022.2,          3850.1,          18.1143
                   0.55,        10000.0,                    21.0,         10473.2,         9708.0,          1063.1,           6.5663
                   0.55,        10000.0,                    22.0,         11331.3,        10107.0,          1273.2,            7.684
                   0.55,        10000.0,                    24.0,         12968.8,        10826.2,          1651.6,           9.4227
                   0.55,        10000.0,                    26.0,         14524.0,        11463.2,          2046.0,          11.6154
                   0.55,        10000.0,                    30.0,         17784.6,        12683.1,          2929.7,          14.7724
                   0.55,        10000.0,                    34.0,         20855.7,        13713.6,          3852.5,          17.4405
                   0.55,        10000.0,                    38.0,         23804.1,        14621.6,          4806.1,          19.7586
                   0.55,        10000.0,                    42.0,         26664.3,        15441.0,          5782.5,          24.8341
                   0.55,        10000.0,                    46.0,         29445.2,        16181.5,          6806.8,          51.3216
```

```
        0.55,       10000.0,                 48.0,    30801.6,    16517.7,     7366.0,    63.4873
        0.55,       10000.0,                 50.0,    32151.3,    16847.1,     7921.7,    72.7201
        0.55,       15000.0,                 21.0,     8589.7,     7962.3,      866.4,     5.2947
        0.55,       15000.0,                 22.0,     9294.1,     8290.2,     1034.6,     6.3704
        0.55,       15000.0,                 24.0,    10637.7,     8880.9,     1337.3,     8.0399
        0.55,       15000.0,                 26.0,    11913.6,     9404.0,     1653.0,       9.52
        0.55,       15000.0,                 30.0,    14588.5,    10405.8,     2359.7,    13.0203
        0.55,       15000.0,                 34.0,    17108.0,    11252.3,     3098.8,    15.5778
        0.55,       15000.0,                 38.0,    19526.7,    11997.8,     3861.4,    17.7957
        0.55,       15000.0,                 42.0,    21872.4,    12670.6,     4642.0,    19.7451
        0.55,       15000.0,                 46.0,    24153.7,    13278.7,     5460.9,    25.1739
        0.55,       15000.0,                 48.0,    25266.4,    13555.1,     5907.7,    37.7816
        0.55,       15000.0,                 50.0,    26373.7,    13825.7,     6352.0,    52.6518
        0.55,       20000.0,                 21.0,     6989.4,     6479.0,      703.3,      6.503
        0.55,       20000.0,                 22.0,     7563.3,     6746.5,      837.0,     5.0064
        0.55,       20000.0,                 24.0,     8657.4,     7228.1,     1077.2,     6.5972
        0.55,       20000.0,                 26.0,     9696.4,     7654.5,     1327.5,     8.0013
        0.55,       20000.0,                 30.0,    11874.4,     8471.2,     1888.0,    11.0514
        0.55,       20000.0,                 34.0,    13925.6,     9161.1,     2474.4,    13.5941
        0.55,       20000.0,                 38.0,    15894.5,     9768.8,     3078.5,      15.69
```

```
                  0.55,         20000.0,                     42.0,
        17804.3,        10317.3,           3698.0,          17.5325
                  0.55,         20000.0,                     46.0,
        19661.3,        10813.0,           4347.4,          19.2513
                  0.55,         20000.0,                     48.0,
        20567.3,        11038.3,           4701.1,          20.1286
                  0.55,         20000.0,                     50.0,
        21468.4,        11258.8,           5053.1,          25.1535
                  0.55,         25000.0,                     21.0,
         5638.5,         5226.6,            569.1,           9.4348
                  0.55,         25000.0,                     22.0,
         6102.2,         5443.2,            674.2,           6.9986
                  0.55,         25000.0,                     24.0,
         6985.9,         5832.7,            863.0,           5.1538
                  0.55,         25000.0,                     26.0,
         7825.0,         6177.5,           1059.5,           6.4814
                  0.55,         25000.0,                     30.0,
         9583.8,         6838.0,           1499.7,           8.8813
                  0.55,         25000.0,                     34.0,
        11240.1,         7395.8,           1960.8,          11.5975
                  0.55,         25000.0,                     38.0,
        12829.7,         7887.1,           2435.5,          13.5701
                  0.55,         25000.0,                     42.0,
        14371.4,         8330.6,           2922.0,          15.2974
                  0.55,         25000.0,                     46.0,
        15870.7,         8731.4,           3431.6,          16.9093
                  0.55,         25000.0,                     48.0,
        16601.9,         8913.6,           3708.9,          17.7331
                  0.55,         25000.0,                     50.0,
        17329.5,         9092.0,           3985.6,          18.5183
                  0.55,         30000.0,                     21.0,
         4506.0,         4176.6,            459.6,          12.3646
                  0.55,         30000.0,                     22.0,
         4877.6,         4350.6,            541.3,           9.9465
                  0.55,         30000.0,                     24.0,
         5585.4,         4663.2,            688.6,           6.3459
                  0.55,         30000.0,                     26.0,
         6257.1,         4939.6,            841.1,           4.9951
                  0.55,         30000.0,                     30.0,
         7665.1,         5469.3,           1183.3,           7.2442
```

```
                   0.55,       30000.0,                    34.0,
         8990.5,        5916.4,            1541.7,             9.1534
                   0.55,       30000.0,                    38.0,
        10262.7,        6310.2,            1910.9,            11.4896
                   0.55,       30000.0,                    42.0,
        11496.3,        6665.6,            2289.2,            13.1001
                   0.55,       30000.0,                    46.0,
        12696.1,        6986.9,            2685.7,            14.6049
                   0.55,       30000.0,                    48.0,
        13281.5,        7133.2,            2900.4,            15.3716
                   0.55,       30000.0,                    50.0,
        13863.3,        7275.8,            3114.4,            16.1055
                    0.6,       15000.0,                    21.0,
         9973.8,        9348.2,             950.5,             5.9783
                    0.6,       15000.0,                    22.0,
        10687.7,        9686.8,            1124.3,             7.0163
                    0.6,       15000.0,                    24.0,
        12054.5,       10302.9,            1438.6,             8.6443
                    0.6,       15000.0,                    26.0,
        13357.4,       10855.1,            1754.7,            10.0484
                    0.6,       15000.0,                    30.0,
        16084.8,       11914.4,            2488.5,             13.592
                    0.6,       15000.0,                    34.0,
        18652.7,       12814.1,            3252.5,            16.1377
                    0.6,       15000.0,                    38.0,
        21114.4,       13607.6,            4042.6,            18.3508
                    0.6,       15000.0,                    42.0,
        23501.9,       14327.1,            4845.2,            20.2894
                    0.6,       15000.0,                    46.0,
        25820.8,       14977.7,            5688.9,             30.203
                    0.6,       15000.0,                    48.0,
        26957.4,       15280.2,            6143.4,            44.8334
                    0.6,       15000.0,                    50.0,
        28089.5,       15578.2,            6600.6,            58.0636
                    0.6,       20000.0,                    21.0,
         8116.3,        7607.3,             769.9,             4.8739
                    0.6,       20000.0,                    22.0,
         8697.8,        7883.4,             907.9,             5.6219
                    0.6,       20000.0,                    24.0,
         9810.9,        8385.7,            1157.0,             7.1692
```

```
                    0.6,        20000.0,                  26.0,
        10872.0,         8836.1,           1408.5,            8.504
                    0.6,        20000.0,                  30.0,
        13092.7,         9699.5,           1989.8,          11.7084
                    0.6,        20000.0,                  34.0,
        15183.5,        10432.9,           2596.5,          14.1211
                    0.6,        20000.0,                  38.0,
        17187.7,        11079.9,           3223.0,          16.2145
                    0.6,        20000.0,                  42.0,
        19131.3,        11666.4,           3858.8,          18.0426
                    0.6,        20000.0,                  46.0,
        21019.2,        12196.8,           4528.3,          19.7561
                    0.6,        20000.0,                  48.0,
        21944.6,        12443.4,           4888.4,          21.6513
                    0.6,        20000.0,                  50.0,
        22865.7,        12685.9,           5250.5,          29.7394
                    0.6,        25000.0,                  21.0,
         6548.0,         6137.4,            621.3,           7.8343
                    0.6,        25000.0,                  22.0,
         7018.0,         6361.0,            730.0,           5.5192
                    0.6,        25000.0,                  24.0,
         7917.0,         6767.3,            925.6,           5.6982
                    0.6,        25000.0,                  26.0,
         8773.7,         7131.3,           1123.2,           6.9559
                    0.6,        25000.0,                  30.0,
        10567.1,         7829.7,           1580.2,           9.3365
                    0.6,        25000.0,                  34.0,
        12254.9,         8422.5,           2056.4,            12.09
                    0.6,        25000.0,                  38.0,
        13872.9,         8945.6,           2548.6,          14.0586
                    0.6,        25000.0,                  42.0,
        15441.9,         9419.5,           3048.5,          15.7736
                    0.6,        25000.0,                  46.0,
        16965.8,         9848.6,           3573.0,          17.3772
                    0.6,        25000.0,                  48.0,
        17712.8,        10048.1,           3855.9,          18.1848
                    0.6,        25000.0,                  50.0,
        18456.6,        10244.4,           4140.2,          18.9671
                    0.6,        30000.0,                  21.0,
         5233.4,         4905.1,            500.0,           10.771
```

```
                    0.6,        30000.0,                22.0,
         5610.0,          5084.7,           584.6,           8.4942
                    0.6,        30000.0,                24.0,
         6329.9,          5410.6,           736.4,           5.0619
                    0.6,        30000.0,                26.0,
         7015.7,          5702.4,           890.5,            5.441
                    0.6,        30000.0,                30.0,
         8451.2,          6262.4,          1245.5,           7.6673
                    0.6,        30000.0,                34.0,
         9802.1,          6737.8,          1616.1,           9.6371
                    0.6,        30000.0,                38.0,
        11096.9,          7157.0,          1998.7,          11.9431
                    0.6,        30000.0,                42.0,
        12352.3,          7536.9,          2387.5,           13.542
                    0.6,        30000.0,                46.0,
        13571.6,          7880.7,          2795.9,          15.0403
                    0.6,        30000.0,                48.0,
        14169.7,          8041.0,          3014.6,          15.7897
                    0.6,        30000.0,                50.0,
        14764.6,          8198.1,          3235.5,          16.5238
                    0.6,        35000.0,                21.0,
         4139.6,          3879.7,           401.9,          13.7651
                    0.6,        35000.0,                22.0,
         4438.7,          4022.7,           467.0,          11.4717
                    0.6,        35000.0,                24.0,
         5009.8,          4281.9,           583.8,           8.0866
                    0.6,        35000.0,                26.0,
         5553.5,          4513.7,           702.3,           5.2844
                    0.6,        35000.0,                30.0,
         6691.7,          4958.7,           975.1,           6.0676
                    0.6,        35000.0,                34.0,
         7762.5,          5336.2,          1259.8,           7.8379
                    0.6,        35000.0,                38.0,
         8788.5,          5669.1,          1553.9,           9.6313
                    0.6,        35000.0,                42.0,
         9783.3,          5970.6,          1853.8,          11.4063
                    0.6,        35000.0,                46.0,
        10749.7,          6243.7,          2167.6,          12.7926
                    0.6,        35000.0,                48.0,
        11223.1,          6370.7,          2335.6,          13.4895
```

```
                    0.6,        35000.0,                    50.0,
        11695.0,           6496.0,              2504.7,          14.1725
                    0.6,        37000.0,                    21.0,
         3758.1,           3522.0,               370.0,          14.2083
                    0.6,        37000.0,                    22.0,
         4030.2,           3652.4,               428.7,          11.9308
                    0.6,        37000.0,                    24.0,
         4549.5,           3888.3,               534.1,           8.6067
                    0.6,        37000.0,                    26.0,
         5043.7,           4099.0,               640.8,           5.8649
                    0.6,        37000.0,                    30.0,
         6078.5,           4504.0,               886.9,             5.62
                    0.6,        37000.0,                    34.0,
         7051.8,           4847.5,              1143.9,           7.3337
                    0.6,        37000.0,                    38.0,
         7984.2,           5150.1,              1409.1,           8.8243
                    0.6,        37000.0,                    42.0,
         8888.2,           5424.2,              1680.1,          10.7527
                    0.6,        37000.0,                    46.0,
         9766.3,           5672.7,              1963.0,          12.0908
                    0.6,        37000.0,                    48.0,
        10197.0,           5788.4,              2114.5,          12.7641
                    0.6,        37000.0,                    50.0,
        10625.8,           5902.3,              2266.3,          13.4179
                    0.6,        39000.0,                    21.0,
         3410.5,           3195.9,               342.8,          13.7365
                    0.6,        39000.0,                    22.0,
         3658.1,           3314.8,               396.0,          11.5252
                    0.6,        39000.0,                    24.0,
         4130.3,           3529.5,               491.6,           8.3235
                    0.6,        39000.0,                    26.0,
         4579.5,           3721.3,               588.5,           5.6738
                    0.6,        39000.0,                    30.0,
         5520.0,           4089.6,               811.7,           5.3948
                    0.6,        39000.0,                    34.0,
         6404.7,           4402.1,              1044.8,           7.0416
                    0.6,        39000.0,                    38.0,
         7251.9,           4677.1,              1285.7,           8.4753
                    0.6,        39000.0,                    42.0,
         8073.2,           4926.2,              1532.0,          10.3344
```

```
                    0.6,        39000.0,                  46.0,
          8871.3,         5152.2,           1788.5,          11.6188
                    0.6,        39000.0,                  48.0,
          9262.7,         5257.5,           1925.9,          12.2647
                    0.6,        39000.0,                  50.0,
          9652.5,         5361.2,           2063.4,          12.8882
                    0.6,        41000.0,                  21.0,
          3094.4,         2899.4,            318.6,           13.295
                    0.6,        41000.0,                  22.0,
          3320.0,         3008.0,            366.6,          11.1448
                    0.6,        41000.0,                  24.0,
          3749.4,         3203.6,            453.0,           8.0632
                    0.6,        41000.0,                  26.0,
          4157.8,         3378.0,            541.0,           5.5011
                    0.6,        41000.0,                  30.0,
          5012.8,         3713.2,            743.4,            5.178
                    0.6,        41000.0,                  34.0,
          5816.8,         3997.3,            954.9,           6.7566
                    0.6,        41000.0,                  38.0,
          6586.8,         4247.5,           1173.9,           8.1404
                    0.6,        41000.0,                  42.0,
          7333.1,         4473.9,           1397.5,           9.9298
                    0.6,        41000.0,                  46.0,
          8058.4,         4679.4,           1630.2,          11.1649
                    0.6,        41000.0,                  48.0,
          8414.1,         4775.2,           1754.8,          11.7844
                    0.6,        41000.0,                  50.0,
          8768.5,         4869.7,           1878.9,           12.376
                   0.65,        15000.0,                  21.0,
         11481.3,        10854.1,           1035.9,           6.6394
                   0.65,        15000.0,                  22.0,
         12207.7,        11204.2,           1215.3,           7.6447
                   0.65,        15000.0,                  24.0,
         13603.0,        11846.9,           1542.7,           9.2444
                   0.65,        15000.0,                  26.0,
         14935.6,        12427.0,           1867.4,           11.011
                   0.65,        15000.0,                  30.0,
         17724.6,        13543.7,           2628.1,          14.1919
                   0.65,        15000.0,                  34.0,
         20349.9,        14496.6,           3420.7,          16.7302
```

```
                    0.65,      15000.0,                  38.0,
        22863.3,         15337.4,          4243.8,          18.9494
                    0.65,      15000.0,                  42.0,
        25300.6,         16102.7,          5073.7,          20.8824
                    0.65,      15000.0,                  46.0,
        27684.0,         16813.4,          5945.8,          37.6447
                    0.65,      15000.0,                  48.0,
        28853.2,         17146.3,          6412.4,          52.4446
                    0.65,      15000.0,                  50.0,
        30016.4,         17473.3,          6885.5,           63.165
                    0.65,      20000.0,                  21.0,
         9343.5,          8833.4,           837.7,           5.2653
                    0.65,      20000.0,                  22.0,
         9935.0,          9118.7,           979.9,           6.2209
                    0.65,      20000.0,                  24.0,
        11071.1,          9642.6,          1239.6,             7.74
                    0.65,      20000.0,                  26.0,
        12156.3,         10115.6,          1497.6,           9.0357
                    0.65,      20000.0,                  30.0,
        14427.2,         11026.1,          2101.3,          12.2787
                    0.65,      20000.0,                  34.0,
        16564.3,         11802.7,          2729.9,          14.6793
                    0.65,      20000.0,                  38.0,
        18610.2,         12488.0,          3382.1,          16.7756
                    0.65,      20000.0,                  42.0,
        20594.6,         13112.0,          4040.1,          18.6009
                    0.65,      20000.0,                  46.0,
        22534.2,         13691.1,          4731.2,          20.3058
                    0.65,      20000.0,                  48.0,
        23485.7,         13962.4,          5100.5,          24.9412
                    0.65,      20000.0,                  50.0,
        24432.6,         14229.1,          5474.5,          36.2661
                    0.65,      25000.0,                  21.0,
         7538.7,          7127.2,           674.5,           6.3118
                    0.65,      25000.0,                  22.0,
         8016.8,          7358.3,           786.7,           4.8016
                    0.65,      25000.0,                  24.0,
         8934.3,          7781.9,           990.5,           6.2373
                    0.65,      25000.0,                  26.0,
         9810.5,          8164.3,          1193.3,           7.4577
```

```
                    0.65,       25000.0,                    30.0,         11644.1,         8900.5,          1667.2,          9.8122
                    0.65,       25000.0,                    34.0,         13369.6,         9528.5,          2161.3,         12.6125
                    0.65,       25000.0,                    38.0,         15021.2,        10082.6,          2673.6,          14.582
                    0.65,       25000.0,                    42.0,         16623.1,        10587.1,          3190.9,         16.2937
                    0.65,       25000.0,                    46.0,         18188.8,        11055.3,          3733.0,         17.8916
                    0.65,       25000.0,                    48.0,         18957.1,        11274.9,          4023.5,         18.6867
                    0.65,       25000.0,                    50.0,         19721.6,        11490.7,          4316.5,         19.4646
                    0.65,       30000.0,                    21.0,          6025.8,         5696.8,           541.2,          9.2686
                    0.65,       30000.0,                    22.0,          6408.8,         5882.4,           628.6,          7.1132
                    0.65,       30000.0,                    24.0,          7143.4,         6222.1,           786.8,          4.7609
                    0.65,       30000.0,                    26.0,          7844.8,         6528.7,           944.9,          5.9115
                    0.65,       30000.0,                    30.0,          9312.7,         7119.2,          1313.5,           8.114
                    0.65,       30000.0,                    34.0,         10693.4,         7622.5,          1697.7,         10.5176
                    0.65,       30000.0,                    38.0,         12015.1,         8066.7,          2096.0,          12.429
                    0.65,       30000.0,                    42.0,         13296.7,         8470.9,          2498.2,         14.0245
                    0.65,       30000.0,                    46.0,         14549.5,         8846.3,          2919.9,         15.5145
                    0.65,       30000.0,                    48.0,         15164.4,         9022.5,          3145.2,         16.2578
                    0.65,       30000.0,                    50.0,         15776.0,         9195.4,          3372.5,         16.9835
                    0.65,       35000.0,                    21.0,          4767.0,         4506.6,           433.5,         12.2477
                    0.65,       35000.0,                    22.0,          5071.1,         4654.4,           500.8,         10.0982
```

```
    0.65,    35000.0,        24.0,    5653.7,    4924.4,    622.1,    6.8639
    0.65,    35000.0,        26.0,    6210.0,    5168.0,    743.9,    4.422
    0.65,    35000.0,        30.0,    7373.7,    5637.1,    1027.3,    6.4841
    0.65,    35000.0,        34.0,    8468.2,    6037.1,    1322.7,    8.2441
    0.65,    35000.0,        38.0,    9515.4,    6389.7,    1628.5,    10.3632
    0.65,    35000.0,        42.0,    10530.9,    6710.5,    1938.8,    11.8544
    0.65,    35000.0,        46.0,    11523.6,    7008.6,    2262.9,    13.2342
    0.65,    35000.0,        48.0,    12010.7,    7148.5,    2436.1,    13.9247
    0.65,    35000.0,        50.0,    12495.8,    7286.2,    2609.9,    14.5969
    0.65,    37000.0,        21.0,    4328.0,    4091.4,    398.3,    12.7114
    0.65,    37000.0,        22.0,    4604.7,    4226.1,    459.1,    10.5799
    0.65,    37000.0,        24.0,    5134.7,    4472.0,    568.6,    7.4067
    0.65,    37000.0,        26.0,    5640.3,    4693.6,    678.3,    4.7981
    0.65,    37000.0,        30.0,    6698.2,    5120.4,    934.1,    6.0241
    0.65,    37000.0,        34.0,    7693.1,    5484.2,    1200.6,    7.7263
    0.65,    37000.0,        38.0,    8645.0,    5805.0,    1476.6,    9.5229
    0.65,    37000.0,        42.0,    9567.7,    6096.5,    1757.2,    11.1876
    0.65,    37000.0,        46.0,    10470.1,    6367.8,    2049.0,    12.5173
    0.65,    37000.0,        48.0,    10912.6,    6494.8,    2205.3,    13.1845
    0.65,    37000.0,        50.0,    11353.7,    6620.3,    2361.7,    13.8301
```

```
                   0.65,      39000.0,                 21.0,          3928.0,        3713.0,          368.4,       12.2906
                   0.65,      39000.0,                 22.0,          4179.8,        3835.7,          423.5,       10.2222
                   0.65,      39000.0,                 24.0,          4661.5,        4059.5,          522.5,        7.1754
                   0.65,      39000.0,                 26.0,          5121.2,        4261.1,          622.3,        4.6517
                   0.65,      39000.0,                 30.0,          6082.8,        4649.3,          854.5,        5.7838
                   0.65,      39000.0,                 34.0,          6987.2,        4980.3,         1096.2,        7.4195
                   0.65,      39000.0,                 38.0,          7852.3,        5272.0,         1347.0,        9.1325
                   0.65,      39000.0,                 42.0,          8690.6,        5536.9,         1601.9,       10.7517
                   0.65,      39000.0,                 46.0,          9510.4,        5783.5,         1866.9,       12.0307
                   0.65,      39000.0,                 48.0,          9912.8,        5899.1,         2008.5,       12.6712
                   0.65,      39000.0,                 50.0,         10313.9,        6013.4,         2149.9,        13.285
                   0.65,      41000.0,                 21.0,          3564.4,        3369.0,          341.4,       11.8988
                   0.65,      41000.0,                 22.0,          3793.7,        3481.1,          391.2,        9.8837
                   0.65,      41000.0,                 24.0,          4231.9,        3684.9,          480.9,        6.9499
                   0.65,      41000.0,                 26.0,          4649.7,        3868.3,          571.5,        4.5148
                   0.65,      41000.0,                 30.0,          5523.7,        4221.4,          782.2,        5.5521
                   0.65,      41000.0,                 34.0,          6345.8,        4522.5,         1001.6,        7.1247
                   0.65,      41000.0,                 38.0,          7131.8,        4787.6,         1229.2,        8.7518
                   0.65,      41000.0,                 42.0,          7893.5,        5028.4,         1460.7,       10.3306
                   0.65,      41000.0,                 46.0,          8638.7,        5252.7,         1701.3,       11.5623
```

```
                     0.65,      41000.0,                 48.0,
         9004.5,          5357.9,          1829.8,          12.177
                     0.65,      41000.0,                 50.0,
         9369.0,          5462.0,          1957.7,          12.7624
                      0.7,      15000.0,                 21.0,
        13112.0,         12479.9,          1130.1,           7.3244
                      0.7,      15000.0,                 22.0,
        13855.4,         12844.0,          1307.0,           8.2542
                      0.7,      15000.0,                 24.0,
        15285.0,         13515.0,          1648.6,           9.8351
                      0.7,      15000.0,                 26.0,
        16649.3,         14120.8,          1994.3,          11.9014
                      0.7,      15000.0,                 30.0,
        19512.9,         15298.6,          2780.9,          14.827
                      0.7,      15000.0,                 34.0,
        22204.6,         16304.7,          3606.0,          17.3622
                      0.7,      15000.0,                 38.0,
        24778.2,         17192.6,          4465.4,          19.5875
                      0.7,      15000.0,                 42.0,
        27296.1,         18024.8,          5329.0,          23.4826
                      0.7,      15000.0,                 46.0,
        29762.1,         18805.2,          6235.3,          46.8096
                      0.7,      15000.0,                 48.0,
        30970.5,         19170.8,          6719.8,          59.383
                      0.7,      15000.0,                 50.0,
        32171.1,         19528.5,          7209.9,          68.7032
                      0.7,      20000.0,                 21.0,
        10671.0,         10156.9,           912.4,           5.9185
                      0.7,      20000.0,                 22.0,
        11276.8,         10454.1,          1053.1,           6.8056
                      0.7,      20000.0,                 24.0,
        12440.5,         11000.8,          1323.8,           8.3011
                      0.7,      20000.0,                 26.0,
        13551.0,         11494.4,          1598.0,           9.6082
                      0.7,      20000.0,                 30.0,
        15882.3,         12454.6,          2221.7,          12.8741
                      0.7,      20000.0,                 34.0,
        18073.4,         13274.7,          2876.5,          15.2715
                      0.7,      20000.0,                 38.0,
        20168.3,         13998.5,          3557.9,          17.3754
```

```
                    0.7,        20000.0,                42.0,
        22217.1,         14676.3,          4242.1,          19.2013
                    0.7,        20000.0,                46.0,
        24224.9,         15312.9,          4960.5,          22.3983
                    0.7,        20000.0,                48.0,
        25208.3,         15610.7,          5344.4,          30.8596
                    0.7,        20000.0,                50.0,
        26185.8,         15902.7,          5731.7,          44.4788
                    0.7,        25000.0,                21.0,
         8610.5,          8195.8,           733.3,           4.7574
                    0.7,        25000.0,                22.0,
         9099.7,          8436.2,           843.9,           5.3542
                    0.7,        25000.0,                24.0,
        10039.4,          8878.3,          1056.5,           6.7662
                    0.7,        25000.0,                26.0,
        10936.2,          9277.4,          1272.2,           7.9972
                    0.7,        25000.0,                30.0,
        12818.5,         10053.9,          1762.2,          10.8094
                    0.7,        25000.0,                34.0,
        14587.3,         10716.9,          2276.5,          13.1665
                    0.7,        25000.0,                38.0,
        16278.4,         11302.1,          2811.5,          15.1405
                    0.7,        25000.0,                42.0,
        17932.2,         11850.1,          3349.2,          16.8532
                    0.7,        25000.0,                46.0,
        19552.5,         12364.7,          3912.9,          18.4489
                    0.7,        25000.0,                48.0,
        20346.0,         12605.2,          4213.9,          19.2419
                    0.7,        25000.0,                50.0,
        21135.9,         12842.2,          4516.7,          20.8046
                    0.7,        30000.0,                21.0,
         6883.1,          6551.6,           586.9,           7.7444
                    0.7,        30000.0,                22.0,
         7275.0,          6744.6,           673.2,           5.7924
                    0.7,        30000.0,                24.0,
         8027.3,          7099.1,           838.4,           5.2616
                    0.7,        30000.0,                26.0,
         8744.9,          7418.9,          1005.9,           6.4147
                    0.7,        30000.0,                30.0,
        10251.7,          8041.7,          1387.1,           8.5806
```

```
                    0.7,      30000.0,                 34.0,
         11667.2,        8573.2,        1787.1,        11.0992
                    0.7,      30000.0,                 38.0,
         13020.3,        9042.3,        2203.0,        12.9445
                    0.7,      30000.0,                 42.0,
         14343.5,        9481.5,        2621.7,        14.5449
                    0.7,      30000.0,                 46.0,
         15639.9,        9894.0,        3059.8,        16.0318
                    0.7,      30000.0,                 48.0,
         16274.9,       10087.0,        3293.1,        16.7705
                    0.7,      30000.0,                 50.0,
         16906.9,       10277.0,        3528.3,        17.4883
                    0.7,      35000.0,                 21.0,
          5445.9,        5183.5,         468.4,        10.7259
                    0.7,      35000.0,                 22.0,
          5757.0,        5337.1,         535.0,         8.7976
                    0.7,      35000.0,                 24.0,
          6353.6,        5618.9,         661.7,         5.6702
                    0.7,      35000.0,                 26.0,
          6922.8,        5873.2,         791.0,         4.8981
                    0.7,      35000.0,                 30.0,
          8117.1,        6367.8,        1083.9,         6.9194
                    0.7,      35000.0,                 34.0,
          9239.1,        6790.1,        1391.6,         8.6731
                    0.7,      35000.0,                 38.0,
         10311.5,        7162.7,        1711.2,        10.8441
                    0.7,      35000.0,                 42.0,
         11359.5,        7511.0,        2033.2,        12.3341
                    0.7,      35000.0,                 46.0,
         12386.9,        7838.7,        2370.3,        13.7136
                    0.7,      35000.0,                 48.0,
         12890.1,        7992.0,        2549.7,        14.4009
                    0.7,      35000.0,                 50.0,
         13391.1,        8143.2,        2729.2,        15.0634
                    0.7,      37000.0,                 21.0,
          4944.6,        4706.2,         429.8,        11.2106
                    0.7,      37000.0,                 22.0,
          5227.7,        4846.2,         489.9,         9.3025
                    0.7,      37000.0,                 24.0,
          5770.2,        5102.7,         604.1,         6.2444
```

```
                0.7,     37000.0,               26.0,    6287.7,     5334.1,      720.9,     4.4868
                0.7,     37000.0,               30.0,    7373.4,     5784.0,      985.0,     6.4444
                0.7,     37000.0,               34.0,    8393.3,     6168.3,     1262.6,     8.1407
                0.7,     37000.0,               38.0,    9368.0,     6507.2,     1551.0,    10.2088
                0.7,     37000.0,               42.0,   10320.4,     6823.9,     1842.1,    11.6506
                0.7,     37000.0,               46.0,   11254.0,     7121.8,     2146.1,    12.9821
                0.7,     37000.0,               48.0,   11711.4,     7261.3,     2308.0,     13.645
                0.7,     37000.0,               50.0,   12167.0,     7399.1,     2469.7,     14.282
                0.7,     39000.0,               21.0,    4487.8,     4271.2,      396.9,    10.8462
                0.7,     39000.0,               22.0,    4745.4,     4398.8,      451.5,     8.9937
                0.7,     39000.0,               24.0,    5238.7,     4632.3,      555.0,     6.0497
                0.7,     39000.0,               26.0,    5709.1,     4842.7,      660.9,     4.3014
                0.7,     39000.0,               30.0,    6696.0,     5252.1,      900.8,     6.1896
                0.7,     39000.0,               34.0,    7623.1,     5601.6,     1152.7,     7.8201
                0.7,     39000.0,               38.0,    8508.9,     5909.8,     1414.4,     9.8095
                0.7,     39000.0,               42.0,    9374.1,     6197.4,     1678.7,     11.196
                0.7,     39000.0,               46.0,   10222.7,     6468.4,     1955.0,    12.4794
                0.7,     39000.0,               48.0,   10638.4,     6595.3,     2101.9,    13.1156
                0.7,     39000.0,               50.0,   11052.6,     6720.7,     2248.2,    13.7233
                0.7,     41000.0,               21.0,    4072.9,     3876.1,      367.0,    10.4999
```

```
                    0.7,        41000.0,                    22.0,
          4307.3,          3992.4,            416.6,             8.698
                    0.7,        41000.0,                    24.0,
          4756.0,          4204.9,            510.3,            5.8669
                    0.7,        41000.0,                    26.0,
          5183.6,          4396.4,            606.4,            4.1214
                    0.7,        41000.0,                    30.0,
          6080.8,          4768.8,            824.2,            5.9412
                    0.7,        41000.0,                    34.0,
          6923.7,          5086.9,           1052.9,            7.5111
                    0.7,        41000.0,                    38.0,
          7728.7,          5367.1,           1290.6,            9.4241
                    0.7,        41000.0,                    42.0,
          8515.0,          5628.6,           1531.1,           10.7626
                    0.7,        41000.0,                    46.0,
          9286.2,          5874.9,           1781.7,           11.9963
                    0.7,        41000.0,                    48.0,
          9664.1,          5990.4,           1914.9,           12.6068
                    0.7,        41000.0,                    50.0,
         10040.5,          6104.4,           2047.5,           13.1893
                    0.7,        43000.0,                    21.0,
          3695.7,          3516.9,            340.2,            10.182
                    0.7,        43000.0,                    22.0,
          3909.2,          3623.1,            385.1,            8.4342
                    0.7,        43000.0,                    24.0,
          4317.5,          3816.8,            469.8,            5.6921
                    0.7,        43000.0,                    26.0,
          4706.4,          3991.1,            557.2,            3.9488
                    0.7,        43000.0,                    30.0,
          5522.1,          4329.9,            754.8,             5.704
                    0.7,        43000.0,                    34.0,
          6288.3,          4619.3,            962.3,            7.2124
                    0.7,        43000.0,                    38.0,
          7019.9,          4874.0,           1178.2,            9.0518
                    0.7,        43000.0,                    42.0,
          7734.6,          5111.8,           1396.8,           10.3435
                    0.7,        43000.0,                    46.0,
          8435.5,          5335.9,           1624.2,           11.5304
                    0.7,        43000.0,                    48.0,
          8778.7,          5440.7,           1745.0,           12.1165
```

```
                    0.7,       43000.0,                  50.0,
          9121.2,        5544.6,           1865.4,           12.6763
                   0.75,       20000.0,                  21.0,
         12099.2,       11578.4,            993.4,            6.5843
                   0.75,       20000.0,                  22.0,
         12724.0,       11890.6,           1129.3,            7.3857
                   0.75,       20000.0,                  24.0,
         13920.6,       12462.1,           1410.5,            8.8599
                   0.75,       20000.0,                  26.0,
         15061.0,       12977.4,           1701.9,           10.3273
                   0.75,       20000.0,                  30.0,
         17460.5,       13988.0,           2352.5,           13.4976
                   0.75,       20000.0,                  34.0,
         19714.1,       14852.5,           3037.5,           15.8999
                   0.75,       20000.0,                  38.0,
         21884.6,       15634.1,           3751.2,           18.0102
                   0.75,       20000.0,                  42.0,
         24018.9,       16379.3,           4468.2,           19.8495
                   0.75,       20000.0,                  46.0,
         26105.5,       17076.8,           5219.0,           27.2502
                   0.75,       20000.0,                  48.0,
         27126.2,       17402.8,           5619.3,           39.5279
                   0.75,       20000.0,                  50.0,
         28140.0,       17722.3,           6024.0,           53.0099
                   0.75,       25000.0,                  21.0,
          9763.6,        9343.5,            797.1,            5.1492
                   0.75,       25000.0,                  22.0,
         10267.9,        9595.8,            903.1,            5.8999
                   0.75,       25000.0,                  24.0,
         11234.1,       10058.0,           1124.6,            7.2923
                   0.75,       25000.0,                  26.0,
         12154.7,       10474.5,           1353.8,            8.5338
                   0.75,       25000.0,                  30.0,
         14092.2,       11291.8,           1864.8,           11.4987
                   0.75,       25000.0,                  34.0,
         15911.3,       11990.8,           2403.1,           13.7532
                   0.75,       25000.0,                  38.0,
         17663.3,       12622.6,           2963.2,            15.732
                   0.75,       25000.0,                  42.0,
         19385.8,       13225.0,           3526.7,           17.4568
```

```
                    0.75,       25000.0,                  46.0,
        21070.1,         13789.1,          4116.3,          19.0568
                    0.75,       25000.0,                  48.0,
        21894.0,         14052.8,          4429.6,          19.8471
                    0.75,       25000.0,                  50.0,
        22712.7,         14311.6,          4746.3,          24.5603
                    0.75,       30000.0,                  21.0,
         7805.4,          7469.7,           636.6,           6.2175
                    0.75,       30000.0,                  22.0,
         8209.2,          7672.0,           718.6,            4.557
                    0.75,       30000.0,                  24.0,
         8982.7,          8042.7,           891.3,           5.7565
                    0.75,       30000.0,                  26.0,
         9719.7,          8376.8,          1069.8,           6.9217
                    0.75,       30000.0,                  30.0,
        11270.2,          9032.1,          1467.1,           9.0687
                    0.75,       30000.0,                  34.0,
        12725.6,          9592.3,          1885.5,          11.6454
                    0.75,       30000.0,                  38.0,
        14127.3,         10098.7,          2320.9,          13.4935
                    0.75,       30000.0,                  42.0,
        15505.2,         10581.3,          2759.3,          15.1024
                    0.75,       30000.0,                  46.0,
        16852.7,         11033.5,          3217.1,          16.5918
                    0.75,       30000.0,                  48.0,
        17511.6,         11244.9,          3461.0,           17.331
                    0.75,       30000.0,                  50.0,
        18167.3,         11452.9,          3705.8,          18.0435
                    0.75,       35000.0,                  21.0,
         6176.3,          5910.6,           506.6,           9.2124
                    0.75,       35000.0,                  22.0,
         6496.8,          6071.7,           569.8,           7.5518
                    0.75,       35000.0,                  24.0,
         7110.0,          6366.1,           702.3,           4.5131
                    0.75,       35000.0,                  26.0,
         7694.4,          6631.6,           839.8,           5.3704
                    0.75,       35000.0,                  30.0,
         8923.6,          7152.1,          1145.3,           7.3736
                    0.75,       35000.0,                  34.0,
        10077.2,          7597.2,          1467.2,           9.1266
```

```
                  0.75,      35000.0,                38.0,
        11187.9,      7999.4,          1801.8,      11.3524
                  0.75,      35000.0,                42.0,
        12279.4,      8382.4,          2139.1,      12.8488
                  0.75,      35000.0,                46.0,
        13347.0,      8741.4,          2491.2,      14.2339
                  0.75,      35000.0,                48.0,
        13869.4,      8909.5,          2678.5,      14.9197
                  0.75,      35000.0,                50.0,
        14389.1,      9075.0,          2865.5,      15.5782
                  0.75,      37000.0,                21.0,
         5608.1,      5366.7,           464.3,       9.7221
                  0.75,      37000.0,                22.0,
         5899.7,      5513.5,           521.3,       8.0824
                  0.75,      37000.0,                24.0,
         6457.4,      5781.4,           640.8,       5.1113
                  0.75,      37000.0,                26.0,
         6988.6,      6023.0,           764.9,       4.9447
                  0.75,      37000.0,                30.0,
         8106.0,      6496.6,          1040.5,       6.8848
                  0.75,      37000.0,                34.0,
         9154.7,      6901.6,          1331.0,         8.58
                  0.75,      37000.0,                38.0,
        10164.2,      7267.4,          1632.9,      10.6998
                  0.75,      37000.0,                42.0,
        11156.1,      7615.6,          1937.4,      12.1486
                  0.75,      37000.0,                46.0,
        12126.5,      7942.2,          2255.5,      13.4852
                  0.75,      37000.0,                48.0,
        12601.0,      8094.8,          2424.1,      14.1465
                  0.75,      37000.0,                50.0,
        13073.6,      8245.6,          2592.2,      14.7777
                  0.75,      39000.0,                21.0,
         5090.3,      4871.0,           428.1,        9.413
                  0.75,      39000.0,                22.0,
         5355.6,      5004.7,           479.9,       7.8205
                  0.75,      39000.0,                24.0,
         5862.7,      5248.6,           588.2,        4.959
                  0.75,      39000.0,                26.0,
         6345.7,      5468.4,           700.9,       4.7432
```

```
                    0.75,      39000.0,                30.0,
         7361.5,          5899.2,          951.1,          6.612
                    0.75,      39000.0,                34.0,
         8314.7,          6267.5,         1214.7,          8.2426
                    0.75,      39000.0,                38.0,
         9232.3,          6600.2,         1488.9,         10.2835
                    0.75,      39000.0,                42.0,
        10133.6,          6916.7,         1765.7,         11.6794
                    0.75,      39000.0,                46.0,
        11015.2,          7213.4,         2054.3,         12.9644
                    0.75,      39000.0,                48.0,
        11446.7,          7352.4,         2207.5,         13.5999
                    0.75,      39000.0,                50.0,
        11876.3,          7489.5,         2360.1,         14.2058
                    0.75,      41000.0,                21.0,
         4620.0,          4420.7,          395.3,          9.1163
                    0.75,      41000.0,                22.0,
         4861.4,          4542.5,          442.3,          7.5735
                    0.75,      41000.0,                24.0,
         5322.7,          4764.6,          540.6,          4.8099
                    0.75,      41000.0,                26.0,
         5761.8,          4964.6,          642.9,          4.5487
                    0.75,      41000.0,                30.0,
         6685.2,          5356.6,          869.9,          6.3495
                    0.75,      41000.0,                34.0,
         7551.8,          5691.7,         1109.2,          7.9189
                    0.75,      41000.0,                38.0,
         8385.8,          5994.3,         1358.1,          9.8824
                    0.75,      41000.0,                42.0,
         9204.8,          6281.8,         1609.8,         11.2269
                    0.75,      41000.0,                46.0,
        10006.1,          6551.7,         1872.0,         12.4638
                    0.75,      41000.0,                48.0,
        10398.3,          6678.1,         2010.9,         13.0741
                    0.75,      41000.0,                50.0,
        10788.7,          6802.8,         2149.4,         13.6567
                    0.75,      43000.0,                21.0,
         4192.7,          4011.6,          365.5,          8.8388
                    0.75,      43000.0,                22.0,
         4412.3,          4122.6,          408.2,          7.3458
```

```
                    0.75,         43000.0,                    24.0,
          4832.0,         4324.9,           497.1,          4.6778
                    0.75,         43000.0,                    26.0,
          5231.3,         4506.9,           590.1,          4.3594
                    0.75,         43000.0,                    30.0,
          6071.0,         4863.7,           796.3,          6.0969
                    0.75,         43000.0,                    34.0,
          6858.8,         5168.6,          1013.4,           7.606
                    0.75,         43000.0,                    38.0,
          7616.8,         5443.7,          1239.5,          9.4944
                    0.75,         43000.0,                    42.0,
          8361.0,         5705.0,          1468.3,         10.7912
                    0.75,         43000.0,                    46.0,
          9089.4,         5950.4,          1706.1,         11.9799
                    0.75,         43000.0,                    48.0,
          9445.7,         6065.4,          1832.3,         12.5674
                    0.75,         43000.0,                    50.0,
          9800.8,         6179.0,          1958.1,         13.1272
                    0.79,         20000.0,                    21.0,
         13316.7,        12788.5,          1056.2,           7.085
                    0.79,         20000.0,                    22.0,
         13958.5,        13113.3,          1191.3,          7.8437
                    0.79,         20000.0,                    24.0,
         15183.9,        13704.9,          1486.2,          9.3251
                    0.79,         20000.0,                    26.0,
         16353.4,        14240.4,          1788.3,         11.2602
                    0.79,         20000.0,                    30.0,
         18814.0,        15292.3,          2465.9,         14.0199
                    0.79,         20000.0,                    34.0,
         21136.0,        16205.7,          3176.5,         16.4247
                    0.79,         20000.0,                    38.0,
         23387.7,        17048.6,          3921.0,         18.5484
                    0.79,         20000.0,                    42.0,
         25599.2,        17851.6,          4668.4,         20.4014
                    0.79,         20000.0,                    46.0,
         27757.9,        18601.6,          5449.7,         33.7104
                    0.79,         20000.0,                    48.0,
         28812.4,        18951.7,          5865.5,         47.6231
                    0.79,         20000.0,                    50.0,
         29859.3,        19294.4,          6286.5,         58.7708
```

```
                    0.79,       25000.0,                 21.0,
         10746.2,         10320.3,            846.3,            5.6217
                    0.79,       25000.0,                 22.0,
         11264.8,         10583.2,            952.1,            6.3348
                    0.79,       25000.0,                 24.0,
         12254.3,         11061.5,           1184.1,              7.73
                    0.79,       25000.0,                 26.0,
         13198.3,         11494.4,           1421.7,            8.9669
                    0.79,       25000.0,                 30.0,
         15184.9,         12345.0,           1954.0,           11.9882
                    0.79,       25000.0,                 34.0,
         17059.4,         13083.5,           2512.5,           14.2443
                    0.79,       25000.0,                 38.0,
         18877.2,         13765.2,           3097.0,           16.2353
                    0.79,       25000.0,                 42.0,
         20662.0,         14414.1,           3683.6,           17.9702
                    0.79,       25000.0,                 46.0,
         22404.6,         15020.7,           4297.5,           19.5778
                    0.79,       25000.0,                 48.0,
         23255.8,         15303.8,           4623.5,           21.9802
                    0.79,       25000.0,                 50.0,
         24101.0,         15581.1,           4952.6,           29.7794
                    0.79,       30000.0,                 21.0,
          8591.6,          8251.2,            675.0,            5.0944
                    0.79,       30000.0,                 22.0,
          9006.5,          8461.9,            756.7,            4.8532
                    0.79,       30000.0,                 24.0,
          9798.6,          8845.4,            937.1,            6.1642
                    0.79,       30000.0,                 26.0,
         10554.1,          9192.5,           1122.5,            7.3259
                    0.79,       30000.0,                 30.0,
         12143.7,          9874.3,           1536.0,            9.4741
                    0.79,       30000.0,                 34.0,
         13643.5,         10466.3,           1970.6,           12.1021
                    0.79,       30000.0,                 38.0,
         15097.6,         11012.7,           2424.8,           13.9599
                    0.79,       30000.0,                 42.0,
         16525.2,         11532.5,           2881.4,           15.5783
                    0.79,       30000.0,                 46.0,
         17919.4,         12018.9,           3357.8,           17.0748
```

```
    0.79,    30000.0,        48.0,
18600.1,   12245.8,    3611.2,    17.8146
    0.79,    30000.0,        50.0,
19277.1,   12468.8,    3865.8,    18.5261
    0.79,    35000.0,        21.0,
 6798.9,    6529.5,     536.1,      8.112
    0.79,    35000.0,        22.0,
 7128.1,    6697.1,     598.7,     6.5792
    0.79,    35000.0,        24.0,
 7756.4,    7002.1,     737.5,     4.6585
    0.79,    35000.0,        26.0,
 8355.2,    7277.7,     880.4,     5.7496
    0.79,    35000.0,        30.0,
 9615.4,    7819.4,    1198.4,     7.7528
    0.79,    35000.0,        34.0,
10803.8,    8289.5,    1532.7,      9.805
    0.79,    35000.0,        38.0,
11956.1,    8723.4,    1881.7,    11.7838
    0.79,    35000.0,        42.0,
13087.1,    9135.9,    2232.9,    13.2881
    0.79,    35000.0,        46.0,
14191.7,    9522.2,    2599.6,     14.681
    0.79,    35000.0,        48.0,
14731.2,    9702.5,    2793.6,    15.3669
    0.79,    35000.0,        50.0,
15267.9,    9880.0,    2987.8,    16.0243
    0.79,    37000.0,        21.0,
 6173.6,    5928.9,     490.9,     8.6418
    0.79,    37000.0,        22.0,
 6473.1,    6081.5,     547.1,     7.1362
    0.79,    37000.0,        24.0,
 7044.4,    6359.1,     672.3,     4.2757
    0.79,    37000.0,        26.0,
 7588.8,    6609.9,     801.5,     5.3123
    0.79,    37000.0,        30.0,
 8734.3,    7102.7,    1088.2,     7.2492
    0.79,    37000.0,        34.0,
 9814.5,    7530.3,    1389.7,     8.9444
    0.79,    37000.0,        38.0,
10862.0,    7925.1,    1704.9,    11.1167
```

```
                   0.79,       37000.0,                  42.0,
         11889.7,        8300.2,           2022.2,           12.5731
                   0.79,       37000.0,                  46.0,
         12893.6,        8651.4,           2352.9,           13.9174
                   0.79,       37000.0,                  48.0,
         13383.9,        8815.5,           2528.3,           14.5794
                   0.79,       37000.0,                  50.0,
         13871.8,        8977.0,           2703.4,           15.2136
                   0.79,       39000.0,                  21.0,
          5603.8,        5381.5,            452.2,            8.3724
                   0.79,       39000.0,                  22.0,
          5876.3,        5520.5,            503.1,            6.9121
                   0.79,       39000.0,                  24.0,
          6395.9,        5773.3,            616.7,            4.1333
                   0.79,       39000.0,                  26.0,
          6890.8,        6001.3,            734.2,            5.0969
                   0.79,       39000.0,                  30.0,
          7932.0,        6449.6,            994.3,             6.962
                   0.79,       39000.0,                  34.0,
          8914.2,        6838.7,           1268.4,            8.5965
                   0.79,       39000.0,                  38.0,
          9866.0,        7197.6,           1554.3,            10.685
                   0.79,       39000.0,                  42.0,
         10799.9,        7538.5,           1842.4,           12.0879
                   0.79,       39000.0,                  46.0,
         11712.2,        7857.8,           2143.1,           13.3818
                   0.79,       39000.0,                  48.0,
         12157.8,        8007.0,           2302.2,           14.0174
                   0.79,       39000.0,                  50.0,
         12601.3,        8153.9,           2461.0,           14.6254
                   0.79,       41000.0,                  21.0,
          5086.2,        4884.2,            417.1,             8.114
                   0.79,       41000.0,                  22.0,
          5334.1,        5010.8,            463.4,            6.7051
                   0.79,       41000.0,                  24.0,
          5806.8,        5241.1,            566.2,            3.9985
                   0.79,       41000.0,                  26.0,
          6256.7,        5448.5,            672.9,            4.8879
                   0.79,       41000.0,                  30.0,
          7203.5,        5856.5,            909.4,            6.6888
```

```
                    0.79,       41000.0,                    34.0,
          8096.2,          6210.4,           1158.0,             8.2594
                    0.79,       41000.0,                    38.0,
          8961.4,          6536.8,           1417.5,            10.2694
                    0.79,       41000.0,                    42.0,
          9810.0,          6846.6,           1679.3,             11.621
                    0.79,       41000.0,                    46.0,
         10639.2,          7136.9,           1952.5,            12.8649
                    0.79,       41000.0,                    48.0,
         11044.2,          7272.6,           2096.9,            13.4762
                    0.79,       41000.0,                    50.0,
         11447.3,          7406.3,           2241.1,            14.0615
                     0.8,       20000.0,                    21.0,
         13631.6,         13101.2,           1071.7,             7.2066
                     0.8,       20000.0,                    22.0,
         14277.9,         13429.2,           1206.9,             7.9564
                     0.8,       20000.0,                    24.0,
         15511.1,         14025.9,           1505.6,             9.4413
                     0.8,       20000.0,                    26.0,
         16688.8,         14567.1,           1810.3,            11.3981
                     0.8,       20000.0,                    30.0,
         19165.5,         15629.4,           2495.5,            14.1535
                     0.8,       20000.0,                    34.0,
         21509.9,         16559.4,           3213.1,            16.5599
                     0.8,       20000.0,                    38.0,
         23783.7,         17418.7,           3966.3,            18.6886
                     0.8,       20000.0,                    42.0,
         26016.0,         18236.5,           4721.8,            20.5452
                     0.8,       20000.0,                    46.0,
         28194.2,         19000.4,           5512.1,            35.7201
                     0.8,       20000.0,                    48.0,
         29257.3,         19356.5,           5932.0,            49.6781
                     0.8,       20000.0,                    50.0,
         30313.1,         19704.9,           6356.8,            59.9573
                     0.8,       25000.0,                    21.0,
         11000.5,         10572.8,            858.4,             5.7369
                     0.8,       25000.0,                    22.0,
         11522.8,         10838.4,            964.5,             6.4424
                     0.8,       25000.0,                    24.0,
         12518.5,         11320.9,           1199.4,             7.8407
```

```
                    0.8,        25000.0,                    26.0,
        13468.9,        11757.9,            1438.7,             9.0737
                    0.8,        25000.0,                    30.0,
        15468.6,        12617.0,            1976.9,            12.1112
                    0.8,        25000.0,                    34.0,
        17361.2,        13369.0,            2541.0,            14.3696
                    0.8,        25000.0,                    38.0,
        19196.7,        14063.9,            3132.5,            16.3657
                    0.8,        25000.0,                    42.0,
        20998.4,        14725.0,            3725.7,            18.1043
                    0.8,        25000.0,                    46.0,
        22756.7,        15342.6,            4346.4,            19.7156
                    0.8,        25000.0,                    48.0,
        23614.9,        15630.5,            4675.4,            22.8123
                    0.8,        25000.0,                    50.0,
        24467.2,        15912.5,            5007.9,            31.4463
                    0.8,        30000.0,                    21.0,
         8794.9,         8453.1,             684.4,             4.8251
                    0.8,        30000.0,                    22.0,
         9212.8,         8666.0,             766.2,             4.9546
                    0.8,        30000.0,                    24.0,
        10009.9,         9052.9,             949.1,             6.2683
                    0.8,        30000.0,                    26.0,
        10770.5,         9403.4,            1135.9,             7.4268
                    0.8,        30000.0,                    30.0,
        12370.4,        10092.0,            1554.2,             9.6026
                    0.8,        30000.0,                    34.0,
        13884.6,        10694.7,            1993.0,            12.2194
                    0.8,        30000.0,                    38.0,
        15352.8,        11251.5,            2452.3,            14.0807
                    0.8,        30000.0,                    42.0,
        16793.7,        11781.1,            2913.8,            15.7013
                    0.8,        30000.0,                    46.0,
        18200.3,        12276.3,            3395.6,            17.2013
                    0.8,        30000.0,                    48.0,
        18886.8,        12507.0,            3651.1,            17.9402
                    0.8,        30000.0,                    50.0,
        19569.5,        12734.0,            3908.4,            18.6521
                    0.8,        35000.0,                    21.0,
         6959.9,         6689.4,             543.4,             7.8502
```

```
                    0.8,      35000.0,                22.0,
     7291.5,          6858.8,           606.0,           6.339
                    0.8,      35000.0,                24.0,
     7923.7,          7166.4,           746.6,          4.7559
                    0.8,      35000.0,                26.0,
     8526.5,          7444.7,           890.7,          5.8442
                    0.8,      35000.0,                30.0,
     9794.9,          7991.8,          1212.3,          7.8493
                    0.8,      35000.0,                34.0,
    10994.6,          8470.3,          1549.7,          9.9967
                    0.8,      35000.0,                38.0,
    12158.1,          8912.6,          1902.7,         11.8951
                    0.8,      35000.0,                42.0,
    13299.6,          9332.9,          2257.9,         13.4031
                    0.8,      35000.0,                46.0,
    14413.9,          9725.9,          2628.2,         14.7963
                    0.8,      35000.0,                48.0,
    14958.1,          9909.5,          2824.3,          15.483
                    0.8,      35000.0,                50.0,
    15499.3,         10090.1,          3020.9,         16.1418
                    0.8,      37000.0,                21.0,
     6319.9,          6074.2,           497.4,          8.3849
                    0.8,      37000.0,                22.0,
     6621.5,          6228.4,           553.7,          6.9012
                    0.8,      37000.0,                24.0,
     7196.4,          6508.4,           680.6,          4.3469
                    0.8,      37000.0,                26.0,
     7744.3,          6761.5,           810.7,          5.4024
                    0.8,      37000.0,                30.0,
     8897.5,          7259.4,          1100.8,          7.3437
                    0.8,      37000.0,                34.0,
     9988.0,          7694.8,          1405.5,          9.0829
                    0.8,      37000.0,                38.0,
    11045.4,          8096.9,          1723.9,         11.2241
                    0.8,      37000.0,                42.0,
    12082.7,          8479.1,          2044.4,         12.6821
                    0.8,      37000.0,                46.0,
    13095.6,          8836.7,          2378.8,         14.0299
                    0.8,      37000.0,                48.0,
    13590.1,          9003.6,          2555.9,         14.6924
```

```
                    0.8,      37000.0,                     50.0,
    14082.1,        9167.9,           2733.1,           15.3273
                    0.8,      39000.0,                     21.0,
     5736.6,        5513.4,            458.1,            8.1251
                    0.8,      39000.0,                     22.0,
     6011.0,        5653.8,            509.0,            6.6908
                    0.8,      39000.0,                     24.0,
     6533.9,        5908.8,            624.2,            4.1644
                    0.8,      39000.0,                     26.0,
     7032.1,        6139.1,            742.6,            5.1849
                    0.8,      39000.0,                     30.0,
     8080.3,        6592.0,           1005.9,            7.0545
                    0.8,      39000.0,                     34.0,
     9071.6,        6988.0,           1282.5,            8.7133
                    0.8,      39000.0,                     38.0,
    10032.7,        7353.7,           1571.6,            10.789
                    0.8,      39000.0,                     42.0,
    10975.1,        7701.0,           1862.9,           12.1943
                    0.8,      39000.0,                     46.0,
    11895.7,        8026.0,           2166.6,           13.4902
                    0.8,      39000.0,                     48.0,
    12345.1,        8177.9,           2327.3,           14.1257
                    0.8,      39000.0,                     50.0,
    12792.3,        8327.3,           2488.0,           14.7358
                    0.8,      41000.0,                     21.0,
     5206.8,        5004.0,            422.5,            7.8759
                    0.8,      41000.0,                     22.0,
     5456.4,        5131.9,            468.7,            6.4921
                    0.8,      41000.0,                     24.0,
     5932.1,        5364.1,            573.0,            3.9876
                    0.8,      41000.0,                     26.0,
     6385.0,        5573.6,            680.6,            4.9727
                    0.8,      41000.0,                     30.0,
     7338.1,        5985.7,            919.7,            6.7752
                    0.8,      41000.0,                     34.0,
     8239.2,        6346.0,           1170.8,            8.3569
                    0.8,      41000.0,                     38.0,
     9112.8,        6678.6,           1433.3,           10.3696
                    0.8,      41000.0,                     42.0,
     9969.4,        6994.3,           1697.9,           11.7243
```

```
                    0.8,        41000.0,                      46.0,
         10805.8,          7289.8,             1973.9,           12.9692
                    0.8,        41000.0,                      48.0,
         11214.3,          7427.8,             2119.8,            13.581
                    0.8,        41000.0,                      50.0,
         11620.8,          7563.8,             2265.5,           14.1666
                    0.8,        43000.0,                      21.0,
          4725.6,          4541.3,              390.2,            7.6384
                    0.8,        43000.0,                      22.0,
          4952.6,          4657.7,              432.1,            6.3043
                    0.8,        43000.0,                      24.0,
          5385.5,          4869.4,              526.6,            3.8166
                    0.8,        43000.0,                      26.0,
          5797.4,          5060.1,              624.5,            4.7702
                    0.8,        43000.0,                      30.0,
          6664.0,          5435.0,              841.4,            6.5063
                    0.8,        43000.0,                      34.0,
          7483.3,          5762.9,             1069.3,            8.0181
                    0.8,        43000.0,                      38.0,
          8277.5,          6065.4,             1307.7,            9.9658
                    0.8,        43000.0,                      42.0,
          9055.9,          6352.3,             1548.6,           11.2718
                    0.8,        43000.0,                      46.0,
          9816.3,          6621.1,             1799.0,           12.4684
                    0.8,        43000.0,                      48.0,
         10187.6,          6746.6,             1931.5,           13.0575
                    0.8,        43000.0,                      50.0,
         10556.9,          6870.2,             2063.5,           13.6188
                   0.85,        20000.0,                      21.0,
         15269.7,         14726.9,             1148.4,            7.7963
                   0.85,        20000.0,                      22.0,
         15940.5,         15071.8,             1286.4,            8.5229
                   0.85,        20000.0,                      24.0,
         17216.3,         15696.2,             1605.0,            10.037
                   0.85,        20000.0,                      26.0,
         18437.2,         16265.7,             1923.0,           12.0458
                   0.85,        20000.0,                      30.0,
         21018.2,         17399.0,             2650.6,            14.837
                   0.85,        20000.0,                      34.0,
         23493.4,         18426.4,             3408.9,           17.2645
```

```
                  0.85,       20000.0,                38.0,
        25884.8,        19370.2,            4206.9,         19.4142
                  0.85,       20000.0,                42.0,
        28229.3,        20267.0,            5007.6,         23.7977
                  0.85,       20000.0,                46.0,
        30511.4,        21101.3,            5843.9,          46.826
                  0.85,       20000.0,                48.0,
        31623.3,        21489.2,            6287.4,         58.5793
                  0.85,       20000.0,                50.0,
        32726.4,        21868.7,            6737.1,         67.2902
                  0.85,       25000.0,                21.0,
        12322.9,        11885.2,             918.3,          6.2921
                  0.85,       25000.0,                22.0,
        12864.9,        12164.6,            1027.0,          6.9769
                  0.85,       25000.0,                24.0,
        13894.8,        12669.2,            1277.4,          8.3905
                  0.85,       25000.0,                26.0,
        14880.2,        13129.4,            1527.5,          9.6187
                  0.85,       25000.0,                30.0,
        16963.6,        14045.5,            2098.8,         12.7509
                  0.85,       25000.0,                34.0,
        18961.6,        14876.3,            2695.0,         15.0269
                  0.85,       25000.0,                38.0,
        20891.7,        15639.3,            3321.4,         17.0425
                  0.85,       25000.0,                42.0,
        22783.5,        16363.9,            3949.4,         18.7996
                  0.85,       25000.0,                46.0,
        24625.9,        17039.0,            4606.3,         21.7264
                  0.85,       25000.0,                48.0,
        25523.3,        17352.8,            4953.5,         29.2527
                  0.85,       25000.0,                50.0,
        26413.9,        17659.8,            5304.8,          41.448
                  0.85,       30000.0,                21.0,
         9852.7,         9503.0,             730.5,           4.817
                  0.85,       30000.0,                22.0,
        10286.3,         9726.8,             814.8,          5.4569
                  0.85,       30000.0,                24.0,
        11110.8,        10131.7,            1009.8,          6.7844
                  0.85,       30000.0,                26.0,
        11899.3,        10500.5,            1204.7,          7.9341
```

```
                   0.85,       30000.0,                  30.0,
         13566.3,        11234.9,             1649.0,          10.6836
                   0.85,       30000.0,                  34.0,
         15164.4,        11900.5,             2112.4,          12.8292
                   0.85,       30000.0,                  38.0,
         16708.5,        12512.1,             2599.3,          14.7084
                   0.85,       30000.0,                  42.0,
         18221.3,        13092.4,             3087.7,           16.346
                   0.85,       30000.0,                  46.0,
         19695.1,        13633.6,             3597.6,          17.8596
                   0.85,       30000.0,                  48.0,
         20413.3,        13885.5,             3867.4,          18.6028
                   0.85,       30000.0,                  50.0,
         21126.4,        14132.3,             4139.7,          19.3169
                   0.85,       35000.0,                  21.0,
          7797.5,         7520.8,              578.7,           6.6087
                   0.85,       35000.0,                  22.0,
          8141.8,         7699.1,              643.4,           5.1541
                   0.85,       35000.0,                  24.0,
          8795.7,         8020.9,              793.4,           5.2408
                   0.85,       35000.0,                  26.0,
          9420.6,         8313.7,              943.8,           6.3186
                   0.85,       35000.0,                  30.0,
         10741.8,         8897.0,             1284.8,           8.3419
                   0.85,       35000.0,                  34.0,
         12008.4,         9425.6,             1641.7,           10.725
                   0.85,       35000.0,                  38.0,
         13232.1,         9911.4,             2016.1,          12.4766
                   0.85,       35000.0,                  42.0,
         14430.9,        10372.2,             2391.9,           13.999
                   0.85,       35000.0,                  46.0,
         15598.6,        10802.0,             2783.9,          15.4058
                   0.85,       35000.0,                  48.0,
         16167.5,        11002.0,             2990.7,           16.095
                   0.85,       35000.0,                  50.0,
         16732.6,        11198.0,             3199.2,          16.7582
                   0.85,       37000.0,                  21.0,
          7080.7,         6829.3,              529.2,           7.1696
                   0.85,       37000.0,                  22.0,
          7393.9,         6991.7,              587.4,           5.7449
```

```
                   0.85,       37000.0,                   24.0,
         7988.5,          7284.6,             722.8,             4.8169
                   0.85,       37000.0,                   26.0,
         8556.5,          7551.0,             858.6,             5.8624
                   0.85,       37000.0,                   30.0,
         9757.7,          8081.8,            1166.5,             7.8217
                   0.85,       37000.0,                   34.0,
        10908.7,          8562.5,            1488.5,            10.0732
                   0.85,       37000.0,                   38.0,
        12020.9,          9004.3,            1826.1,            11.7857
                   0.85,       37000.0,                   42.0,
        13110.2,          9423.3,            2165.5,            13.2592
                   0.85,       37000.0,                   46.0,
        14171.0,          9813.8,            2518.9,            14.6171
                   0.85,       37000.0,                   48.0,
        14688.6,          9996.1,            2705.6,            15.2834
                   0.85,       37000.0,                   50.0,
        15202.3,         10174.6,            2893.4,            15.9241
                   0.85,       39000.0,                   21.0,
         6427.5,          6199.1,             487.0,             6.9546
                   0.85,       39000.0,                   22.0,
         6712.2,          6346.7,             539.4,              5.589
                   0.85,       39000.0,                   24.0,
         7253.1,          6613.6,             662.4,             4.6165
                   0.85,       39000.0,                   26.0,
         7769.4,          6855.8,             785.8,             5.6258
                   0.85,       39000.0,                   30.0,
         8861.3,          7338.7,            1065.5,              7.514
                   0.85,       39000.0,                   34.0,
         9907.4,          7775.8,            1357.5,             9.6651
                   0.85,       39000.0,                   38.0,
        10918.3,          8177.7,            1664.3,            11.3296
                   0.85,       39000.0,                   42.0,
        11908.0,          8558.4,            1972.4,            12.7479
                   0.85,       39000.0,                   46.0,
        12872.3,          8913.7,            2293.5,             14.056
                   0.85,       39000.0,                   48.0,
        13342.3,          9079.2,            2463.0,            14.6952
                   0.85,       39000.0,                   50.0,
        13809.4,          9241.8,            2633.5,            15.3107
```

```
                    0.9,       35000.0,                  21.0,
          8689.9,         8405.4,            612.6,            5.4642
                    0.9,       35000.0,                  22.0,
          9048.1,         8592.9,            681.7,            4.4604
                    0.9,       35000.0,                  24.0,
          9727.3,         8930.8,            841.5,            5.7208
                    0.9,       35000.0,                  26.0,
         10383.2,         9245.3,            999.3,            6.7969
                    0.9,       35000.0,                  30.0,
         11785.1,         9888.5,           1365.6,            8.8636
                    0.9,       35000.0,                  34.0,
         13126.8,        10471.6,           1745.1,           11.3302
                    0.9,       35000.0,                  38.0,
         14418.8,        11005.1,           2142.8,           13.0965
                    0.9,       35000.0,                  42.0,
         15682.3,        11509.9,           2543.0,           14.6396
                    0.9,       35000.0,                  46.0,
         16909.5,        11978.5,           2959.3,           16.0593
                    0.9,       35000.0,                  48.0,
         17506.0,        12195.7,           3179.8,           16.7568
                    0.9,       35000.0,                  50.0,
         18098.7,        12409.1,           3401.9,           17.4262
                    0.9,       37000.0,                  21.0,
          7891.3,         7632.9,            559.6,            6.0554
                    0.9,       37000.0,                  22.0,
          8217.1,         7803.6,            621.9,            4.6273
                    0.9,       37000.0,                  24.0,
          8834.9,         8111.3,            766.2,            5.2826
                    0.9,       37000.0,                  26.0,
          9431.0,         8397.2,            908.7,            6.3254
                    0.9,       37000.0,                  30.0,
         10705.5,         8982.6,           1239.4,             8.326
                    0.9,       37000.0,                  34.0,
         11924.9,         9512.8,           1581.3,           10.6732
                    0.9,       37000.0,                  38.0,
         13099.4,         9998.1,           1940.8,           12.3857
                    0.9,       37000.0,                  42.0,
         14247.6,        10457.1,           2302.0,            13.878
                    0.9,       37000.0,                  46.0,
         15363.1,        10883.5,           2678.2,           15.2518
```

```
                    0.9,        37000.0,                         48.0,
        15905.5,        11081.3,            2876.5,              15.9229
                    0.9,        37000.0,                         50.0,
        16443.8,        11275.0,            3076.2,              16.5691
                    0.9,        39000.0,                         21.0,
         7163.6,         6928.7,             514.4,               5.8837
                    0.9,        39000.0,                         22.0,
         7459.9,         7084.2,             570.9,               4.5067
                    0.9,        39000.0,                         24.0,
         8021.6,         7364.1,             701.6,               5.0624
                    0.9,        39000.0,                         26.0,
         8563.7,         7624.5,             831.4,               6.0727
                    0.9,        39000.0,                         30.0,
         9722.3,         8156.9,            1131.6,               7.9996
                    0.9,        39000.0,                         34.0,
        10830.8,         8639.2,            1442.2,              10.2576
                    0.9,        39000.0,                         38.0,
        11898.0,         9080.3,            1768.2,              11.9058
                    0.9,        39000.0,                         42.0,
        12941.4,         9497.6,            2096.6,              13.3448
                    0.9,        39000.0,                         46.0,
        13955.2,         9885.2,            2438.2,              14.6669
                    0.9,        39000.0,                         48.0,
        14447.9,        10064.9,            2618.2,              15.3126
                    0.9,        39000.0,                         50.0,
        14937.5,        10241.3,            2799.5,              15.9332
```

## B.8 Aviary: DCA Patrol Mission

```
import aviary.api as av
from aviary.api import Aircraft, Mission



aircraft_filename = "models/test_aircraft/
    Interceptor1_Aircraft_FwFm.csv"
optimizer = "IPOPT"
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'aviary_history.db'
```

```
restart_filename = None
max_iter = 5

phase_info ={
    "pre_mission": {"include_takeoff": True, "optimize_mass": True},
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": True,
            "num_segments": 3,
            "order": 3,
            "solve_for_distance": True,
            "initial_mach": (0.15, "unitless"),
            "final_mach": (0.5, "unitless"),
            "mach_bounds": ((0.0, 0.6), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((0.0, 40000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 15.0), "min"),
            "duration_bounds": ((10.0, 30.0), "min"),
            "add_initial_mass_constraint": False,
        },
    },
    "cruise": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": True,
            "num_segments": 10,
            "order": 5,
            "solve_for_distance": True,
            "initial_mach": (0.5, "unitless"),
```

```
                "final_mach": (0.78, "unitless"),
                "mach_bounds": ((0.6, 0.8), "unitless"),
                "initial_altitude": (35000.0, "ft"),
                "final_altitude": (38000.0, "ft"),
                "altitude_bounds": ((34000.0, 40000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((15.0, 30.0), "min"),
                "duration_bounds": ((40.0, 80.0), "min"),
            },
        },
# Combat Air-Patrol
    "cruise_1": {
        "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 5,
            "solve_for_distance": False,
            "initial_mach": (0.78, "unitless"),
            "final_mach": (0.5, "unitless"),
            "mach_bounds": ((0.4, 0.8), "unitless"),
            "initial_altitude": (38000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((34000.0, 38000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": True,
            "fix_duration": True,
            "initial_bounds": ((60.0, 200.0), "min"),
            "duration_bounds": ((240.0, 240.0), "min"),
        },
    },
 #Dash
    "cruise_2": {
```

```
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.5, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.4, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((35000.0, 35000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((200, 260.0), "min"),
            "duration_bounds": ((2.0, 5.0), "min"),
        },
    },

    #Combat
    "cruise_3": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 2,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.9, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((35000.0, 35000.0), "ft"),
            "throttle_enforcement": "path_constraint",
```

```
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((200.0, 250.0), "min"),
                "duration_bounds": ((2.0, 2.0), "min"),
            },
    },

    #Climb/accel
    "climb_2": {
        "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.78, "unitless"),
            "mach_bounds": ((0.7, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (38000.0, "ft"),
            "altitude_bounds": ((35000.0, 40000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((180, 220.0), "min"),
            "duration_bounds": ((5.0, 20.0), "min"),
        },

    },
    "cruise_4": {
        "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 10,
```

```
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.78, "unitless"),
            "final_mach": (0.8, "unitless"),
            "mach_bounds": ((0.7, 0.9), "unitless"),
            "initial_altitude": (38000.0, "ft"),
            "final_altitude": (40000.0, "ft"),
            "altitude_bounds": ((35000.0, 40000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((250.0, 300.0), "min"),
            "duration_bounds": ((40.0, 80.0), "min"),
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.8, "unitless"),
            "final_mach": (0.25, "unitless"),
            "mach_bounds": ((0.1, 0.8), "unitless"),
            "initial_altitude": (40000.0, "ft"),
            "final_altitude": (500.0, "ft"),
            "altitude_bounds": ((0.0, 45000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((300.0, 320.0), "min"),
            "duration_bounds": ((10.0, 20.0), "min"),
        },

    },
```

```
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (800, "nmi"),
    },
}

prob = av.AviaryProblem(analysis_scheme)

# Load aircraft and options data from user
# Allow for user overrides here
prob.load_inputs(aircraft_filename, phase_info)

prob.check_and_preprocess_inputs()

prob.add_pre_mission_systems()

prob.add_phases()

prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

prob.add_driver(optimizer, max_iter=max_iter)


prob.model.add_design_var(av.Mission.Design.GROSS_MASS, units='lbm
    ',
                            lower=32000, upper=37000.0, ref=36733)
prob.model.add_design_var(av.Aircraft.Wing.AREA, units='ft**2',
                            lower=400, upper=500, ref=445.03)
prob.model.add_design_var(av.Aircraft.Wing.SPAN, units='ft',
                            lower=42, upper=50, ref=42.08)
prob.model.add_design_var(av.Mission.Design.RANGE, units='NM',
                            lower=500, upper=800, ref=1000)
prob.model.add_design_var(av.Aircraft.Wing.ASPECT_RATIO,
                            lower=3, upper=5, ref=3.98)
prob.model.add_design_var(av.Aircraft.HorizontalTail.AREA, units="
    ft**2",
```

```
                            lower=150, upper=160, ref=153.46)
prob.model.add_design_var(av.Aircraft.VerticalTail.AREA, units="ft**2",
                            lower=50, upper=55, ref=53.9)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_OVER_DRAG,
                            lower=3, upper=5, ref=3.5)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                            lower=1.6, upper=2, ref=1.8)
prob.model.add_design_var(av.Mission.Landing.LIFT_COEFFICIENT_MAX,
                            lower=1.8, upper=2.5, ref=2.2)
prob.add_objective(objective_type=objective_type)

prob.setup()

prob.set_initial_guesses()

prob.run_aviary_problem(record_filename, restart_filename=restart_filename)

print('--------------')
print('Sizing Results')
print('--------------')
print(f'Design Range = {prob.get_val(av.Mission.Design.RANGE, units="NM")}')
print(f'Summary Range = {prob.get_val(av.Mission.Summary.RANGE, units="NM")}')
print(f'Total fuel capacity = {prob.get_val(av.Aircraft.Fuel.TOTAL_CAPACITY, units="lbm")}')
print(f'Empty mass = {prob.get_val(av.Aircraft.Design.OPERATING_MASS, units="lbm")}')
print(f'Total Payload mass = {prob.get_val(av.Aircraft.CrewPayload.TOTAL_PAYLOAD_MASS, units="lbm")}')
print(f'Design Gross mass = {prob.get_val(av.Mission.Design.GROSS_MASS, units="lbm")}')
print(f'Summary Gross mass = {prob.get_val(av.Mission.Summary.GROSS_MASS, units="lbm")}')
print(f'Wing Area = {prob.get_val(av.Aircraft.Wing.AREA, units="ft**2")}')
print(f'Wing Span = {prob.get_val(av.Aircraft.Wing.SPAN, units="ft")}')
```

```python
print(f'Aspect Ratio = {prob.get_val(av.Aircraft.Wing.ASPECT_RATIO)}')
print(f'HT Area = {prob.get_val(av.Aircraft.HorizontalTail.AREA, units="ft**2")}')
print(f'VT Area = {prob.get_val(av.Aircraft.VerticalTail.AREA, units="ft**2")}')
print(f'Takeoff L/D = {prob.get_val(av.Mission.Takeoff.LIFT_OVER_DRAG)}')
print(f'Takeoff Clmax = {prob.get_val(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX)}')
print(f'Landing Clmax = {prob.get_val(av.Mission.Landing.LIFT_COEFFICIENT_MAX)}')
```

## B.9 Aviary: PDI Mission

```python
import aviary.api as av
from aviary.api import Aircraft, Mission



aircraft_filename = "models/test_aircraft/Interceptor1_Aircraft_FwFm.csv"
optimizer = "IPOPT"
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'aviary_history.db'
restart_filename = None
max_iter = 5

phase_info ={
    "pre_mission": {"include_takeoff": True, "optimize_mass": True},
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
```

```
            "solve_for_distance": False,
            "initial_mach": (0.2, "unitless"),
            "final_mach": (0.8, "unitless"),
            "mach_bounds": ((0.1, 0.9), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((0.0, 35000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((4.0, 20.0), "min"),
            "duration_bounds": ((15.0, 35.0), "min"),
            "add_initial_mass_constraint": False,
        },
    },
    #dash
    "cruise": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.8, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.8, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((35000.0, 35000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((10.0, 20.0), "min"),
            "duration_bounds": ((15.0, 25.0), "min"),
        },
    },
```

```
    #combat
    "cruise_1": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.85, 0.95), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((34000.0, 36000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 15.0), "min"),
            "duration_bounds": ((5.0, 10.0), "min")
        },
    },
    # climb/Accel
     "cruise_2": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.78, "unitless"),
            "mach_bounds": ((0.7, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (40000.0, "ft"),
            "altitude_bounds": ((34000.0, 42000.0), "ft"),
```

```
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 10.0), "min"),
            "duration_bounds": ((5.0, 15.0), "min")
        },
    },
    "cruise_3": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.6, "unitless"),
            "final_mach": (0.6, "unitless"),
            "mach_bounds": ((0.5, 0.7), "unitless"),
            "initial_altitude": (40000.0, "ft"),
            "final_altitude": (34000.0, "ft"),
            "altitude_bounds": ((33000.0, 40000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((60.0, 100.0), "min"),
            "duration_bounds": ((100.0, 200.0), "min")
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
```

```
                "initial_mach": (0.7, "unitless"),
                "final_mach": (0.5, "unitless"),
                "mach_bounds": ((0.4, 0.8), "unitless"),
                "initial_altitude": (34000.0, "ft"),
                "final_altitude": (500.0, "ft"),
                "altitude_bounds": ((0.0, 35000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": False,
                "constrain_final": True,
                "fix_duration": False,
                "initial_bounds": ((5.0, 15.0), "min"),
                "duration_bounds": ((15.0, 30.0), "min")
        },
    },
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (500, "nmi"),
    },
}

prob = av.AviaryProblem(analysis_scheme)

# Load aircraft and options data from user
# Allow to overrides here
prob.load_inputs(aircraft_filename, phase_info)

prob.check_and_preprocess_inputs()

prob.add_pre_mission_systems()

prob.add_phases()

prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

prob.add_driver(optimizer, max_iter=max_iter)

```

```

prob.model.add_design_var(av.Mission.Design.GROSS_MASS, units='lbm',
                          lower=32000, upper=37000.0, ref=36733)
prob.model.add_design_var(av.Aircraft.Wing.AREA, units='ft**2',
                          lower=400, upper=500, ref=445.03)
prob.model.add_design_var(av.Aircraft.Wing.SPAN, units='ft',
                          lower=42, upper=50, ref=42.08)
prob.model.add_design_var(av.Mission.Design.RANGE, units='NM',
                          lower=500, upper=800, ref=1000)
prob.model.add_design_var(av.Aircraft.Wing.ASPECT_RATIO,
                          lower=3, upper=5, ref=3.98)
prob.model.add_design_var(av.Aircraft.HorizontalTail.AREA, units="ft**2",
                          lower=150, upper=160, ref=153.46)
prob.model.add_design_var(av.Aircraft.VerticalTail.AREA, units="ft**2",
                          lower=50, upper=55, ref=53.9)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_OVER_DRAG,
                          lower=3, upper=5, ref=3.5)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                          lower=1.6, upper=2, ref=1.8)
prob.model.add_design_var(av.Mission.Landing.LIFT_COEFFICIENT_MAX,
                          lower=1.8, upper=2.5, ref=2.2)
prob.add_objective(objective_type=objective_type)

prob.setup()

prob.set_initial_guesses()

prob.run_aviary_problem(record_filename, restart_filename=restart_filename)

print('--------------')
print('Sizing Results')
print('--------------')
print(f'Design Range = {prob.get_val(av.Mission.Design.RANGE, units="NM")}')
print(f'Summary Range = {prob.get_val(av.Mission.Summary.RANGE, units="NM")}')
```

```
print(f'Total fuel capacity = {prob.get_val(av.Aircraft.Fuel.
    TOTAL_CAPACITY, units="lbm")}')
print(f'Empty mass = {prob.get_val(av.Aircraft.Design.
    OPERATING_MASS, units="lbm")}')
print(f'Total Payload mass = {prob.get_val(av.Aircraft.CrewPayload
    .TOTAL_PAYLOAD_MASS, units="lbm")}')
print(f'Design Gross mass = {prob.get_val(av.Mission.Design.
    GROSS_MASS, units="lbm")}')
print(f'Summary Gross mass = {prob.get_val(av.Mission.Summary.
    GROSS_MASS, units="lbm")}')
print(f'Wing Area = {prob.get_val(av.Aircraft.Wing.AREA, units="ft
    **2")}')
print(f'Wing Span = {prob.get_val(av.Aircraft.Wing.SPAN, units="ft
    ")}')
print(f'Aspect Ratio = {prob.get_val(av.Aircraft.Wing.ASPECT_RATIO
    )}')
print(f'HT Area = {prob.get_val(av.Aircraft.HorizontalTail.AREA,
    units="ft**2")}')
print(f'VT Area = {prob.get_val(av.Aircraft.VerticalTail.AREA,
    units="ft**2")}')
print(f'Takeoff L/D = {prob.get_val(av.Mission.Takeoff.
    LIFT_OVER_DRAG)}')
print(f'Takeoff Clmax = {prob.get_val(av.Mission.Takeoff.
    LIFT_COEFFICIENT_MAX)}')
print(f'Landing Clmax = {prob.get_val(av.Mission.Landing.
    LIFT_COEFFICIENT_MAX)}')
```

## B.10 Aviary: Escort Mission

```
import aviary.api as av
from aviary.api import Aircraft, Mission



aircraft_filename = "models/test_aircraft/
    Interceptor_Aircraft_FwFm.csv"
optimizer = "IPOPT"
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'aviary_history.db'
```

```
restart_filename = None
max_iter = 5

phase_info ={
     "pre_mission": {"include_takeoff": True, "optimize_mass": True},
     "climb": {
         "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
         "user_options": {
             "optimize_mach": False,
             "optimize_altitude": False,
             "num_segments": 10,
             "order": 3,
             "solve_for_distance": False,
             "initial_mach": (0.2, "unitless"),
             "final_mach": (0.8, "unitless"),
             "mach_bounds": ((0.1, 0.9), "unitless"),
             "initial_altitude": (0.0, "ft"),
             "final_altitude": (35000.0, "ft"),
             "altitude_bounds": ((0.0, 40000.0), "ft"),
             "throttle_enforcement": "path_constraint",
             "fix_initial": True,
             "constrain_final": False,
             "fix_duration": False,
             "initial_bounds": ((4.0, 20.0), "min"),
             "duration_bounds": ((15.0, 35.0), "min"),
             "add_initial_mass_constraint": False,
         },
     },
     #dash
    "cruise": {
         "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
         "user_options": {
             "optimize_mach": False,
             "optimize_altitude": False,
             "num_segments": 10,
             "order": 3,
             "solve_for_distance": False,
```

```
            "initial_mach": (0.8, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.8, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((30000.0, 40000.0), "ft")"initial_bounds": ((60.0, 100.0), "min"),
            "duration_bounds": ((100.0, 200.0), "min"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((10.0, 20.0), "min"),
            "duration_bounds": ((15.0, 25.0), "min"),
        },
    },
    #escort
    "cruise_1": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.5, "unitless"),
            "mach_bounds": ((0.4, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (38000.0, "ft"),
            "altitude_bounds": ((34000.0, 40000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((200.0, 280.0), "min"),
            "duration_bounds": ((150.0, 360.0), "min"),
        },
    },
```

```
# climb/Accel
  "cruise_2": {
      "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
      "user_options": {
          "optimize_mach": True,
          "optimize_altitude": True,
          "num_segments": 10,
          "order": 3,
          "solve_for_distance": False,
          "initial_mach": (0.5, "unitless"),
          "final_mach": (0.78, "unitless"),
          "mach_bounds": ((0.4, 0.8), "unitless"),
          "initial_altitude": (38000.0, "ft"),
          "final_altitude": (40000.0, "ft"),
          "altitude_bounds": ((35000.0, 40000.0), "ft"),
          "throttle_enforcement": "path_constraint",
          "fix_initial": False,
          "constrain_final": False,
          "fix_duration": False,
          "initial_bounds": ((5.0, 10.0), "min"),
          "duration_bounds": ((5.0, 15.0), "min"),
      },
  },
  "cruise_3": {
      "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
      "user_options": {
          "optimize_mach": True,
          "optimize_altitude": True,
          "num_segments": 10,
          "order": 3,
          "solve_for_distance": False,
          "initial_mach": (0.78, "unitless"),
          "final_mach": (0.6, "unitless"),
          "mach_bounds": ((0.5, 0.8), "unitless"),
          "initial_altitude": (40000.0, "ft"),
          "final_altitude": (35000.0, "ft"),
          "altitude_bounds": ((34000.0, 40000.0), "ft"),
          "throttle_enforcement": "boundary_constraint",
```

```
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((60.0, 100.0), "min"),
            "duration_bounds": ((100.0, 200.0), "min")
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.6, "unitless"),
            "final_mach": (0.3, "unitless"),
            "mach_bounds": ((0.3, 0.6), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (500.0, "ft"),
            "altitude_bounds": ((0.0, 30000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((5.0, 25.0), "min"),
            "duration_bounds": ((15.0, 30.0), "min"),
        },
    },
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (500, "nmi"),
    },
}

prob = av.AviaryProblem(analysis_scheme)

# Load aircraft and options data from user
```

```
# Allow for user overrides here
prob.load_inputs(aircraft_filename, phase_info)

prob.check_and_preprocess_inputs()

prob.add_pre_mission_systems()

prob.add_phases()

prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

prob.add_driver(optimizer, max_iter=max_iter)


prob.model.add_design_var(av.Mission.Design.GROSS_MASS, units='lbm',
                            lower=32000, upper=37000.0, ref=36733)
prob.model.add_design_var(av.Aircraft.Wing.AREA, units='ft**2',
                            lower=400, upper=500, ref=445.03)
prob.model.add_design_var(av.Aircraft.Wing.SPAN, units='ft',
                            lower=42, upper=50, ref=42.08)
prob.model.add_design_var(av.Mission.Design.RANGE, units='NM',
                            lower=500, upper=800, ref=1000)
prob.model.add_design_var(av.Aircraft.Wing.ASPECT_RATIO,
                            lower=3, upper=5, ref=3.98)
prob.model.add_design_var(av.Aircraft.HorizontalTail.AREA, units="ft**2",
                            lower=150, upper=160, ref=153.46)
prob.model.add_design_var(av.Aircraft.VerticalTail.AREA, units="ft**2",
                            lower=50, upper=55, ref=53.9)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_OVER_DRAG,
                            lower=3, upper=5, ref=3.5)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                            lower=1.6, upper=2, ref=1.8)
prob.model.add_design_var(av.Mission.Landing.LIFT_COEFFICIENT_MAX,
                            lower=1.8, upper=2.5, ref=2.2)
```

```
prob.add_objective(objective_type=objective_type)

prob.setup()

prob.set_initial_guesses()

prob.run_aviary_problem(record_filename, restart_filename=restart_filename)

print('--------------')
print('Sizing Results')
print('--------------')
print(f'Design Range = {prob.get_val(av.Mission.Design.RANGE, units="NM")}')
print(f'Summary Range = {prob.get_val(av.Mission.Summary.RANGE, units="NM")}')
print(f'Total fuel capacity = {prob.get_val(av.Aircraft.Fuel.TOTAL_CAPACITY, units="lbm")}')
print(f'Empty mass = {prob.get_val(av.Aircraft.Design.OPERATING_MASS, units="lbm")}')
print(f'Total Payload mass = {prob.get_val(av.Aircraft.CrewPayload.TOTAL_PAYLOAD_MASS, units="lbm")}')
print(f'Design Gross mass = {prob.get_val(av.Mission.Design.GROSS_MASS, units="lbm")}')
print(f'Summary Gross mass = {prob.get_val(av.Mission.Summary.GROSS_MASS, units="lbm")}')
print(f'Wing Area = {prob.get_val(av.Aircraft.Wing.AREA, units="ft**2")}')
print(f'Wing Span = {prob.get_val(av.Aircraft.Wing.SPAN, units="ft")}')
print(f'Aspect Ratio = {prob.get_val(av.Aircraft.Wing.ASPECT_RATIO)}')
print(f'HT Area = {prob.get_val(av.Aircraft.HorizontalTail.AREA, units="ft**2")}')
print(f'VT Area = {prob.get_val(av.Aircraft.VerticalTail.AREA, units="ft**2")}')
print(f'Takeoff L/D = {prob.get_val(av.Mission.Takeoff.LIFT_OVER_DRAG)}')
print(f'Takeoff Clmax = {prob.get_val(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX)}')
```

```
print(f'Landing Clmax = {prob.get_val(av.Mission.Landing.
    LIFT_COEFFICIENT_MAX)}')
```

## B.11 Aviary: DCA Patrol Phase Info

```
phase_info ={
    "pre_mission": {"include_takeoff": True, "optimize_mass": True
    },
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": True,
            "initial_mach": (0.2, "unitless"),
            "final_mach": (0.8, "unitless"),
            "mach_bounds": ((0.0, 0.8), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (40000.0, "ft"),
            "altitude_bounds": ((0.0, 45000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 15.0), "min"),
            "duration_bounds": ((10.0, 30.0), "min"),
            "add_initial_mass_constraint": False,
        },
    },
    "cruise": {
        "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
```

```
           "solve_for_distance": False,
           "initial_mach": (0.8, "unitless"),
           "final_mach": (0.8, "unitless"),
           "mach_bounds": ((0.75, 0.9), "unitless"),
           "initial_altitude": (40000.0, "ft"),
           "final_altitude": (40000.0, "ft"),
           "altitude_bounds": ((40000.0, 40000.0), "ft"),
           "throttle_enforcement": "path_constraint",
           "fix_initial": False,
           "constrain_final": True,
           "fix_duration": False,
           "initial_bounds": ((60.0, 120.0), "min"),
           "duration_bounds": ((100.0, 300.0), "min"),
       },
    },
# Combat Air-Patrol
   "cruise_1": {
       "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
       "user_options": {
           "optimize_mach": True,
           "optimize_altitude": False,
           "num_segments": 1,
           "order": 3,
           "solve_for_distance": False,
           "initial_mach": (0.8, "unitless"),
           "final_mach": (0.6, "unitless"),
           "mach_bounds": ((0.55, 0.9), "unitless"),
           "initial_altitude": (40000.0, "ft"),
           "final_altitude": (35000.0, "ft"),
           "altitude_bounds": ((35000.0, 45000.0), "ft"),
           "throttle_enforcement": "path_constraint",
           "fix_initial": False,
           "constrain_final": False,
           "fix_duration": False,
           "initial_bounds": ((200.0, 280.0), "min"),
           "duration_bounds": ((240.0, 240.0), "min"),
       },
   },
 #Dash
```

```
    "cruise_2": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.6, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.4, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((35000.0, 35000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((1.0, 2.5), "min"),
            "duration_bounds": ((1.5, 3.0), "min"),
        },
    },
    #Accel
    "cruise_3": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.9, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (25000.0, "ft"),
            "altitude_bounds": ((20000.0, 35000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
```

```
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 15.0), "min"),
            "duration_bounds": ((10.0, 30.0), "min"),
        },

    },
    "cruise_4": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.7, "unitless"),
            "mach_bounds": ((0.6, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (40000.0, "ft"),
            "altitude_bounds": ((30000.0, 45000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((10.0, 30.0), "min"),
            "duration_bounds": ((20.0, 60.0), "min"),
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
```

```
                "initial_mach": (0.6, "unitless"),
                "final_mach": (0.4, "unitless"),
                "mach_bounds": ((0.4, 0.6), "unitless"),
                "initial_altitude": (45000.0, "ft"),
                "final_altitude": (0.0, "ft"),
                "altitude_bounds": ((0.0, 30000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": False,
                "constrain_final": True,
                "fix_duration": False,
                "initial_bounds": ((5.0, 15.0), "min"),
                "duration_bounds": ((10.0, 25.0), "min"),
            },

    },
    "post_mission": {
        "include_landing": True,
        "constrain_range": True,
        "target_range": (800, "nmi"),
    },
}
```

## B.12 Aviary: PDI Phase Info

```
aircraft_filename = "models/test_aircraft/Interceptor1_Aircraft_FwFm.csv"
optimizer = "IPOPT"
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'aviary_history.db'
restart_filename = None
max_iter = 5

phase_info ={
    "pre_mission": {"include_takeoff": True, "optimize_mass": True},
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
```

```
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.2, "unitless"),
            "final_mach": (0.8, "unitless"),
            "mach_bounds": ((0.1, 0.9), "unitless"),
            "initial_altitude": (0.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((0.0, 35000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": True,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((4.0, 20.0), "min"),
            "duration_bounds": ((15.0, 35.0), "min"),
            "add_initial_mass_constraint": False,
        },
    },
    #dash
    "cruise": {
        "subsystem_options": {"core_aerodynamics": {"method": "
computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 5,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.8, "unitless"),
            "final_mach": (0.9, "unitless"),
            "mach_bounds": ((0.8, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((35000.0, 35000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
```

```
                "initial_bounds": ((10.0, 20.0), "min"),
                "duration_bounds": ((15.0, 25.0), "min"),
            },
        },
        #combat
        "cruise_1": {
            "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
            "user_options": {
                "optimize_mach": False,
                "optimize_altitude": False,
                "num_segments": 5,
                "order": 3,
                "solve_for_distance": False,
                "initial_mach": (0.9, "unitless"),
                "final_mach": (0.9, "unitless"),
                "mach_bounds": ((0.85, 0.95), "unitless"),
                "initial_altitude": (35000.0, "ft"),
                "final_altitude": (35000.0, "ft"),
                "altitude_bounds": ((34000.0, 36000.0), "ft"),
                "throttle_enforcement": "boundary_constraint",
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((5.0, 15.0), "min"),
                "duration_bounds": ((5.0, 10.0), "min")
            },
        },
        # climb/Accel
         "cruise_2": {
            "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
            "user_options": {
                "optimize_mach": True,
                "optimize_altitude": True,
                "num_segments": 10,
                "order": 3,
                "solve_for_distance": False,
                "initial_mach": (0.9, "unitless"),
                "final_mach": (0.78, "unitless"),
```

```
                "mach_bounds": ((0.7, 0.9), "unitless"),
                "initial_altitude": (35000.0, "ft"),
                "final_altitude": (40000.0, "ft"),
                "altitude_bounds": ((34000.0, 42000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((5.0, 10.0), "min"),
                "duration_bounds": ((5.0, 15.0), "min")
            },
        },
         "cruise_3": {
            "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
            "user_options": {
                "optimize_mach": True,
                "optimize_altitude": True,
                "num_segments": 10,
                "order": 3,
                "solve_for_distance": False,
                "initial_mach": (0.6, "unitless"),
                "final_mach": (0.6, "unitless"),
                "mach_bounds": ((0.5, 0.7), "unitless"),
                "initial_altitude": (40000.0, "ft"),
                "final_altitude": (34000.0, "ft"),
                "altitude_bounds": ((33000.0, 40000.0), "ft"),
                "throttle_enforcement": "boundary_constraint",
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((60.0, 100.0), "min"),
                "duration_bounds": ((100.0, 200.0), "min")
            },
        },
        "descent": {
            "subsystem_options": {"core_aerodynamics": {"method": "
    computed"}},
            "user_options": {
                "optimize_mach": False,
```

```
                "optimize_altitude": False,
                "num_segments": 5,
                "order": 3,
                "solve_for_distance": False,
                "initial_mach": (0.7, "unitless"),
                "final_mach": (0.5, "unitless"),
                "mach_bounds": ((0.4, 0.8), "unitless"),
                "initial_altitude": (34000.0, "ft"),
                "final_altitude": (500.0, "ft"),
                "altitude_bounds": ((0.0, 35000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": False,
                "constrain_final": True,
                "fix_duration": False,
                "initial_bounds": ((5.0, 15.0), "min"),
                "duration_bounds": ((15.0, 30.0), "min")
            },
        },
        "post_mission": {
            "include_landing": False,
            "constrain_range": True,
            "target_range": (500, "nmi"),
        },
}
```

## B.13 Aviary: Escort Phase Info

```
phase_info ={
    "pre_mission": {"include_takeoff": True, "optimize_mass": True},
    "climb": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.2, "unitless"),
```

```
                "final_mach": (0.8, "unitless"),
                "mach_bounds": ((0.1, 0.9), "unitless"),
                "initial_altitude": (0.0, "ft"),
                "final_altitude": (35000.0, "ft"),
                "altitude_bounds": ((0.0, 40000.0), "ft"),
                "throttle_enforcement": "path_constraint",
                "fix_initial": True,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((4.0, 20.0), "min"),
                "duration_bounds": ((15.0, 35.0), "min"),
                "add_initial_mass_constraint": False,
            },
        },
        #dash
        "cruise": {
            "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
            "user_options": {
                "optimize_mach": False,
                "optimize_altitude": False,
                "num_segments": 10,
                "order": 3,
                "solve_for_distance": False,
                "initial_mach": (0.8, "unitless"),
                "final_mach": (0.9, "unitless"),
                "mach_bounds": ((0.8, 0.9), "unitless"),
                "initial_altitude": (35000.0, "ft"),
                "final_altitude": (35000.0, "ft"),
                "altitude_bounds": ((30000.0, 40000.0), "ft")"initial_bounds": ((60.0, 100.0), "min"),
                "duration_bounds": ((100.0, 200.0), "min"),
                "throttle_enforcement": "boundary_constraint",
                "fix_initial": False,
                "constrain_final": False,
                "fix_duration": False,
                "initial_bounds": ((10.0, 20.0), "min"),
                "duration_bounds": ((15.0, 25.0), "min"),
            },
        },
```

```
    #escort
    "cruise_1": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.9, "unitless"),
            "final_mach": (0.5, "unitless"),
            "mach_bounds": ((0.4, 0.9), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (38000.0, "ft"),
            "altitude_bounds": ((34000.0, 40000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((200.0, 280.0), "min"),
            "duration_bounds": ((150.0, 360.0), "min"),
        },
    },
  # climb/Accel
    "cruise_2": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.5, "unitless"),
            "final_mach": (0.78, "unitless"),
            "mach_bounds": ((0.4, 0.8), "unitless"),
            "initial_altitude": (38000.0, "ft"),
            "final_altitude": (40000.0, "ft"),
            "altitude_bounds": ((35000.0, 40000.0), "ft"),
```

```
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((5.0, 10.0), "min"),
            "duration_bounds": ((5.0, 15.0), "min"),
        },
    },
    "cruise_3": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": True,
            "optimize_altitude": True,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
            "initial_mach": (0.78, "unitless"),
            "final_mach": (0.6, "unitless"),
            "mach_bounds": ((0.5, 0.8), "unitless"),
            "initial_altitude": (40000.0, "ft"),
            "final_altitude": (35000.0, "ft"),
            "altitude_bounds": ((34000.0, 40000.0), "ft"),
            "throttle_enforcement": "boundary_constraint",
            "fix_initial": False,
            "constrain_final": False,
            "fix_duration": False,
            "initial_bounds": ((60.0, 100.0), "min"),
            "duration_bounds": ((100.0, 200.0), "min")
        },
    },
    "descent": {
        "subsystem_options": {"core_aerodynamics": {"method": "computed"}},
        "user_options": {
            "optimize_mach": False,
            "optimize_altitude": False,
            "num_segments": 10,
            "order": 3,
            "solve_for_distance": False,
```

```
            "initial_mach": (0.6, "unitless"),
            "final_mach": (0.3, "unitless"),
            "mach_bounds": ((0.3, 0.6), "unitless"),
            "initial_altitude": (35000.0, "ft"),
            "final_altitude": (500.0, "ft"),
            "altitude_bounds": ((0.0, 30000.0), "ft"),
            "throttle_enforcement": "path_constraint",
            "fix_initial": False,
            "constrain_final": True,
            "fix_duration": False,
            "initial_bounds": ((5.0, 25.0), "min"),
            "duration_bounds": ((15.0, 30.0), "min"),
        },
    },
    "post_mission": {
        "include_landing": False,
        "constrain_range": True,
        "target_range": (500, "nmi"),
    },
}
```

## B.14 Aviary: Aviary Run Mission Command

```
aircraft_filename = "models/test_aircraft/Interceptor1_Aircraft_FwFm.csv"
optimizer = "IPOPT"
analysis_scheme = av.AnalysisScheme.COLLOCATION
objective_type = None
record_filename = 'DCAP-sizing.db'
restart_filename = None
max_iter = 5
```

## B.15 Aviary: Aviary Analysis Problem

```
prob = av.AviaryProblem(analysis_scheme)

# Load aircraft and options data from user
# Allow for user overrides here
prob.load_inputs(aircraft_filename, phase_info)

```

```python
prob.check_and_preprocess_inputs()

prob.add_pre_mission_systems()

prob.add_phases()

prob.add_post_mission_systems()

# Link phases and variables
prob.link_phases()

prob.add_driver(optimizer, max_iter=max_iter)


prob.model.add_design_var(av.Mission.Design.GROSS_MASS, units='lbm',
                          lower=32000, upper=37000.0, ref=36733)
prob.model.add_design_var(av.Aircraft.Wing.AREA, units='ft**2',
                          lower=400, upper=500, ref=486.26)
prob.model.add_design_var(av.Aircraft.Wing.SPAN, units='ft',
                          lower=43, upper=50, ref=44)
prob.model.add_design_var(av.Mission.Design.RANGE, units='NM',
                          lower=500, upper=800, ref=700)
prob.model.add_design_var(av.Aircraft.Wing.ASPECT_RATIO,
                          lower=3, upper=5, ref=3.98)
prob.model.add_design_var(av.Aircraft.HorizontalTail.AREA, units="ft**2",
                          lower=85, upper=100, ref=86)
prob.model.add_design_var(av.Aircraft.VerticalTail.AREA, units="ft**2",
                          lower=58, upper=65, ref=58.6)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_OVER_DRAG,
                          lower=10, upper=12, ref=10)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                          lower=1.6, upper=2, ref=1.8)
prob.model.add_design_var(av.Mission.Landing.LIFT_COEFFICIENT_MAX,
                          lower=1.8, upper=2.5, ref=2.2)
prob.add_objective(objective_type=objective_type)

prob.setup()
```

```

prob.set_initial_guesses()

prob.run_aviary_problem(record_filename, restart_filename=
    restart_filename)
```

## B.16 Aviary: Design Variables

```
prob.model.add_design_var(av.Mission.Design.GROSS_MASS, units='lbm
    ',
                            lower=32000, upper=37000.0, ref=36733)
prob.model.add_design_var(av.Aircraft.Wing.AREA, units='ft**2',
                            lower=400, upper=500, ref=486.26)
prob.model.add_design_var(av.Aircraft.Wing.SPAN, units='ft',
                            lower=43, upper=50, ref=44)
prob.model.add_design_var(av.Mission.Design.RANGE, units='NM',
                            lower=500, upper=800, ref=700)
prob.model.add_design_var(av.Aircraft.Wing.ASPECT_RATIO,
                            lower=3, upper=5, ref=3.98)
prob.model.add_design_var(av.Aircraft.HorizontalTail.AREA, units="
    ft**2",
                            lower=85, upper=100, ref=86)
prob.model.add_design_var(av.Aircraft.VerticalTail.AREA, units="ft
    **2",
                            lower=58, upper=65, ref=58.6)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_OVER_DRAG,
                            lower=10, upper=12, ref=10)
prob.model.add_design_var(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX,
                            lower=1.6, upper=2, ref=1.8)
prob.model.add_design_var(av.Mission.Landing.LIFT_COEFFICIENT_MAX,
                            lower=1.8, upper=2.5, ref=2.2)
```

## B.17 Aviary: Print Desired Sizing Variables

```
print('---------------')
print('Sizing Results')
print('---------------')
print(f'Design Range = {prob.get_val(av.Mission.Design.RANGE,
    units="NM")}')
```

```python
print(f'Summary Range = {prob.get_val(av.Mission.Summary.RANGE, units="NM")}')
print(f'Total fuel capacity = {prob.get_val(av.Aircraft.Fuel.TOTAL_CAPACITY, units="lbm")}')
print(f'Empty mass = {prob.get_val(av.Aircraft.Design.OPERATING_MASS, units="lbm")}')
print(f'Total Payload mass = {prob.get_val(av.Aircraft.CrewPayload.TOTAL_PAYLOAD_MASS, units="lbm")}')
print(f'Design Gross mass = {prob.get_val(av.Mission.Design.GROSS_MASS, units="lbm")}')
print(f'Summary Gross mass = {prob.get_val(av.Mission.Summary.GROSS_MASS, units="lbm")}')
print(f'Wing Area = {prob.get_val(av.Aircraft.Wing.AREA, units="ft**2")}')
print(f'Wing Span = {prob.get_val(av.Aircraft.Wing.SPAN, units="ft")}')
print(f'Aspect Ratio = {prob.get_val(av.Aircraft.Wing.ASPECT_RATIO)}')
print(f'HT Area = {prob.get_val(av.Aircraft.HorizontalTail.AREA, units="ft**2")}')
print(f'VT Area = {prob.get_val(av.Aircraft.VerticalTail.AREA, units="ft**2")}')
print(f'Takeoff L/D = {prob.get_val(av.Mission.Takeoff.LIFT_OVER_DRAG)}')
print(f'Takeoff Clmax = {prob.get_val(av.Mission.Takeoff.LIFT_COEFFICIENT_MAX)}')
print(f'Landing Clmax = {prob.get_val(av.Mission.Landing.LIFT_COEFFICIENT_MAX)}')
```

# Turnitin Report

Samuel-Theofilus-thesis-manuscript-2024.pdf

ORIGINALITY REPORT

| 7% | 5% | 2% | 3% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Embry Riddle Aeronautical University<br>Student Paper | 1% |
| 2 | vdocuments.mx<br>Internet Source | 1% |
| 3 | www.coursehero.com<br>Internet Source | <1% |
| 4 | repository.sgu.ac.id<br>Internet Source | <1% |
| 5 | Submitted to Virginia Polytechnic Institute and State University<br>Student Paper | <1% |
| 6 | repository.ipmi.ac.id<br>Internet Source | <1% |
| 7 | Submitted to Cranfield University<br>Student Paper | <1% |
| 8 | archive.aoe.vt.edu<br>Internet Source | <1% |
| 9 | Submitted to University of Hertfordshire<br>Student Paper | <1% |
| 10 | sjsu.edu<br>Internet Source | <1% |
| 11 | Mohammad H. Sadraey. "Aircraft Performance - An Engineering Approach", CRC Press, 2017 | <1% |

Publication

| | | |
|---|---|---|
| 12 | smartech.gatech.edu<br>Internet Source | <1% |
| 13 | Submitted to University of Stirling<br>Student Paper | <1% |
| 14 | www.apra.gov.au<br>Internet Source | <1% |
| 15 | Submitted to Kaplan College<br>Student Paper | <1% |
| 16 | timesofindia.indiatimes.com<br>Internet Source | <1% |
| 17 | eprints.nottingham.ac.uk<br>Internet Source | <1% |
| 18 | www-dweb-cors.dev.archive.org<br>Internet Source | <1% |
| 19 | www.aiaa.org<br>Internet Source | <1% |
| 20 | Pedro David Bravo-Mosquera, Hernán Darío Cerón-Muñoz, Guillermo Díaz-Vázquez, Fernando Martini Catalano. "Conceptual design and CFD analysis of a new prototype of agricultural aircraft", Aerospace Science and Technology, 2018<br>Publication | <1% |
| 21 | mafiadoc.com<br>Internet Source | <1% |
| 22 | Submitted to University of Florida<br>Student Paper | <1% |
| 23 | aerospace.gdgoenka-university.com<br>Internet Source | <1% |

| | | |
|---|---|---|
| 24 | www.scribd.com<br>Internet Source | <1 % |
| 25 | Submitted to University of Sheffield<br>Student Paper | <1 % |
| 26 | d-nb.info<br>Internet Source | <1 % |
| 27 | deepblue.lib.umich.edu<br>Internet Source | <1 % |
| 28 | Submitted to Middle East Technical University<br>Student Paper | <1 % |
| 29 | Robert A. McDonald. "Mission Performance Considered as Point Performance in Aircraft Design", Journal of Aircraft, 2011<br>Publication | <1 % |
| 30 | Submitted to Emirates Aviation College, Aerospace & Academic Studies<br>Student Paper | <1 % |
| 31 | Submitted to University of Aberdeen<br>Student Paper | <1 % |
| 32 | repositories.nust.edu.pk<br>Internet Source | <1 % |
| 33 | Submitted to Kingston University<br>Student Paper | <1 % |
| 34 | airliquide.com.ph<br>Internet Source | <1 % |
| 35 | etd.aau.edu.et<br>Internet Source | <1 % |
| 36 | psasir.upm.edu.my<br>Internet Source | <1 % |
| | repository.sustech.edu | |

| | | |
|---|---|---|
| 37 | Internet Source | <1% |
| 38 | Submitted to Universidad Catolica San Antonio de Murcia<br>Student Paper | <1% |
| 39 | Submitted to University of Adelaide<br>Student Paper | <1% |
| 40 | ia601005.us.archive.org<br>Internet Source | <1% |
| 41 | iarjset.com<br>Internet Source | <1% |
| 42 | Pandu Prayoga. "Chapter 4 The Indonesian Political-Economy: Maritime Development in Fisheries and Commerce", Springer Science and Business Media LLC, 2022<br>Publication | <1% |
| 43 | elibrary.tucl.edu.np<br>Internet Source | <1% |
| 44 | firstaerosquadron.com<br>Internet Source | <1% |
| 45 | article.sapub.org<br>Internet Source | <1% |
| 46 | www.prestigeaustralia.com<br>Internet Source | <1% |
| 47 | Raul Corralejo, Peter Harley. "Smith diagram for Low Reynolds number axial compressor rotors", European Turbomachinery Society, 2017<br>Publication | <1% |
| 48 | library.berlin-international.de<br>Internet Source | <1% |

| | | |
|---|---|---|
| 49 | patents.justia.com<br>Internet Source | <1% |
| 50 | www.scoop.it<br>Internet Source | <1% |
| 51 | Ekaterina Kozaeva, Svetlana Volkova, Marta R.A. Matos, Mariela P. Mezzina et al. "Model-guided dynamic control of essential metabolic nodes boosts acetyl-coenzyme A–dependent bioproduction in rewired Pseudomonas putida", Metabolic Engineering, 2021<br>Publication | <1% |
| 52 | Kenneth E. Cox, K. D. Williamson. "Hydrogen: Its Technology and Implications - Utilization of Hydrogen", CRC Press, 2018<br>Publication | <1% |
| 53 | Submitted to The University of Texas at Arlington<br>Student Paper | <1% |
| 54 | Zou, Xiao Kang, Hoi Yuen Lee, Edward Sai Cheong Chan, and Ping Xiang. "Cost design optimisation of concrete transfer beam structures", HKIE Transactions, 2014.<br>Publication | <1% |
| 55 | etheses.whiterose.ac.uk<br>Internet Source | <1% |
| 56 | hdl.handle.net<br>Internet Source | <1% |
| 57 | worldlibrary.org<br>Internet Source | <1% |
| 58 | www.eucass.eu<br>Internet Source | <1% |

| | | |
|---|---|---|
| 59 | Andrew B. Lambe, Joaquim R. R. A. Martins. "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes", Structural and Multidisciplinary Optimization, 2012<br>Publication | <1 % |
| 60 | G. Daniel Brewer. "Hydrogen Aircraft Technology", CRC Press, 2017<br>Publication | <1 % |
| 61 | Karim Abu Salem, Giuseppe Palaia, Aldo Frediani, Erasmo Carrera. "The box-wing configuration: a critical review of design approaches and applications", Progress in Aerospace Sciences, 2025<br>Publication | <1 % |
| 62 | Yicheng Sun, Howard Smith. "Low-boom low-drag optimization in a multidisciplinary design analysis optimization environment", Aerospace Science and Technology, 2019<br>Publication | <1 % |
| 63 | Zhijie Lu, Eun-Suk Yang, Daniel DeLaurentis, Dimitri Mavris. "Formulation and Test of an Object-Oriented Approach to Aircraft Sizing", 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004<br>Publication | <1 % |
| 64 | archive.org<br>Internet Source | <1 % |
| 65 | ebin.pub<br>Internet Source | <1 % |
| 66 | elib.dlr.de<br>Internet Source | <1 % |

67 engrxiv.org
Internet Source
<1 %

68 eprints.soton.ac.uk
Internet Source
<1 %

69 eurasiantimes.com
Internet Source
<1 %

70 fenix.tecnico.ulisboa.pt
Internet Source
<1 %

71 lib.dr.iastate.edu
Internet Source
<1 %

72 ntrs.nasa.gov
Internet Source
<1 %

73 oddjob.utias.utoronto.ca
Internet Source
<1 %

74 pdfs.semanticscholar.org
Internet Source
<1 %

75 peterdsharpe.github.io
Internet Source
<1 %

76 publications.polymtl.ca
Internet Source
<1 %

77 sofrep.com
Internet Source
<1 %

78 Jack D. Mattingly, William H. Heiser, Keith M. Boyer, Brenda A. Haven, David T. Pratt. "Aircraft Engine Design, Third Edition", American Institute of Aeronautics and Astronautics (AIAA), 2018
Publication
<1 %

79 Jun-xue Leng, Yang Shen, Wei Huang, Kai An, Can-can Zhou, Zhen-guo Wang. "Multi-fidelity
<1 %

data mining-based design optimization framework for scramjet-based aircraft", Energy, 2024
Publication

80 Mohammad H. Sadraey. "Aircraft Design", Wiley, 2012 <1%
Publication

81 Steven A. Brandt, John J. Bertin, Randall J. Stiles, Ray Whitford. "Introduction to Aeronautics", American Institute of Aeronautics and Astronautics (AIAA), 2004 <1%
Publication

82 Yu Cai, Dushhyanth Rajaram, Dimitri N. Mavris. "Simultaneous aircraft sizing and multi-objective optimization considering off-design mission performance during early design", Aerospace Science and Technology, 2022 <1%
Publication

83 Jenkinson, Lloyd R., and James F. Marchman. "Preliminary design", Aircraft Design Projects, 2003. <1%
Publication

84 S. Amirreza S. Madani, Erfan Vaezi, Amir Keshmiri. "CFD-driven evaluation of winglet cant angle: Insights into aerodynamic and aeroacoustic performance", Results in Engineering, 2025 <1%
Publication

85 Samuele Graziani, Leonardo Ialongo, Davide Ferretto. "A Numerical Approach for the Integration of Sonic Boom Requirements into the Matching Chart", Aerospace, 2025 <1%
Publication

| | | |
|---|---|---|
| 86 | StarBriefs Plus, 2004.<br>Publication | <1 % |
| 87 | Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025<br>Publication | <1 % |
| 88 | Yiyuan Ma, Ali Elham. "Twin-fuselage configuration for improving fuel efficiency of passenger aircraft", Aerospace Science and Technology, 2021<br>Publication | <1 % |
| 89 | Ziemer, Sven, Martin Glas, and Gernot Stenz. "A Conceptual Design Tool for multi-disciplinary aircraft design", 2011 Aerospace Conference, 2011.<br>Publication | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | On | | |

# Curriculum Vitae

| | Basic Information |
|---|---|
| Name | Samuel Theofilus Tamonob |
| Place of Birth | Tangerang |
| Date of Birth | 22/09/2001 |
| Address | Alam Sutera, South Tangerang, Banten, Indonesia |
| Phone Number | +62 8880106077 |
| Email | samueltamonob2001@gmail.com |
| **Year** | **Education** |
| 2021 - Present | International University Liaison Indonesia (IULI) |
| 2016 - 2019 | Saint Johns Catholic Senior High School |
| 2013 - 2015 | Saint Johns Catholic Junior High School |
| 2007 - 2012 | Binus Elementary School Serpong |
| **Year** | **Experience** |
| 2023 - 2024 | Leader of BUING Team AIAA Aircraft Design Competition (4$^{th}$ place) |
| | - Led Heavy Lift Aircraft (HLA) design and team operations |
| | - Managed stability and control division |
| | - Ensured communication and collaboration among members |
| 2023 | Aircraft Maintenance Training General Aircraft Maintenance |
| | - Sheet metal fabrication, composite lamination |
| | - Engine ground runs, landing gear inspection |
| 2023 | Product Development Project Quadcopter Drone Team |
| | - Designed small cargo quadcopter for payload optimization |
| **Section** | **Skills and Tools** |

| | |
|---|---|
| | Teamwork, Leadership, Critical Thinking<br>Communication Skills, MS Office, Project Libre<br>XFLR5 |
| Section | Languages |
| | Indonesian, English, Deutsch |