Multidisciplinary Design Optimization of an Aircraft Considering Path-Dependent Performance

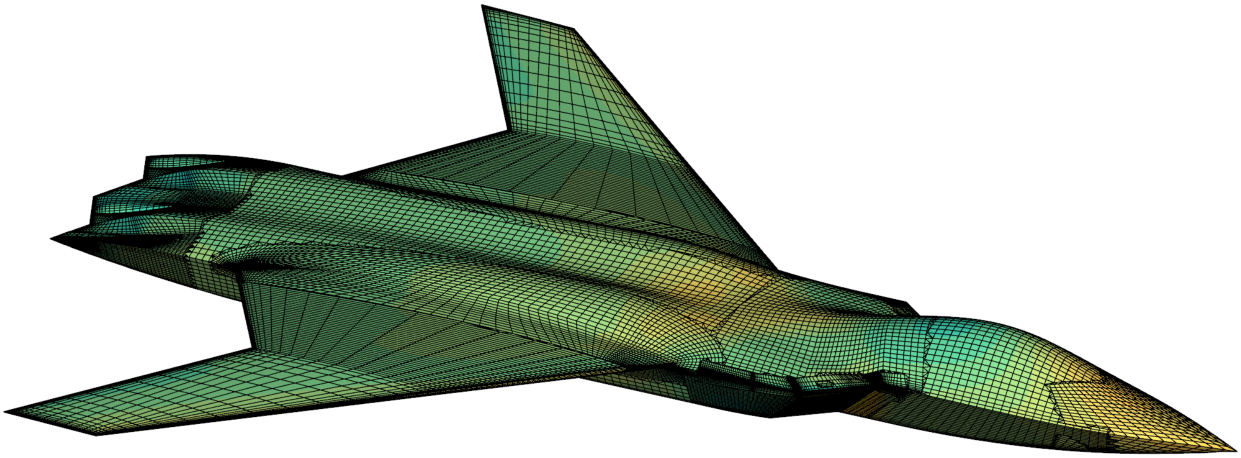
by

John P. Jasa

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2020

Doctoral Committee:

Professor Joaquim R. R. A. Martins, Co-Chair
Assistant Research Scientist Charles A. Mader, Co-Chair
Professor Krzysztof J. Fidkowski
Dr. Justin S. Gray, NASA Glenn Research Center
Professor Yin Lu Young



John P. Jasa

johnjasa@umich.edu

ORCID iD: 0000-0001-5442-2792

To my mother, Lisa, for being the most selfless person that I will ever know.

Acknowledgments

There are numerous people who directly or indirectly helped me create this dissertation, and I truly wish I could thank them all by name here.

My advisor, Prof. Joaquim R.R.A. Martins, gave me the intrinsic desire to optimize everything – aircraft, trajectories, my life – and for that I am extremely thankful. Prof. Martins helped me look at the world with a curious and critical eye, something I will take with me through the rest of my research journey. I especially thank Justin Gray, who encouraged me to apply to the MDO Lab, mentored me while I was at NASA, and worked with me through my PhD with my best interests in mind. I cannot overstate how positively Justin influenced my life. I also owe Dr. Charles Mader a huge thanks for patiently helping me (remotely and in person) conduct better research. I joke with my labmates that we stand on the shoulders of Canadians, and Dr. Mader provides a strong foundation. I thank Prof. Krzysztof Fidkowski, who beyond serving on my committee, worked to improve the lives of graduate students by serving as the graduate chair during my last few years here. I also thank Prof. Julie Young for serving on my committee and for providing constructive feedback to my research.

I'll miss many facets of Ann Arbor, but most of all I'll miss my colleagues in the MDO Lab. Countless wonderful hours have been spent ideating, debugging, sketching, writing, eating, playing frisbee, and traveling with my labmates. Prof. John Hwang (then a post-doc) was the first to greet me, and he served as a mentor, collaborator, and friend in my first few years getting my PhD. So many veteran members also helped me along the way, including Ney Secco, Tim Brooks, Gaetan Kenway, Dave Burdette, Mohamed Bouhlel, Ping He, and Xiaosong Du. My contemporary labmates are all fantastic, especially Shamsheer Chauhan, Nick Bons, Josh Anibal, Ben Brelje, Eirikur Jonsson, Anil Yildirim, Yingqian Liao, Sicheng He, and Gustavo Halila. I know the future of the group is bright with Neil Wu, Marco Mangano, Sabet Seraj, and Shugo Kaneko leading tomorrow's research efforts.

During my PhD I was fortunate to also work at NASA Glenn Research Center, where only the most kind and helpful people are doing great work. Jon Seidel patiently helped me understand supersonic engines. Eric Hendricks poured engine modeling knowledge into me and can smoke me on any bike ride. Rob Falck assisted me with mission optimization more than any normal person would have. To Eliot Aretskin-Hariton, thanks for housing me and making me think about potential side hustles. Many other people made my time at NASA wonderful.

The Aerospace Department at the University of Michigan is home to an assortment of sensational people. My fellow graduate students and GSAC members have continually made the department better for all. I hope the great work of the diversity, equity, and inclusion committee continues and grows. To the support staff – Denise Phelps and Kathy Miller especially – I owe you many thanks. I also thank the custodial staff for their tireless and oft-unnoticed efforts.

I wouldn't be here without my family and their unending support, and I tell them that as often as I can. Mom and Dad, thank you for flipping over the placemats at restaurants and writing out math problems for me to solve. You sacrificed so much to give me the best life you could. Katie, thank you for being a shining example of fun-loving academic excellence. To Philip, Mike, Mariah, Helen, and the other newer members of my family, I'm so glad you're with me now. To Banjo Pete and Smokey, who are sleeping on a blanket beside me as I write this; thank you for being so large and so fluffy.

Thank you to my dearest, Kate, for continually choosing to be with me despite us meeting in high school. Your belief in my abilities astounds me and your unequivocal support in our pursuits of happiness propels me. I can't imagine a better partner for life than you.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ACRONYMS

XDSM extended design structure matrix

CFD computational fluid dynamics

MDO multidisciplinary design optimization

CRM Common Research Model

uCRM undeflected Common Research Model

ESAV Efficient Supersonic Air Vehicle

DEW directed energy weapons

AFRL Air Force Research Laboratory

NASA National Aeronautics and Space Administration

RANS Reynolds-averaged Navier–Stokes

NPSS Numerical Propulsion System Simulation

EBD effectiveness-based design

MPP multi-parameter performance

RMTS regularized minimal-energy tensor-product spline

SMT Surrogate Modeling Toolbox

UQ uncertainty quantification

TMS thermal management system

FTMS fuel thermal management system

eVTOL electric vertical takeoff and landing

MAUD modular analysis and unified derivatives

ODE ordinary differential equation

LGL Legendre-Gauss-Lobatto

LGR Legendre-Gauss-Radau

DOE design of experiments

VLM vortex lattice method

FEA finite element analysis

DOF degree-of-freedom

ACM air-cycle machine

HPC high-pressure compressor

IGV inlet guide vanes

VCE variable cycle engine

VABI variable area bypass injector

SLS sea-level static

OPR overall pressure ratio

FPR fan pressure ratio

ER extraction ratio

BPR bypass ratio

FAR fuel-to-air ratio

MTOW maximum takeoff weight

RTO rolling takeoff

TOC top of climb

SC subsonic

AOD after ordinance drop

TSFC thrust-specific fuel consumption

CEA chemical equilibrium analysis

ABSTRACT

Aircraft are multidisciplinary systems that are challenging to design due to interactions between the subsystems. The relevant disciplines, such as aerodynamic, thermal, and propulsion systems, must be considered simultaneously using a path-dependent formulation to accurately assess aircraft performance. The overarching contribution of this work is the construction and exploration of a coupled aero-thermal-propulsive-mission multidisciplinary model to optimize supersonic aircraft considering their path-dependent performance.

First, the mission, thermal, and propulsion disciplines are examined in detail. The aerostructural design and mission of a morphing-wing aircraft is optimized before the optimal flight profile for a supersonic strike mission is investigated. Then a fuel thermal management system, commonly used to dissipate excess thermal energy from supersonic aircraft, is constructed and presented. Engine design is then investigated through two main applications: multipoint optimization of a variable-cycle engine and coupled thermal-engine optimization considering a bypass duct heat exchanger.

This culminates into a fully-coupled path-dependent mission optimization problem considering the aerodynamic, propulsion, and thermal systems. This large-scale optimization problem captures non-intuitive design trades that single disciplinary models and path-independent methods cannot resolve. Although the focal application is a supersonic aircraft, the methods presented here are applicable to any air or space vehicle and other path-dependent problems.

This level of highly-coupled design optimization considering these disciplines has not been performed before. The framework, modeling, and results from this dissertation will be useful for designers of commercial and military aircraft. Specifically, optimizing the design and trajectory of commercial aircraft to minimize fuel usage leads to a more sustainable and more connected world as the rate of air travel continues to increase. The methods presented are flexible and powerful enough to design supersonic military aircraft systems, as demonstrated using an application aircraft. This dissertation has been shaped through direct collaborations with NASA, the Air Force Research Lab (AFRL), and other academic institutions, which shows the broad appeal and applicability of this work to a multitude of design problems.

CHAPTER 1

Introduction

1.1 What are path-dependent systems?

Path-dependent systems, where the path taken affects the system performance, are common in aerospace engineering. The path-dependent nature of aircraft must be considered to accurately assess their performance. In some cases, we are tracking the literal path of the aircraft, but the phrase *path-dependent* here refers to any system where the time history of states within that system affects performance. Interesting state histories include short-term effects, such as the temperatures of components within the aircraft, to long-term effects, such as the structural fatigue accumulated in the airframe from previous missions. For instance, if engineers ignored the time-dependent thermal history of the aircraft when designing the coolant subsystems, they would inaccurately estimate the aircraft's capabilities. Because we seek to maximize the capabilities of performance-critical aircraft and to evaluate new aircraft configurations, we study path-dependent design optimization in this dissertation.

To better discuss path-dependent problems, let us briefly introduce some relevant terminology. *States* are the values that we are interested in tracking across time, such as the temperature of a component, the amount of fuel in a tank, or the voltage at a terminal. *Controls* are any value that we vary across time to affect the system, such as throttle, coolant mass flow rate, or control surface deflection. In some literature, these are called *dynamic controls*. *Design parameters* are other controllable variables that are fixed with time, such as wingspan of an aircraft, the thickness of structural supports, or the size of the engines. These might also be called *static controls*. *Constraints* are any values from the system that we want to constrain at certain time points. Constraints may be also be states, but they do not need to be. Any value from the system may be a constraint, such as the temperature of a component, the altitude of the aircraft, or the structural stress. In an optimization context, we also have the *objective*, which is the quantity we are trying to minimize or maximize. These terms will be used throughout this dissertation, and we now provide some example problems to reinforce the terms' meanings.

Path-dependent problems exist in almost every field. To draw parallels across applications and disciplines, we introduce a table listing the states, controls, design parameters, and constraints for a few simple representative problems in Table 1.1. We include example problems from aircraft design, wind farm layout design, and plant growth optimization. We do not go into great detail about these problems or why they are interesting within their individual fields of study, but simply show the relevant quantities in each case as examples of path-dependent problems. We also present representative governing equations for each of these example, some of which take the form of an ordinary differential equation (ODE).

The first example in the table is optimizing an electric aircraft by maximizing mission range. We would track the aircraft's position, velocity, and thermal states while varying the throttle and angle of attack. Since it is an electric aircraft, we might be interested in constraining the thermal states of the motor or batteries to ensure they do not violate a limit. Because thermal energy in the aircraft accumulates over time, the temperatures of components are dependent on their previous states, leading to path-dependent effects that may limit flight capabilities. Here, the performance of the aircraft is dependent on a literal, physical path. Example ODEs for this case come from Falck et al. [1], where we solve for

$$\dot{r} = v\alpha \cos \gamma$$

and

$$\dot{Q}_{\text{motor}} = P_{\text{motor}}(1 - \eta_{\text{motor}})$$

where r is the aircraft range, v is its velocity, α is the angle of attack, γ is the flight path angle, Q_{motor} is the heat generated by the motor, P_{motor} is the motor power, and η_{motor} is the motor efficiency.

The wind farm example, though closely related to aerospace engineering, is not centered around aircraft design. We could maximize energy production from a collection of wind turbines while controlling the turbine yaw angles. Instead of tracking a physical path, we would track the wind speed and direction, as well as turbine blade speed and power production so the optimizer would know how best to yaw each individual turbine. Downstream turbines are dependent on the fluid path that is obstructed by the upstream turbines, leading to a path-dependent system when evaluating performance of the entire farm. The expected power density of the wind farm can be expressed as

$$E[P](x, y, \Psi)/A$$

where P is the power, x and y are the locations of the wind turbines, Ψ is the yaw angle of the turbines, and A is the area occupied by the wind farm, as described by Fleming et al. [2].

Our last example is completely removed from aerospace engineering, but is an everyday path-dependent problem. Considering a field of plants, we would want to maximize the amount of viable crop production from that field. The history of sunlight and water that the plants receive influences how they grow and their ability to generate the desired crop output. By controlling the amount of sunlight, water, fertilizer, and pesticide that the plants get, we can maximize the crop production. For a given field, there is a constraint on the amount of biomass that can be produced based on the carbon and nitrogen availability in the soil. Plant growth rate can be modeled as a

Table 1.1: Comparison of example path-dependent problems.

Value	Electric aircraft	Wind farm	Plant growth
Objective	Mission range	Energy production	Crop output
States	Position, velocity, motor temperature	Wind speed, direction	Root, stem, and leaf sizes
Controls	Throttle, angle of attack	Turbine yaw angles	Sunlight, water, fertilizer, pesticides
Design parameters	Wingspan, battery size	Turbine placement, blade length	Number of plants per area
Constraints	State of charge of battery, component temperatures	Structural stresses, yaw rates	Potential biomass based on carbon availability

function of a parameter called the GM-factor, R, which can be computed as

$$\dot{R} = Cg(R) - \sigma R$$

where C is the concentration of nutrients, $g(R)$ is a modified offset step-function, and σ is a parameter that depends on the type of plant, as described by Bessonov and Volpert [3].

We now return to the design and performance of aircraft, which will be the focus of the remainder of this dissertation.

1.2 Review of path-dependent optimization

In this section, we focus on detailing the history of path-dependent optimization, how researchers advanced capabilities to where we are today, and the state-of-the-art of contemporary work. Subsequent chapters will go into more technical detail about specific methods and disciplines, whereas the immediately following subsections provide a broad and roughly chronological history of path-dependent optimization literature.

1.2.1 Early beginnings in a post-war world

The earliest published study on path-dependent optimization of aircraft comes from Fritz Kaiser in 1944 Germany [4]. Kaiser presented the minimum time-to-climb problem using the energy-state approximation for a jet-propelled aircraft, specifically the Me 262, which was the world's first operational jet-powered fighter aircraft [4]. Multiple reviews and reports [5–8] have chronicled the large impact that Kaiser's work had on aircraft performance optimization. Merritt et al. [5] claimed that Kaiser's work was "perhaps the most important analytical development in flight performance to come out of WWII".

With the advent of jet propulsion, the path-dependent performance of aircraft becomes much more important to consider than that of piston-powered aircraft due to jet-powered aircraft being capable of reaching higher speeds. Eugene Cliff [8] provides an explanation on why this is the case.

> The piston-powered aircraft of WWII vintage had limited speed-range so that the amount of stored kinetic energy was generally small compared to the potential energy. Note, however, that the kinetic energy varies as the square of the speed; doubling the speed will increase the kinetic energy by a factor of four. As the speed capabilities increase, the kinetic energy becomes increasingly important [8].

Jet-powered aircraft could operate at much greater speeds, making the coupled consideration of both potential and kinetic energy important. Through this consideration, Kaiser found the optimal minimum time-to-climb profiles for the Me 262 and was able to validate them in the field via flight tests at Lager Lechfeld [4, 5]. In later letters to researchers, Kaiser admitted that his "first report met no interest at all," [5], but his work was the foundation for many investigations into optimal control of aircraft.

In the 1950s, other researchers built upon Kaiser's efforts and continued to develop the energy-state method as a way to perform path-dependent optimization of aircraft. Lush [9] generalized Kaiser's work by extending it to multiple climb techniques and desired end conditions, whereas Rutowski [10] developed a general method to obtain the minimum time or minimum fuel needed to go from one combination of altitude and speed to another. These papers provided the necessary tools to perform general mission optimization of aircraft, though they were still limited in the propulsion and aerodynamic considerations, using simplifications and assumptions to simplify the problem due to the lack of adequate computational resources. The work on flight path optimization from 1944 to 1960 is detailed in a review by Bell [11], which provides a convenient milestone marker for the progress of path-dependent optimization before the advent of digital computers.

1.2.2 Computer-aided design optimization

Solving path-dependent optimization problems became much easier as computers became more common. Researchers computed optimal trajectories for many air and space applications, especially high-speed military vehicles, including rockets, supersonic aircraft, and planetary vehicles [12–15]. A few of these papers solved for the optimal trajectories using gradient-based algorithms [14, 15], which were some of the first works to incorporate numerical optimization into the study of path-dependent problems.

A seminal paper by Bryson and Denham [16] detailed the steepest ascent method for solving optimum programming problems using the adjoint method. This paper included three numerical examples, including the minimum time-to-climb to a set speed and heading for a generic supersonic interceptor, the same problem to maximum altitude, and a hypersonic orbital glider descending from outside the atmosphere [16]. The now-famous optimized flight profiles from their paper are shown in Fig. 1.1. Bryson and Denham presented the mathematics for the optimum programming problems in detail, which led to further adoption of these adjoint and gradient-based methods in the optimal control field. Bryson et al. [17] later further developed the optimal interceptor flight path problem for both range maximization and climb duration minimization and presented what has become a canonical case in optimal control theory. Contemporary work continues to develop and compare methods using the interceptor problem that Bryson et al. developed.

In the 1970s and 1980s, path-dependent optimization evolved through the inclusion of path

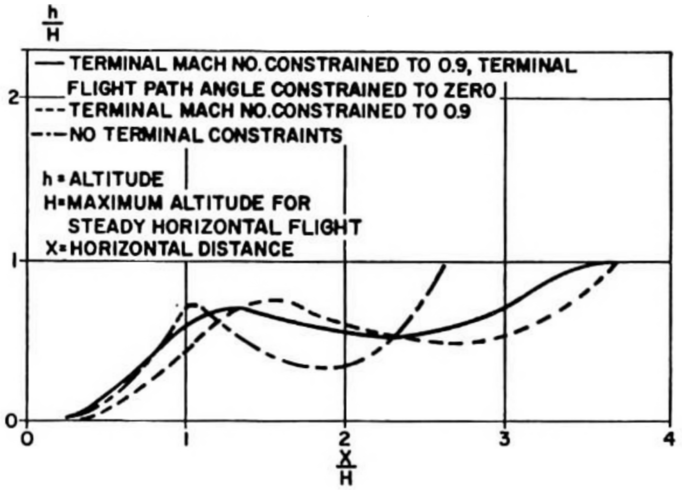


Figure 1.1: Optimized flight profiles for the supersonic interceptor from Bryson and Denham [16].

constraints, uncertainty quantification, and advanced controller theory, as detailed by Bryson in a review article [6]. During this same time, work concerning coupled multidisciplinary design of path-dependent systems began to emerge from the fields of both aircraft design and optimal control. Most of this work focused on gradient-based coupled structural and control optimization to improve vehicle dynamic response to disturbances [18–21]. Although these studies were limited in the number of disciplines they considered, they serve as the basis for multidisciplinary optimization of coupled system in a path-dependent manner.

Near the turn of the century, the systems being studied began to include multiple fidelities and more disciplines. Rahn et al. [22] presented a tool for launch vehicle optimization considering the vehicle shape, mission profile, weights, aerodynamics, and flight control systems, whereas Braun et al. [23] examined a branching launch vehicle trajectory while optimizing the propulsion and stage separation system. Researchers at National Aeronautics and Space Administration (NASA) Langley published a summary of their multidisciplinary design optimization (MDO) and trajectory studies, which included launch vehicles and acoustic noise minimization during trajectory optimization for a rotorcraft [24]. At the same time, Boeing [25] optimized performance for an airbreathing hypersonic vehicle considering its trajectory, aerodynamics, propulsion, stability and control, and weights. They found a 46% increase in performance over the baseline configuration and noted, “such a dramatic performance increase is indicative not only of the power of optimization, but of the difficulty in configuring hypersonic vehicles to synergize the interaction of all vehicle components without MDO methods” [25]. This directly motivates the use of MDO for path-dependent design optimization, and shows its necessity to evaluate new configurations and increase vehicle performance.

1.2.3 Contemporary works and the state-of-the-art

Through the 2000s and 2010s, work examining optimal aircraft trajectories for commercial vehicles became more common, including contrail minimization [26], energy-optimal landings [27], reducing airport congestion [28–30], and reducing environmental impact [28, 31]. The commercial sector is subject to more regulatory constraints than the military sector, especially concerning allowable trajectories, which leads to more constrained optimization problems. Most of these studies focused on optimizing the path of the aircraft or group of aircrafts but did not focus on expanding the problem to include multiple disciplines.

Contemporary work has both expanded the number of disciplines considered and increasing the fidelity of the physical analysis methods. Multiple papers from Air Force Research Laboratory (AFRL) detail their path-dependent MDO capabilities, which include propulsion, thermal management, electrical systems, stability, aerodynamics, economics, and overall utility [32–37]. Although many of these papers cannot go into detail about specific model performance or details, they present general optimization tools and methodologies in use at AFRL. Some terms that these works introduce are multi-parameter performance (MPP), which enables more broad analysis of aircraft performance as detailed by Alyanak and Allison [34], and effectiveness-based design (EBD), which couples MDO and uncertainty quantification (UQ) as detailed by Clark et al. [37]. AFRL researchers developed these methodologies to more accurately assess the complete performance of the aircraft by having holistic vehicle-level metrics instead of single-discipline objectives.

Some of the most relevant preceding work to this dissertation comes from AFRL concerning optimization of the thermal management system (TMS) in a path-dependent context [38–43]. Specifically, Alyanak and Allison [39] design a fuel thermal management system (FTMS) and show how much of an effect the transient nature of the thermal problem and the architecture selected impact aircraft performance. They show that a nominal aircraft's weight increases by 282% when considering the flight heat loads and thermal constraints. Although they admit that the result is unrealistic due to the simplicity of the thermal system architecture, it does show the impact that thermal constraints have on aircraft sizing [39]. The remainder of the relevant papers concerning TMS design will be discussed in Sec. 3.1.

Recently, NASA has been doing mission-based MDO as part of its Transformational Tools and Technologies project at both its Glenn and Langley Research Centers. Falck et al. [1] and Hendricks et al. [44] implemented a collocation-based approach for solving multidisciplinary trajectory problems that efficiently solves both the time integration and optimization problem simultaneously. Schnulo et al. [45] used that same approach to optimize the X-57 aircraft across a full mission with multiple flight segments. Falck et al. [46] continued by expanding the mission optimization tool, Dymos, and investigating its use and scalability for MDO problems. Hendricks et al. [47] expanded Dymos' limits by optimizing the aerostructural, propulsive, and electric performance of a tiltwing urban air mobility vehicle. Simultaneously, researchers at NASA Langley are developing LEAPS, which is a modular multidisciplinary tool to assess aircraft performance [48–50].

This dissertation directly builds on the work presented by Falck et al. [1] and Hendricks et al. [44] that coupled mission optimization with propulsion analysis and thermal constraints. Their work showed that by using a higher-order collocation method with gradient-based optimization and analytic derivatives, fully coupled mission problems are tractable. Falck et al. [1] specifically examined how path-dependent thermal constraints affect the optimal mission profile of an electric

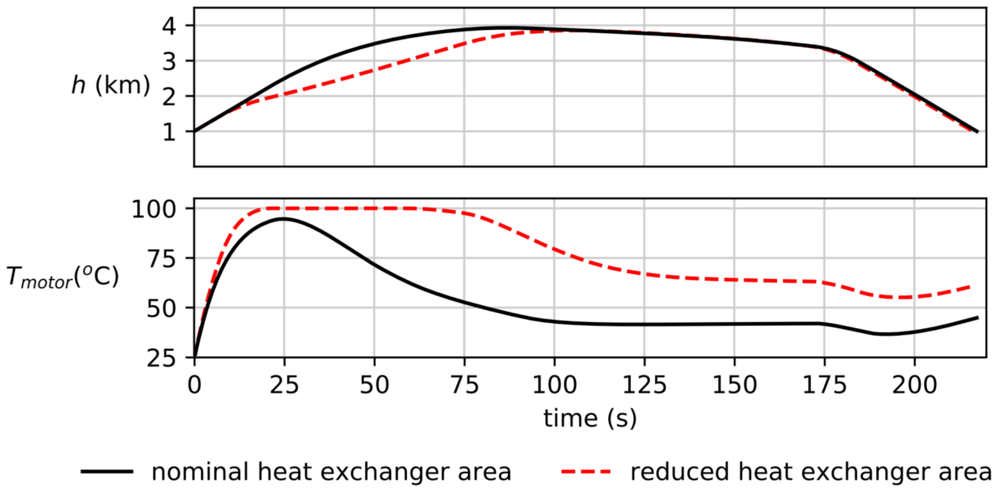


Figure 1.2: Comparison between optimal flight profiles for a thermally constrained electric aircraft with two differently sized heat exchangers from Falck et al. [1].

aircraft, an application case that necessitates the use of fully coupled mission and design optimization. A comparison plot of the optimal flight profiles from that work is shown in Fig. 1.2. Although individual disciplinary analyses could be performed to evaluate the aircraft performance, a fully coupled model more accurately resolves the interdisciplinary trade-offs between the thermal constraints and heat generation.

At the same time that AFRL and NASA were expanding their mission optimization capabilities, the University of Michigan MDO Lab was approaching similar problems from multiple different angles [51–55]. Liem et al. [51, 56, 57] sought to maximize aircraft performance by considering multiple representative missions with different payloads and ranges. They considered the aerostructural design of the wing across missions obtained from actual flight data to get more representative results than conventional multipoint design. Kao et al. [52, 53] developed a modular adjoint-based approach to optimize the mission profile for a commercial airliner for multiple missions simultaneously. They used surrogate models for aerodynamic and propulsion data to make the mission optimization tractable. Burdette and Martins [58] considered the effect that a morphing trailing edge has on mission performance using high-fidelity aerostructural optimization. They created the surrogate using training points from the Mach-C_L space based on where the aircraft would fly during the fixed mission.

Brelje and Martins [59] developed an aircraft conceptual design tool, OpenConcept. This tool models many subsystems within the aircraft, including the propulsion, electrical, and thermal systems, and can be used to optimize the states of the aircraft across its mission [59]. Simultaneously, Chauhan and Martins [60] developed a method to optimize electric vertical takeoff and landing (eVTOL) takeoff trajectories. They presented multiple optimal trajectories for a tandem tiltwing aircraft considering many different path constraints and flight assumptions. Most recently, Hwang et al. [55] performed fully coupled allocation-mission-design of a commercial airliner. This work included Reynolds-averaged Navier–Stokes (RANS)-based computational fluid dynamics (CFD) in the loop, 128 missions, and a ticket-selling economics model to maximize the profit for a given airline considering a new aircraft design.

Beyond work dedicated to solving path-dependent problems, many methods developed for large-scale design optimization are used in this dissertation. Developments from the MDO community provide the foundation for some of these methods, including the ADjoint approach [61–64], different MDO strategies [65, 66], and modular analysis and unified derivatives (MAUD) [67]. Because of these influences, this dissertation approaches optimization problems from more of an MDO mindset than an optimal control theory mindset.

We use theory and methods from a subset of these presented works and other papers that will be discussed in later chapters. Much of the work presented here uses tools developed together by the MDO Lab and NASA.

1.3 Dissertation objectives

We want to solve large-scale path-dependent aircraft optimization problems to better design aircraft accounting for interaction between relevant systems. This will allow designers to better understand realistic trade-offs between coupled aircraft systems earlier in the design process. The overarching contribution of this dissertation is the construction and exploration of a fully-coupled aero-thermal-propulsive-mission model to optimize aircraft considering their path-dependent performance.

I first construct a general toolset that can model and optimize the performance of an aircraft across a full mission. Our first application case involves path-dependent optimization of commercial airliners because of their relatively simple mission profiles. Then I examine the performance of a supersonic aircraft performing minimum time-to-climb missions, for which we have many benchmarks from the literature. In this process, I extend the capabilities of engine modeling and delve into design studies of complex engine models and thermal-propulsive coupling. I then focus on the thermal-mission problem before discussing the full aero-thermal-propulsion-mission problem. A notional extended design structure matrix (XDSM) diagram [68] of the fully coupled optimization problem is shown in Fig. 1.3.

This dissertation strives to show that it is possible to use gradient-based optimization to solve large-scale path-dependent MDO problems. Through this process, I identify relevant challenges, as well as propose, implement, and verify solutions to these challenges. Specific technical challenges will be addressed within the relevant chapters. Lastly, all of the tools used here are open-source, and the majority of the models used are available in the public domain, which allows other researchers to more easily use and expand upon this work.

In completing this work, I encountered three broad challenges:

Challenge 1: Complex aircraft systems are difficult to optimize due to their large design space.

Challenge 2: Performing MDO of aircraft accounting for their path-dependent states is challenging due to the computational cost and model complexity.

Challenge 3: Computational engine design must be done incrementally and carefully to

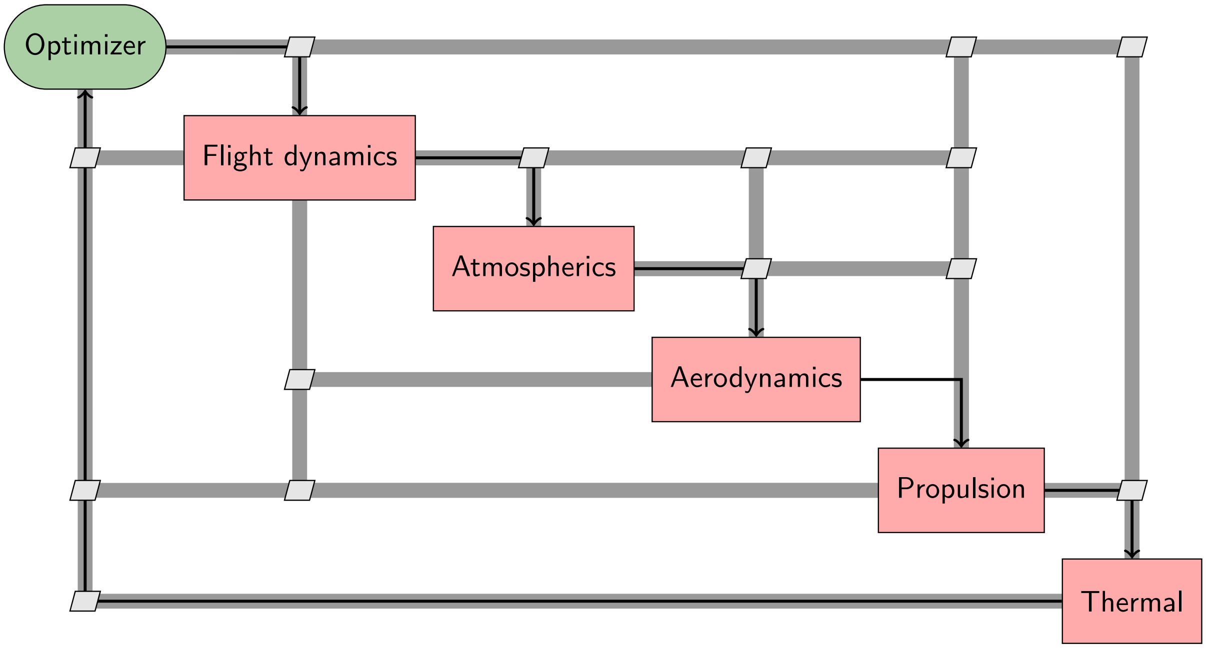


Figure 1.3: Notional XDSM diagram [68] of the fully coupled optimization problem presented in Chapter 5.

ensure the models converge.

In this dissertation, I address all three of these challenges by both implementing solutions that exist elsewhere in the literature as well as developing my own novel solutions.

To minimize the effects of Challenge 1, I use gradient-based optimization and choose to pay the high implementation cost of obtaining analytic derivatives to enable efficient large-scale optimization. Over the past three decades, the MDO community has advanced methods to optimize coupled analysis models [65, 67, 69, 70], and the use of gradient-based optimizers with efficiently-computed derivatives is paramount. For each model introduced in this dissertation, I compute the analytic derivatives to enable efficient gradient-based optimization.

Challenge 2 is partially addressed through the use of pseudospectral collocation schemes, which decrease the total cost of path-dependent design optimization by posing the time integration problem as a subset of the optimization problem [71–74]. I also take advantage of the sparse nature of the multidisciplinary problem to decrease the cost of computing the derivatives needed for gradient-based optimization.

Challenge 3 summarizes a well-known detail of engine modeling: it is challenging to create robust models [37, 75, 76]. To combat this, I construct a novel surrogate-enhanced guess table to enable the engine model to converge across a wide flight envelope across different designs. This allows us to query the engine model at any point in the mission profile without needing to incrementally step up to analysis points from a previously-converged point.

Although there were many more challenges encountered during this work, these are the three that I have deemed most impactful. In each chapter, I discuss these challenges and my solutions in detail, clearly demarcating which solutions I developed as part of this work. My specific contributions to the field as a result of this dissertation and their impact are detailed in Sec. 6.2.

1.4 Computational tools used

1.4.1 Optimization framework: OpenMDAO

Throughout this work, we use OpenMDAO[1] [77] as the underlying optimization framework. OpenMDAO was developed at NASA Glenn and uses the MAUD theory to allow for modular construction and execution of complicated models [67]. OpenMDAO has been used to optimize a huge variety of problems, including wind turbines [78–80], boundary layer ingestion aircraft [81–84], thermodynamic engine cycles [85–88], and MDO of aircraft considering mission performance [47, 89, 90].

OpenMDAO was designed from the ground-up with gradient-based optimization in mind. Because of this, a lot of the challenging aspects associated with propagating derivatives through a complicated model are handled behind-the-scenes by OpenMDAO without user input. Specifically, if a user is creating a model that consists of multiple analysis blocks, or components, and wants to connect them into a group to perform gradient-based optimization, the user only needs to provide partial derivatives for the individual analysis blocks, not total derivatives for the entire model. OpenMDAO uses a generalized version of the chain rule to compute the total derivatives based on the partial derivatives computed from each analysis block. Additionally, if the user is prototyping a new model or does not want to provide derivatives, they can tell OpenMDAO to finite-difference or complex-step [91] across the component to obtain the derivatives. This kind of heterogeneous model and derivative construction allows users to create complicated models that are suitable for gradient-based optimization relatively quickly.

Many features of OpenMDAO directly enabled the research presented in this dissertation. Model construction was greatly simplified by the ability to connect analysis blocks easily from different tools. Tools for model layout visualization and debugging that come packaged with OpenMDAO decreased the time to set up and verify the problem. Solver convergence informa-

---

[1] https://github.com/openmdao/openmdao

tion and model debugging tools helped us determine what to fix when the model did not converge successfully. Additionally, multiple open-source tools have been written using OpenMDAO and were directly used in this work, including Dymos[2] [46] for mission optimization and pyCycle[3] [87] for engine design. These tools and features make OpenMDAO a useful and necessary tool for setting up, analyzing, and optimizing multidisciplinary systems efficiently within this dissertation. Dymos and pyCycle are detailed in Sec. 2.2.3 and Sec. 4.2 respectively.

For all optimizations presented in this dissertation, we use pyOptSparse [92] and SNOPT [93]. pyOptSparse[4] is an open-source Python framework that provides a common interface for many gradient-based and gradient-free optimization methods. SNOPT is a sequential quadratic programming approach that efficiently solves large sparse nonlinear constrained optimization problems [93]. OpenMDAO has native support for pyOptSparse and any of its wrapped optimization methods.

1.4.2 Surrogate modeling

When we have data we want to interpolate or approximate, we can use surrogate models to create a low-cost alternative model. The regularized minimal-energy tensor-product spline (RMTS) interpolant [94] implemented in Surrogate Modeling Toolbox (SMT) [95][5] is used throughout this dissertation. We employ surrogate models to provide good initial conditions to complicated solver hierarchies and to approximate disciplinary models to reduce computational cost by removing the physics-based analysis from the optimization loop.

As detailed in Sec. 4.4.2, this dissertation presents a surrogate-enhanced guess table approach that creates a more robust engine model. Surrogate models provide the initial guesses for the complex hierarchy of solvers within the engine model, which help to converge the model from

---

[2] https://github.com/openmdao/dymos

[3] https://github.com/openmdao/pycycle

[4] https://github.com/mdolab/pyoptsparse

[5] https://github.com/SMTorg/smt

a variety of flight conditions more easily than would otherwise be possible. Normally, engine designers have to incrementally change the engine design and converge the model in an iterative process, but the use of surrogate models removes the need for that process.

The other main use for surrogate models in this dissertation is to approximate disciplinary analyses to remove expensive computational models from the optimization loop. An example of this is performing CFD on an aerodynamic model at a wide range of flight conditions, then using the resulting data to create a surrogate model that is queried during the mission optimization. This is only possible when we do not change the design of the disciplinary model during the optimization process. It is possible to design disciplinary systems and still take advantage of surrogate models to reduce computational cost by retraining the surrogate model at each optimization iteration, as detailed by Liem et al. [51] and later used by Hwang et al. [55], though that method is not employed in this dissertation.

1.5 Overview of models

Throughout this dissertation, we use application cases to demonstrate the coupled interactions we aim to study. Some of these applications are based on existing models from the literature and are introduced in this section. Each chapter will go into more detail about which parts of the models are used and how they are used, but we introduce these applications here since their use spans multiple chapters.

1.5.1 Common Research Model

In some studies, we use a wing model based on the uCRM, a large-scale transport aircraft wing developed by Brooks et al. [96]. Figure 1.4 shows the Boeing 777 wing-body next to the uCRM wing-body. The uCRM is based on the Common Research Model (CRM) developed by Vassberg et al. [97] and was adapted for use in aerostructural analyses and optimizations [96]. They did this

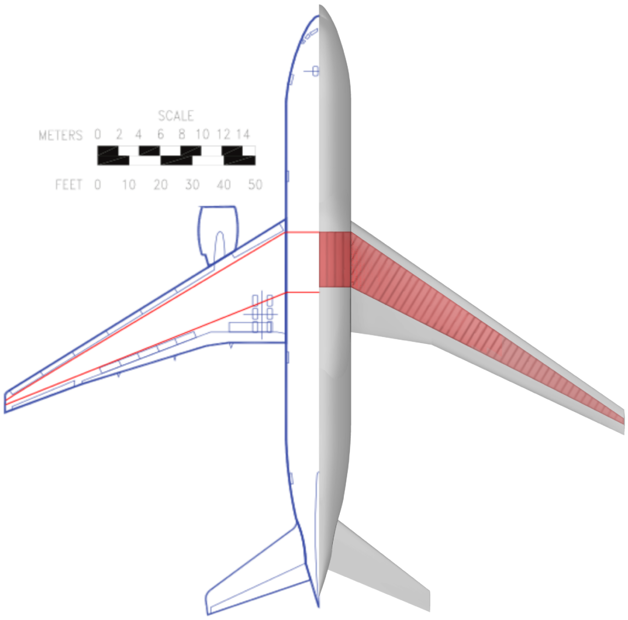


Figure 1.4: Boeing 777 (left) and uCRM-9 (right). The uCRM has a slightly lower wing area and span, and more sweep than the Boeing 777 [96].

by back-calculating the jig shape of the CRM wing and producing an aerostructural model.

The CRM was originally developed as a benchmark configuration for aerodynamic analysis to validate CFD. It has been used extensively to compare simulation methods against each other and against wind tunnel tests [98]. The CRM model is also used in the Drag Prediction Workshop, where multiple institutions and companies run the same analysis cases to compare results from different CFD solvers. Because of this, a large trove of information is available on the performance of the CRM, which makes it an excellent candidate for investigating new analysis and optimization methods. We use this wing model for commercial aircraft mission optimization because it is well-studied across multiple levels of fidelity and experiments.

1.5.2 Efficient Supersonic Air Vehicle

The ESAV aircraft provides a common research model for an advanced military fighter aircraft configuration, and has been studied before using MDO with a variety of disciplines, including

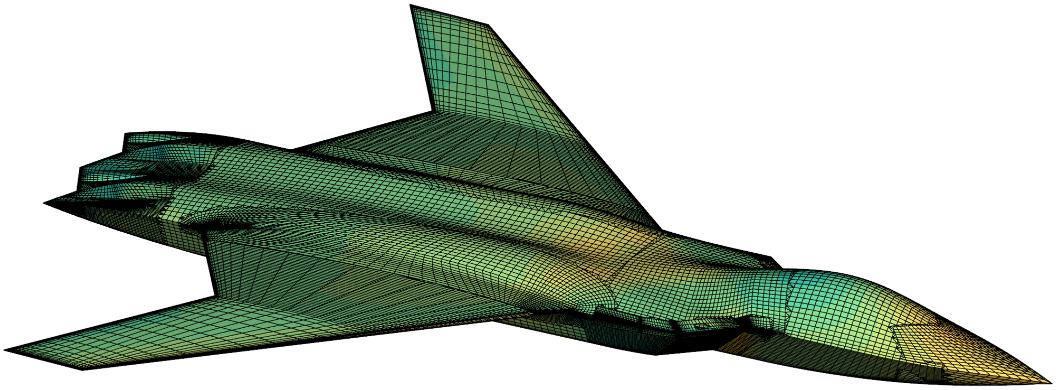


Figure 1.5: ESAV CFD model used for supersonic aircraft aerodynamic data

aerodynamics, structures, stability, mission, noise, and emissions [99–102]. To obtain the aerodynamic properties of the aircraft, we built a RANS CFD model of the aircraft, which we evaluated at a range of different flight condition to generate an aerodynamic surrogate model. The ESAV mesh and nominal CFD solution are shown in Fig. 1.5. In this work, the shape of the aircraft was held fixed, so we do not need to perform physics-based analysis in the loop to update the aerodynamic properties. This greatly reduces the cost of running the optimization.

We began with a CAD surface definition of the aircraft as provided by Lockheed Martin. We plugged the engine inlet and nozzle and removed control surface gaps to simplify the model. These simplifications are possible in this context because we are only considering the aerodynamic properties of the vehicle at different flight conditions without assessing the effects of control surfaces or coupled aeropropulsive effects.

1.5.3 Supersonic engine model

For modeling the propulsion systems in the ESAV aircraft, we constructed a generic supersonic engine model in pyCycle. pyCycle is an open-source engine modeling tool developed by NASA that is detailed in Sec. 4.2. Our engine model was based on an existing Numerical Propulsion

System Simulation (NPSS) model of the GE F110 engine obtained from NASA. We adapted the NPSS model for use in pyCycle by translating its architecture and element maps.

Throughout this dissertation, we use a few variations of this engine for different studies. Although the base version of this engine model is a mixed-flow turbofan, we introduce installation effects, a heat exchanger in the bypass duct, and examine its split-flow performance as well. In each section where we use a variant of this engine model, we first introduce what features are used for that particular study.

1.6 Dissertation outline

This chapter provides a high-level overview of the foundation of this work, details on the novel contributions, and information about the models used in this dissertation. We present the fine-grain technical details in each relevant chapter. The literature review in this chapter serves as a general and broad introduction to path-dependent optimization. Each subsequent chapter has a smaller and more focused review of the literature relevant for that chapter. Chapters 2, 3, and 4 go into detail about individual disciplinary models, analysis, and optimization, and Chapter 5 presents results from the fully coupled system.

Chapter 2 focuses on mission optimization and more broadly, on the integration of path-dependent states. We examine mission optimization of a commercial airliner and a supersonic aircraft. Chapter 3 details the thermal systems we consider and discusses the construction of a modular toolset to model general thermal systems. We present results for simple optimization cases, keeping in mind these thermal systems will be integrated into a larger mission optimization problem. Chapter 4 covers engine modeling and optimization, including improvements we made to the process as a result of this dissertation. Specifically, we introduce installation effects and more robust convergence capabilities before presenting multiple engine design optimization problems.

Combining all of these disciplines and models, we examine the full aero-thermal-propulsive-mission optimization problem in Chapter 5. We build up to the full optimization problem by examining subsets of the problem first, including thermal-mission and thermal-engine optimization. The full optimization problem is then constructed, solved, and discussed. Lastly, Chapter 6 summarizes the work presented in this dissertation, details my novel contributions, and suggests future avenues to study in path-dependent design optimization.

CHAPTER 2

Mission Optimization

2.1 Review of mission optimization

The most realistic and robust way to optimize an aircraft is to consider its performance along its entire mission trajectory, as shown in previous work [52, 57, 103–105]. By doing this, we are able to track any path-dependent states of the aircraft across its mission, allowing us to better evaluate the performance. However, integrating across the aircraft's mission generally requires a large number of model evaluations. We want to be able to generally analyze and optimize mission profiles of aircraft regardless of their size, propulsion systems, or use cases. We will now detail previous work in mission analysis and optimization with a focus on methods that incorporate physics-based analyses within the optimization, also known as physics-in-the-loop methods.

Trajectory analysis and optimization has long been studied in the field of optimal control theory, with automotive, aerospace, naval, and robotics applications [16, 71, 72, 106–108]. A survey paper by Betts [107] is an exhaustive primer for the state-of-the-art on trajectory optimization through 1998, with a digestible summary and understandable introduction to the foundational mathematics. Since then, Waschl et al. [106] have published a book on optimal control in automotive systems. Although not focused on aerospace systems, the fundamental theories and solutions presented are relevant to this dissertation.

Many papers present collocation methods as an efficient method for optimal control problems where the analyses are computationally costly [71–74]. These methods, a subset of direct integration methods, are used throughout this dissertation and are detailed in Sec. 2.2. Formulated differently from these direct methods, indirect methods are sometimes a better option for some trajectory optimization problems, but we are not examining them here. Indirect methods have a smaller region of convergence, meaning that they usually require a better initial guess, which may lead to convergence issues within the optimization problem [107].

A large body of work exists on spacecraft trajectory optimization [71, 109–111]. Specifically, we may be interested in low-thrust trajectories for orbiting spacecraft [111], or trajectories of multistage rockets exiting Earth's atmosphere [112]. Although the trajectories and associated problems in spacecraft applications are markedly different than aircraft trajectories, the methods used there are directly applicable to aircraft mission optimization.

New aircraft technologies have made mission analysis and optimization a more integral part of aircraft design. Hybrid electric propulsion systems [113], morphing wings [105, 114–116], and active load alleviation [117, 118] all directly couple the physical design of the aircraft with its mission profile. Additionally, operational concepts such as formation flying [119, 120], real-time weather avoidance routing [121], and continuous descent profiles [122] drive the need for path-dependent modeling capabilities. Looking towards the future, the advent of urban air mobility will create huge path-dependent optimization problems in terms of routing [123, 124] and airspace integration with existing aircraft [125].

The literature reviews presented here and in Chapter 1 provide the necessary context and foundation for the work contained in this chapter. We will now explain the mission optimization methodology used throughout this dissertation, mission problem formulations, and the tools used to solve these problems. We close this chapter by presenting results for two application cases: design and mission optimization of a commercial airliner with morphing wings and mission optimization of a supersonic military aircraft.

2.2 Mission optimization methodology

2.2.1 ODE construction

To discuss how path-dependent problems can be solved, we need to develop the equations that dictate the time-varying properties of the system. These equations called ODEs are used to model the behavior of any subsystem across time. For physical objects moving through space, the most common ODEs are the equations of motion. Depending on the application, relevant terms within these ODEs can include effects from gravity, engine thrust, wing lift and drag, and electromagnetic forces. To obtain the thermal states of the aircraft, we need ODEs that track component temperature, coolant flow and temperature, and air flow through heat exchangers. We need an ODE for any time-varying state that we are interested in monitoring throughout the optimization process. In this chapter, we focus on the equations of motion that dictate how aircraft move through space.

The unsteady aircraft equations of motion for 2D planar flight are

$$m\dot{V} = T\cos\alpha - D - W\sin\gamma, \tag{2.1}$$

$$mV\dot{\gamma} = L + T\sin\alpha - W\cos\gamma, \tag{2.2}$$

$$\dot{h} = V\sin\gamma, \tag{2.3}$$

$$\dot{x} = V\cos\gamma, \tag{2.4}$$

where L, D, T, and W are the forces of lift, drag, thrust, and weight respectively, m is the mass of the aircraft, V is the aircraft velocity, γ is the flight-path angle, h is the altitude, x is the horizontal distance, and α is the angle of attack. Figure 2.1 shows how these forces are oriented relative to an aircraft in flight.

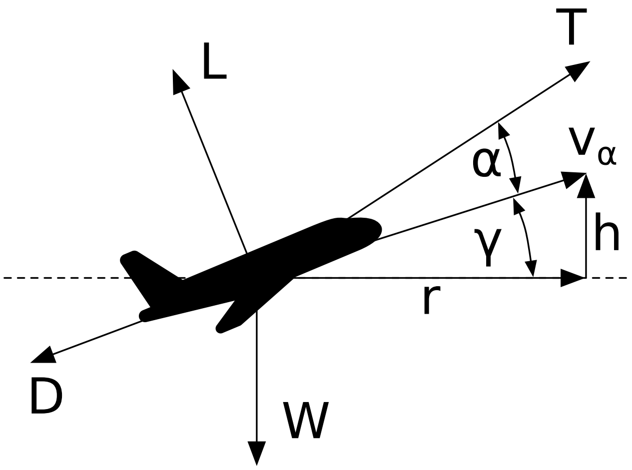


Figure 2.1: Free-body diagram for the forces acting on an aircraft used in the equations of motions.

For some studies, a quasi-steady approximation can be used, leading to

$$0 \cong T \cos \alpha - D - W \sin \gamma, \tag{2.5}$$

$$0 \cong L + T \sin \alpha - W \cos \gamma. \tag{2.6}$$

Subsonic aircraft can be modeled using these quasi-steady equations because their kinetic energy is almost negligible compared to their potential energy [17]. For supersonic aircraft, their acceleration capability and higher speeds necessitate the use of the unsteady equations of motion.

Using the energy-state approximations as described by Bryson et al. [17], these quasi-steady equations of motion can be further reduced to a simpler dynamic system. Their derivation is not reproduced here, though the final result is

$$h = (E - \frac{1}{2}V^2)/g. \tag{2.7}$$

Through this approximation, the only state variable is the energy E and V can be considered a control variable. Through this energy-state approximation, it is possible for the aircraft to move along constant E contours instantaneously, which unrealistically models rapid dives and climbs. The simplicity of the equations of motion for the energy-state approximation made it popular in early mission optimization, and it serves as a reasonable approximation for many problems.

Throughout the remainder of this dissertation, we use the unsteady equations of motion to better capture the actual aircraft physics. When first implementing mission optimization problems or developing new applications, the quasi-steady and energy-state approximations are useful to reduce problem complexity. To better understand the trends in mission optimization using a more stable and less costly method, the energy approximation was used for some of the studies precluding those presented in this dissertation.

Depending on how we choose to solve for these equations of motion, we can select which parameters are control variables, states, or solved for via an internal solver. For example, if $\dot{V}$ and $\dot{\gamma}$ are control variables, then h, x, V, and γ are all states that will be solved with an ODE integrator. Alternatively, α and T could be control variables, which again leaves h, x, V, and γ as states. The flexibility to choose which variables are controls and which are solved for internally comes from the use of nonlinear solvers within the ODE. OpenMDAO enables the use of nonlinear solvers at any level in the model hierarchy.

The *differential inclusion* approach introduced by Seywald [126] is another application of nonlinear solvers within mission optimization. Through this method, the state time-histories are treated as "truth" and the dynamic system equations are solved for the corresponding control time-history. This means that the trajectory is feasible when the state time-history is within the allowable control variable bounds. When formulating the integration problem without differential inclusion, the trajectory is not necessarily feasible during the optimization process, and is only guaranteed to be feasible at the fully-converged optimum.

There are both benefits and drawbacks to using differential inclusion. Benefits include that each optimization iteration produces a feasible mission because the operational constraints are satisfied using internal solvers instead of the optimizer. This also reduces the optimization problem size by moving constraints to the solver level instead of satisfying them the optimizer level. However, that means that one drawback is that each optimization iteration will be more costly to analyze due to the need to resolve constraints at the solver level. Usually this introduces the need for an additional problem-level solver.

In this dissertation, we perform mission optimization both with and without using the differential inclusion approach. Which approach is used depends on the application case and the robustness of the dynamics model.

2.2.2 Integration techniques

Once we have the ODEs that describe the dynamic system, we need to integrate them across time to evaluate aircraft performance. There are many integration techniques; some are better suited than others depending on the problem context. Hwang and Munster [127] compared the accuracy and speed of a large number of integration techniques. For time integration within an MDO context, we want to use methods that use efficient derivative information to reduce the computational cost. Some previous work in mission optimization used explicit integration schemes, but implicit collocation schemes have a significant computational advantage in the context of large-scale gradient-based MDO [46, 71, 128, 129], so that is where we focus for this dissertation.

Implicit collocation schemes are well-suited for gradient-based optimization for two main reasons [46]. First, they pose optimal control problems in a way that the number of calculations performed each iteration is constant. This is important so the optimization problem size does not change during the course of solving the optimization problem. Secondly, these collocation schemes produce sparse Jacobian structures, which means that large-scale problems can be solved efficiently by reducing computational cost. OpenMDAO is designed to take advantage of this sparsity, especially when computing the total derivatives of the problem.

The two implicit collocation methods that we study here are the Legendre-Gauss-Lobatto (LGL) and Legendre-Gauss-Radau (LGR) schemes. There are many intricacies concerning the theory and implementation of these methods, though we focus on how they can be used for large-scale MDO, which has also been examined by Falck and Gray [46]. Simply, both methods use a series of interpolative polynomials to represent the states and controls within a problem, then solve for the values of the states until the physics of the dynamic system are respected.

Originally developed by Herman and Conway [73], the high-order LGL method is a general extension of Hermite-Simpson collocation to higher orders [71]. A sample interpolative polynomial with the relevant nodes is shown in Fig. 2.2. The states and state-rates are interpolated to the collocation nodes, and then the state dynamics are again evaluated at the collocation nodes. The actual state dynamics are then compared to the interpolated state dynamics, and this difference between the values (the collocation defects) are driven to 0 by changing the state values at the discretization nodes. Because the optimizer controls the state values at the discretization nodes, this method moves the duty of converging the state time-history to the optimizer level.

The LGR method is similar to the LGL method, except the states and controls are discretized at all nodes. Figure 2.3 shows the interpolative polynomial and nodes for the LGR method. Here, the state defects are calculated at all nodes except the last in the phase, and the collocation nodes are just a subset of the state discretization nodes.

In terms of performance and robustness, the differences between LGL and LGR methods are subtle [46]. The LGL method may not perform well if given a poor initial guess and the LGR method results in path constraints that can be accomplished with bounds on design variables. The LGL method has fewer design variables and constraints than the LGR method, but requires multiple evaluations of the ODE, which may become costly if there is significant overhead associated with evaluation. Both methods depend on a relatively good initial guess, especially in complex MDO system design.

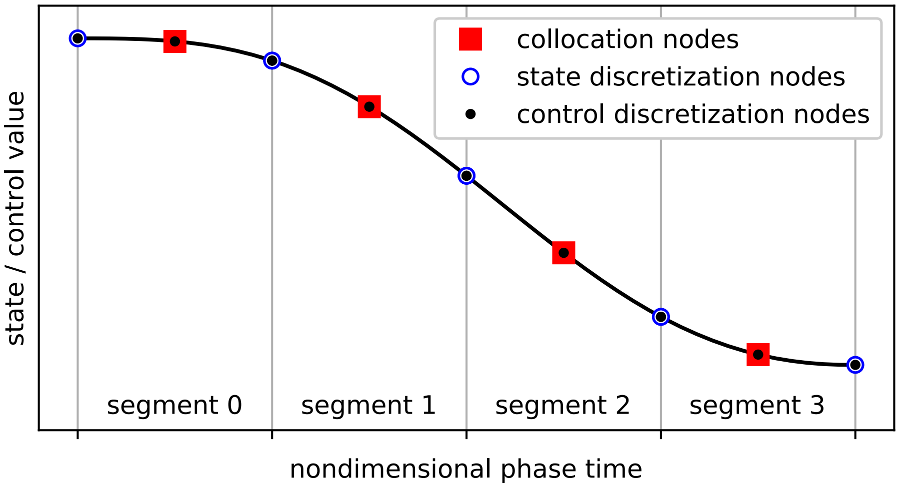


Figure 2.2: An example of the state interpolating polynomials for the LGL transcription, adapted from Falck and Gray [46].

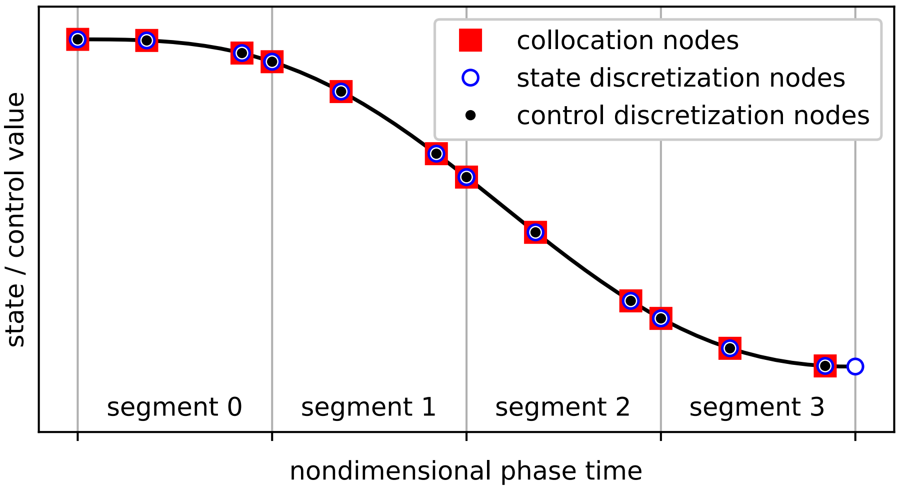


Figure 2.3: An example of the state interpolating polynomials for the LGR transcription, adapted from Falck and Gray [46].

Table 2.1: General form of the optimal control problem solved by Dymos, adapted from Falck and Gray [46]

Minimize	$f_{obj}(x, t, u, d)$
With respect to:	
State dynamics	$\dot{x} = f_{ODE}(x, t, u, d)$
Time	$t_{lb} \leq t \leq t_{ub}$
State variables	$x_{lb} \leq x \leq x_{ub}$
Dynamics controls	$u_{lb} \leq u \leq u_{ub}$
Design parameters	$d_{lb} \leq d \leq d_{ub}$
Subject to:	
Boundary constraints	$g_{lb} \leq p(x, t, u, d) \leq g_{ub}$
Path constraints	$p_{lb} \leq p(x, t, u, d) \leq p_{ub}$

2.2.3 Mission integration tool: Dymos

Throughout this dissertation, we use Dymos to solve the path-dependent MDO problems. Dymos [46] is an open-source tool built upon OpenMDAO and provides an interface for integration of ODEs with MDO in mind. Portions of Dymos' implementation is based on NASA's OTIS optimal control software [71]. Dymos contains both high-order LGL, LGR, and Runge–Kutta schemes and provides convenient user-facing methods to solve optimal control and design problems. Dymos has been used to optimize aircraft trajectories in parallel [130], aircraft trajectories with propulsion in the loop [44], and next generation aircraft concepts [1, 47, 131].

Dymos breaks a trajectory into one or more *phases*, and then each of those phases into *segments*. These segments are modeled as a polynomial and values of the states and controls are considered at *nodes* across these segments. In OpenMDAO terms, each phase is a group, and each trajectory is a group of phases. Within a trajectory, time, states, and controls can be linked between phases to ensure continuity. Boundary or path constraints can be used on any output from the model, regardless if it is a state or not. Table 2.1 shows the general form of the optimal control problem solved by Dymos. Many more capabilities about Dymos are detailed by Falck and Gray [46]. Dymos has evolved from collaborative efforts at NASA Glenn, and portions of its codebase have previously been presented under the names *Pointer* [1, 44] and *Ozone* [127].

2.3 Design-mission optimization of a morphing wing aircraft

2.3.1 Overview

As a first application case for mission optimization, we perform design and mission optimization of a morphing wing aircraft [90]. We directly couple the trajectory and aerostructural wing analysis to capture the physics-based performance directly in the mission simulation. Using high-fidelity methods for aerostructural morphing wing and trajectory optimization is intractable because of the high computational cost to analyze the complex system. We must use lower-fidelity models or surrogate models to reduce the design space so we can find an optimal design in a suitable timeframe.

In this section we analyze and optimize the nominal design, morphing inputs, and altitude profile over the mission of the CRM-based configuration. To solve the coupled aerostructural system, we use a vortex lattice method (VLM) for the aerodynamic analysis and a 6-DOF finite element analysis (FEA) model. Additionally, we use a surrogate model for the propulsion analysis and a collocation-based approach for the coupled trajectory-propulsion system. Then we optimize the design and trajectory of an aircraft with a morphing wing and compare the performance against a conventional aircraft.

This section aims to quantify the added benefits of morphing technology by performing fully coupled mission-design-morphing optimizations.

2.3.1.1 Aerostructural design tool: OpenAeroStruct

We can efficiently analyze the coupled aerostructural system using inexpensive physics-based models. VLM and 1-D FEA models can accurately capture the coupling between the aerodynamic and structural disciplines. A coupled 1-D VLM-FEA model enables optimization of the spanwise distributions of the aerodynamic twist, chord, and structural thickness. We have implemented this model in the OpenAeroStruct software package [132]. It is an open-source aerostructural optimization tool coupling an extended VLM model with FEA using 6-DOF spatial beam elements that support axial, bending, and torsional loads. OpenAeroStruct is written entirely in Python on top of OpenMDAO [77]. Fully analytic derivatives are provided for each component in the aerostructural optimization problem.

Although OpenAeroStruct does not model some of the physics actually present in aerostructural wing design, it generally captures the same trends as high-fidelity methods. For example, compressibility effects, wave drag, and flow separation are not directly modeled here, but we use empirical corrections to account for the absence of these effects. The structural system is modeled as an equivalent representative circular spar, which simplifies the moments of inertia about the elastic axis as radially symmetric. Since this work was completed, OpenAeroStruct now includes the ability to model wave drag and use a wingbox cross-section for the structural spar [133]. OpenAeroStruct's relatively low cost allows us to use it in this fully coupled optimization while providing adequate resolution of the physics of the system.

OpenAeroStruct has proven useful beyond this application case, both in research and the classroom. Many more details on the theory, implementation, and applications of OpenAeroStruct are presented in Appendix A. Recently, researchers at NASA Glenn Research Center [47] are using OpenAeroStruct in their MDO studies of hybrid-electric commuter aircraft to study path-dependent problems similar to what is presented in this dissertation.

2.3.1.2 Selection of morphing type

Wing morphing allows a single aircraft to perform optimally in different scenarios. The term "morphing" is used to describe vehicle adaptability of many types, including conformal control surfaces and camber variations, but also large changes in span, wing area, and chord [134]. The benefits of many different types of wing morphing have been studied in detail through simulations, wind tunnel, prototypes, and full-scale aircraft [135, 136]. To justify adding morphing technology to an aircraft, a designer must consider if the performance benefits outweigh the increased weight, system complexity, and program costs. Specifically, some engineering challenges that must be addressed are distributed high-power density actuation concepts, structural mechanization concepts, and flexible skins [134].

We examine twist morphing, where the aerodynamic twist at sections along the span can vary during flight while the airfoil profile remains unchanged. Twist morphing can be achieved by servos, shape memory actuators, piezoelectric actuators, or hydraulic actuators [136]. Depending on how the aircraft is configured, the twist morphing capability can have a minimal impact on the internal structural components of the wing. For example, if the main structural spar in the wing is at the quarter-chord and has a circular cross-section, the wing twist can more simply be actuated around this spar. Vos et al. [137] achieved twist morphing in a prototype wing using threaded rods to independently rotate four ribs around a circular main spar. Twist morphing is primarily examined for use in micro-air and small-scale vehicles, though it could be used in any scale of aircraft.

2.3.1.3 Mission optimization methodology

Do we need to explicitly evaluate the aerostructural performance within the mission analysis in a fully coupled manner, or can we use surrogate models and obtain reasonably accurate results? For applications and missions that are not path-dependent, we could expect that the surrogate models could reasonably resolve the design-mission trade-offs with reduced computational expense. However, these surrogate models are trained with data that does not consider the aircraft's previous state in the mission profile. Because of this, a surrogate-based approach cannot accurately resolve path-dependent effects like thermal constraints, actuator motion limits, or discontinuous aircraft weight resulting for ordinance delivery, for example. These applications require the fully coupled design-mission approach presented in this section.

In this section, we describe the methods and algorithms in the multidisciplinary model that incorporates mission analysis, aerostructural analysis, and a propulsion surrogate model. We discretize the mission profile, enforcing the horizontal and vertical equations of motion and integrating an ODE to solve for the fuel burn profile. At each mission point, the aerodynamic and structural responses are computed, and the propulsion characteristics are predicted using a surrogate model. In the subsections that follow, we describe the mission equations, the integration of the resulting ODE, and the solution algorithm for the coupled system.

We use the unsteady equations of motion detailed in Sec. 2.2.1 and these equations are discretized and enforced at each mission point. At the same time, we integrate the ODE for fuel weight,

$$\dot{W}_f = -T \cdot SFC \tag{2.8}$$

where W_f is the fuel weight and SFC is thrust-specific fuel consumption. Both thrust and SFC are functions of W_f. We use the solver-based approach with the 6th-order Gauss–Legendre integration scheme.

2.3.1.4 Coupled solution

The mission equations have a coupling loop, as we can see in a model visualization in Fig. 2.4. The figure shows the hierarchy tree for the model and the dependency graph for the constituent components and variables. The ‘sys_coupled_analysis’ group contains the coupling loop: the vertical equation of motion computes lift from weight, the OpenAeroStruct group computes drag from lift, the horizontal equation of motion computes thrust from drag, then we compute throttle from thrust, then thrust-specific fuel consumption from throttle, and finally we integrate the fuel burn ODE. The integration produces fuel weight, from which we obtain the total aircraft weight, and the loop closes.

The aerostructural analysis using the VLM and FEA methods is one of the blocks in the coupling loop in ‘sys_coupled_analysis’. We can view this block as a mapping from lift to drag; however, the aerostructural model computes both lift and drag as outputs as a function of angle

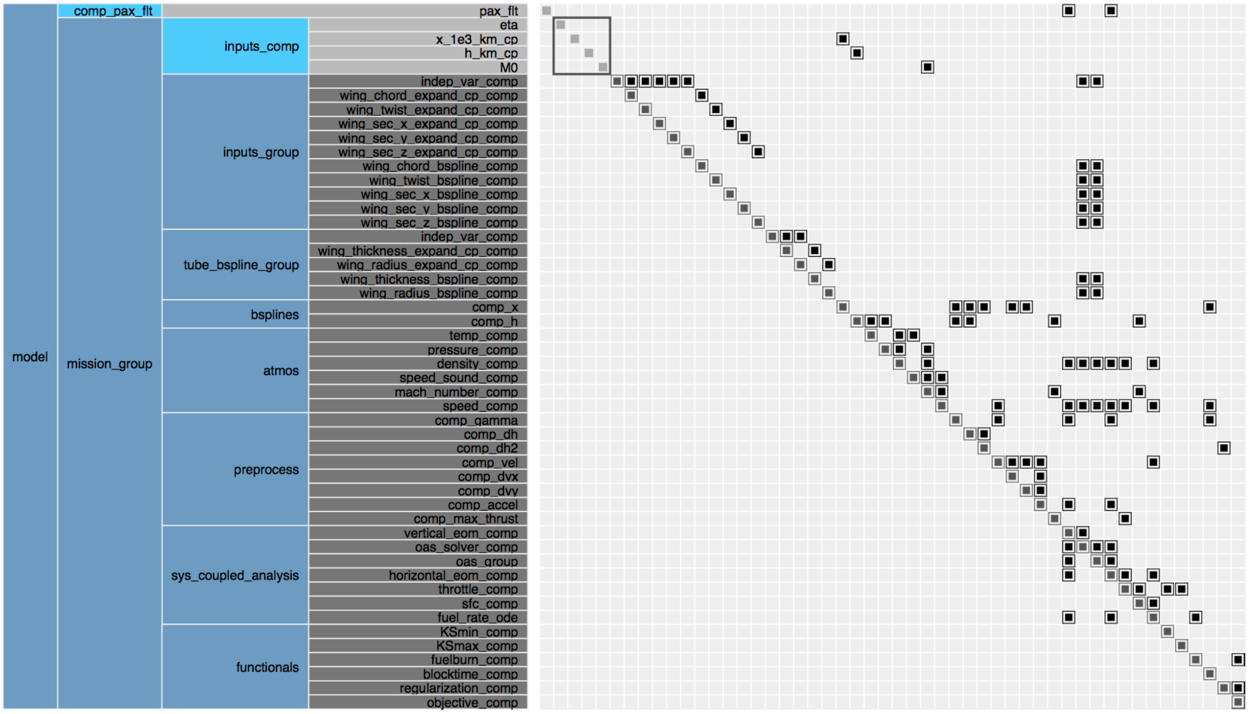


Figure 2.4: Hierarchy tree and dependency graph for the overall model. The 'sys_coupled_analysis' group contains the feedback coupling relating the fuel burn rate and the weight of the aircraft. This directly couples the aerostructural wing properties to the changing weight of the aircraft across the mission profile.

of attack. Since robustness and efficiency are both required, we use a bracketed root finding algorithm. A bracket is almost always successfully found by choosing a large negative and a large positive angle of attack since we do not model stall.

The propulsion surrogate model uses the regularized minimal-energy tensor-product spline interpolant [94]. The inputs are throttle setting, Mach number, and altitude, and the outputs are thrust and thrust-specific fuel consumption. The data is generated from the NPSS software [138] for a Boeing 777-sized engine. Thrust is linear in throttle setting for this data, so the throttle is computed in 'throttle_comp' by evaluating the maximum thrust for the Mach number and altitude of interest and normalizing the current thrust by the maximum thrust.

2.3.2 Problem formulation

In this subsection, we describe the three types of optimization problems that we solve: fixed-design, static-design, direct morphing. The objective function, design variables, and constraint functions for each problem are presented in Table 2.2. Fixed-design optimization is the baseline problem where only the mission is optimized, and static-design optimization optimizes the mission and the wing design without considering morphing. The direct morphing case simultaneously optimizes the mission and the morphed design at each point. In all 4 cases, the objective function is fuel burn over the entire mission. The structural failure and thrust constraints are aggregated using a Kreisselmeier–Steinhauser (KS) function [139, 140].

In all design and morphing optimizations, the wing twist and thickness profiles are optimized using B-splines where the design variables are the B-spline control points. The aerostructural model used in OpenAeroStruct is shown in Figure 2.5. The vortex-lattice and finite-element meshes for the wing have 19 span-wise nodes and 5 span-wise control points. The mission profiles are discretized with 25 nodes in all cases. The initial twist and thickness distributions for all optimizations were determined using a 5-point multipoint optimization with 5 different cruise conditions. This ensures that the fixed-design optimization has near-optimal twist and thickness

Table 2.2: Optimization problem formulations for each of the three cases.

Category	Name	Quantity				Upper	Units
		Fixed-design	Static-design	Direct morphing	Lower		
Objective	fuel burn	1	1	1	–	–	kg
Variables	twist	0	5	125	–3	8	degrees
	spar thickness	0	5	5	0.01	0.5	m
	altitude profile	25	25	25	25	14	km
	Total	**25**	**35**	**155**			
Constraints	von Mises (KS)	0	1	1	0	$\frac{\sigma_{yield}}{2.5}$	Pa
	min. slope	25	25	25	–20	20	degrees
	max. slope	25	25	25	–20	20	degrees
	min. thrust (KS)	1	1	1	0.01		
	max. thrust (KS)	1	1	1		1.00	
	Total	**52**	**53**	**53**			

distributions as opposed to the baseline design.

We now discuss each of the three types of optimization problems in more detail.

2.3.2.1 Fixed-design optimization

The first optimization problem is a mission-only optimization that provides a baseline for evaluating the impact of static-design optimization on fuel burn. The design variables are the altitude values at the 25 mission points, as we see in Table 2.2. Minimum and maximum slope constraints are enforced at each mission point; these are linear constraints that improve robustness of the optimization by eliminating unreasonable altitude profiles. Minimum and maximum thrust constraints are also enforced since thrust profile is an output of the model given altitude profile is an input. Since these are nonlinear constraints, they are aggregated using Kreisselmeier–Steinhauser functionals [139] for more efficient derivative computation. These mission-related design variables and constraints are included in the other three optimization problems as well.

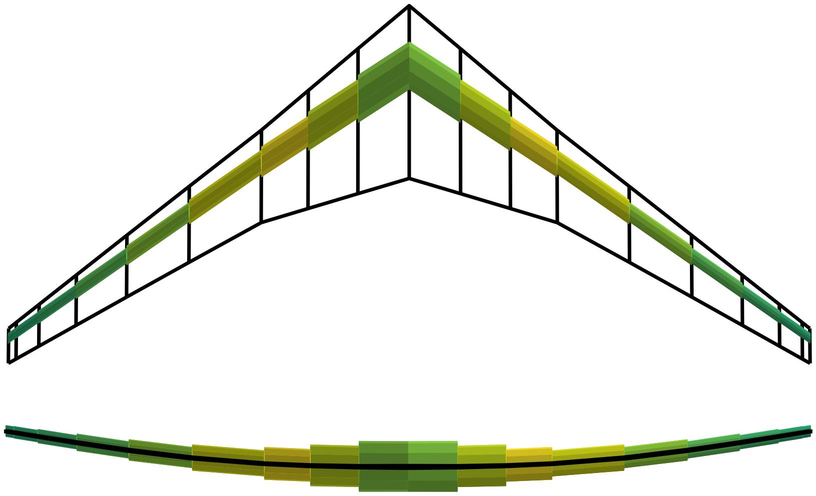


Figure 2.5: Top and front view of the multipoint optimized wing from OpenAeroStruct showing the structural spar and the lifting surface mesh.

2.3.2.2 Static-design optimization

The second optimization problem includes the mission-related design variables and constraints and adds the wing design without morphing to the problem. As we see in Table 2.2, the twist and structural thickness parameters are now design variables, but these values are static throughout the mission. Therefore, this is a simultaneous mission-design optimization that designs the optimal wing with no morphing capability. Since the structural thickness parameters are optimized, we enforce von Mises stress constraints, aggregated using Kreisselmeier–Steinhauser functionals [139] as we did with the thrust constraints.

2.3.2.3 Direct morphing optimization

The third optimization problem now adds morphing to the previous problem. The only difference from the static-design optimization is that the twist distribution is not static over the mission, but allowed to take on different values at each mission point. As we see in Table 2.2, this is the largest optimization problem because the twist is discretized both span-wise and over the mission

Table 2.3: The morphing wing always has a lower fuel burn value than either other case, as expected. The benefits of morphing technology are more pronounced in the long range mission where the fuel burn benefit from lower structural weight is larger.

	Fuel burn, kg		
	1000 nm	3000 nm	7000 nm
Fixed-design	14,697.7	41,879.9	116,774.0
Static-design	14,215.4	40,228.5	114,613.0
Direct morphing	14,181.9	40,053.3	113,848.8

profile. We call this the 'direct' morphing optimization problem because the final problem uses an indirect approach to capturing morphing that is cheaper, but introduces some error.

2.3.3 Quantifying the benefits of morphing

Here, we solve the three optimization problems described in Table 2.2 with three mission ranges—1000 nm, 3000 nm, and 7000 nm. The resulting optimal fuel burn values are shown in Table 2.3 and plotted in Fig. 2.6 as normalized fuel burn values with the fixed-design mission optimization result as the reference. For each of the three mission ranges, the morphing wing outperforms the static-design as expected. Compared to the static-design, the morphing wing sees between 0.2 to 0.7% fuel burn improvement, with the long range mission seeing the biggest improvement. The fixed-design approach has a set thickness distribution which is overly conservative for the short and medium range missions, but is closer to the optimal thickness distribution for the long range mission. This is why we see the fixed-design for the long-range mission yield a fuel burn value closer to the static-design and direct morphing cases. For different aircraft with more varied mission profiles, we would expect to see a larger increase in performance from the addition of morphing technology.

We now investigate the results from the 1000 nmi and 7000 nmi missions in more detail, in Fig. 2.7 and Fig. 2.8 respectively. Figure 2.7 shows more information from the short-range optimizations for the fixed-design, static-design, and morphing cases. Here we see that the morphing

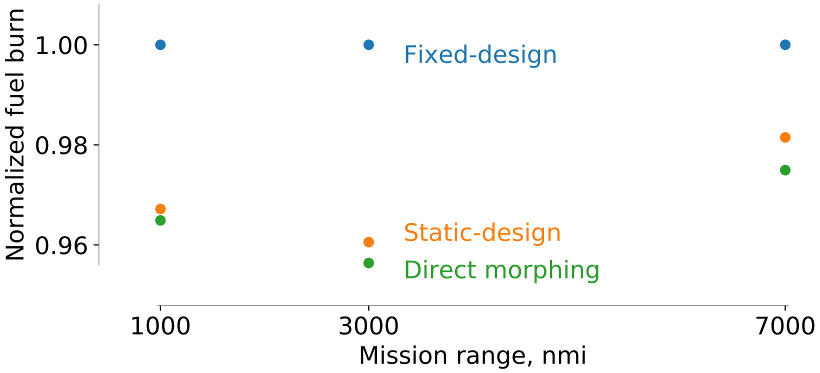


Figure 2.6: The relative decrease in fuel burn obtained with morphing technology varies based on the mission range. The long range mission using direct morphing sees the biggest improvement in fuel burn compared to the static-design.

wing uses slightly less fuel for the entire mission, but all of the resulting curves are very close to the results from the static-design. The altitude profile is limited by the upper bound of 14 km during the cruise segment. For this short mission, the fuel weight and thus the aircraft weight are relatively low, which means that the optimal flight altitude is higher than that for a heavier aircraft. This altitude limit forces the aircraft to fly at a suboptimal L/D ratio, which we see during the cruise segment of the mission. We would expect to see a greater benefit from morphing technology if we included operational constraints, such as constant flight level cruise segments. This would force the aircraft to fly at more distinct flight conditions, which would cause the static-design to be less optimal over the entire mission.

Each optimization method yields roughly the same altitude, path angle, and velocity profile, but the largest differences are seen in the C_L and L/D profiles. The fixed-design case is forced to optimize the mission with a fixed structural weight for a conservative structure, which forces it to fly at a slightly higher C_L. For this optimization case and others, we see that the L/D ratio is higher than expected. We use C_L and C_D offsets to account for the lift at zero angle of attack and the drag components from the rest of the aircraft; however, we suspect that the drag is underpredicted, which is the cause of the relatively high L/D values we see.

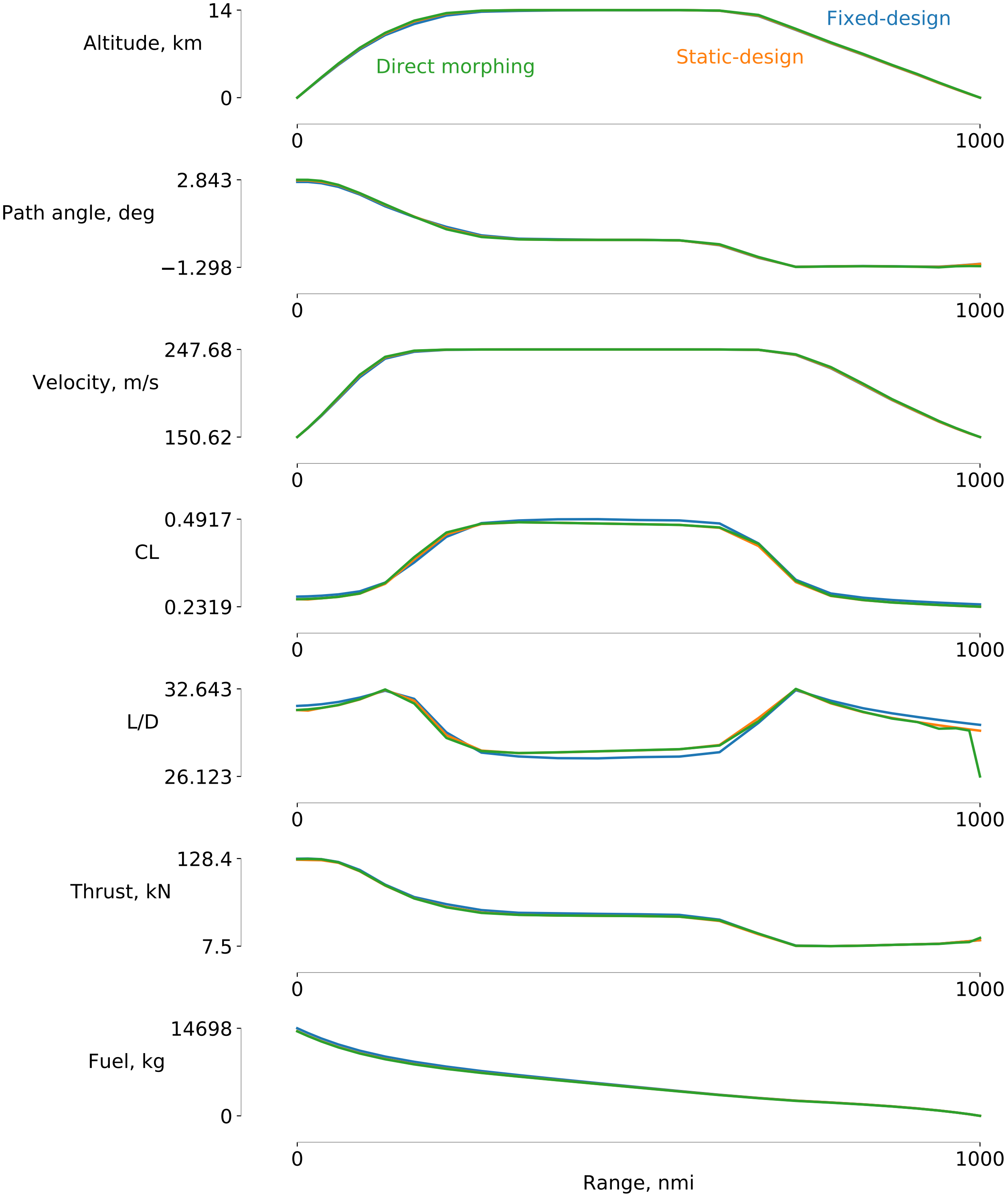


Figure 2.7: Each of the optimization methods yields aerodynamic and mission profiles that are similar. The main differences come from the higher C_L seen for the fixed-design case, where the relatively inefficient design necessitates a larger amount of lift.

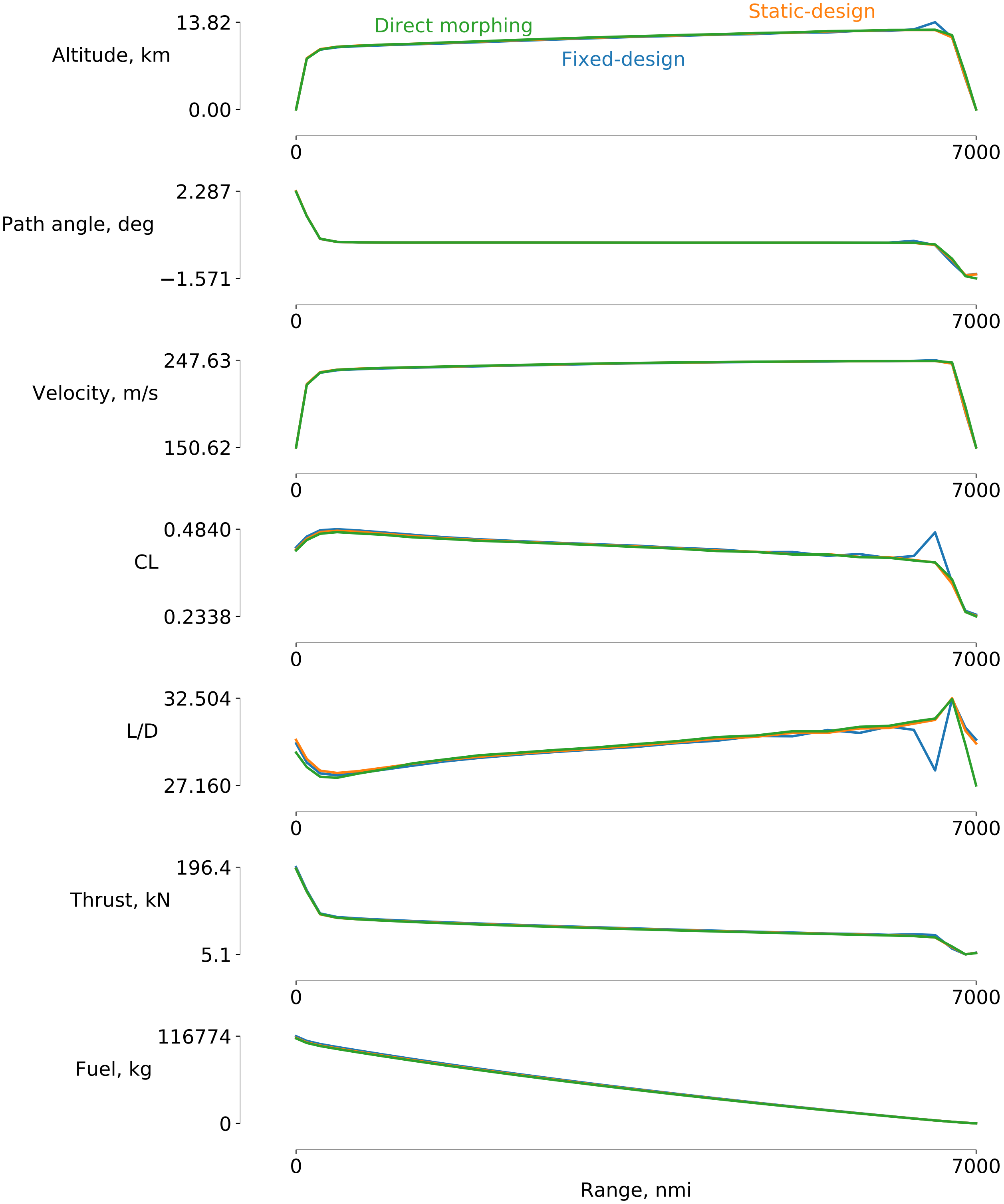


Figure 2.8: Most of the gains of morphing technology come from tailoring the structural properties of the wing. The lighter weight aircraft can fly at a slightly lower C_L which produces a lower fuel burn. Near the end of some mission profiles, we see sharp changes in the C_L and L/D, which are due to the problem discretization.

Figure 2.8 shows results from the long range mission using the same three optimization formulations. As expected, the altitude increases and the angle of attack decreases as fuel is burned, since the aircraft can sustain level flight with less total lift. Thrust slightly decreases throughout the cruise segment, and L/D improves throughout the mission as the aircraft can fly at a more optimal flight condition. Additionally, the fixed-design optimization requires a higher C_L at the beginning of the mission due to its aerodynamic inefficiency. This also leads to the increased thrust for the fixed-design optimization. Near the end of the mission profiles, we see jagged jumps in the C_L and L/D profiles, especially for the fixed-design case. We believe this result is due to numerical effects, such as the relatively coarse mission discretization or noise in the function evaluation, and they merit further investigation.

2.3.4 Summary of results

We used a direct morphing formulation to compare optimization results for a morphing wing to a static-design wing for a commercial airliner. The direct morphing result achieved a 0.2 to 0.7% fuel burn reduction compared to the static-design optimized result, where the reduction is larger for longer missions. As expected, the performance gains from adding morphing technology to a commercial airliner wing are very small. Using this direct morphing approach to evaluate the gains from morphing technology for other aircraft with more complex missions or more articulate morphing mechanisms would result in a larger increase in performance.

2.4 Mission optimization of a supersonic aircraft

The previous section examined mission optimization of a subsonic aircraft, though as Sec. 1.2.1 discussed, optimal trajectories become much more important for supersonic aircraft. In this section, we introduce a model for the ESAV aircraft and perform trajectory optimization for a few representative missions.

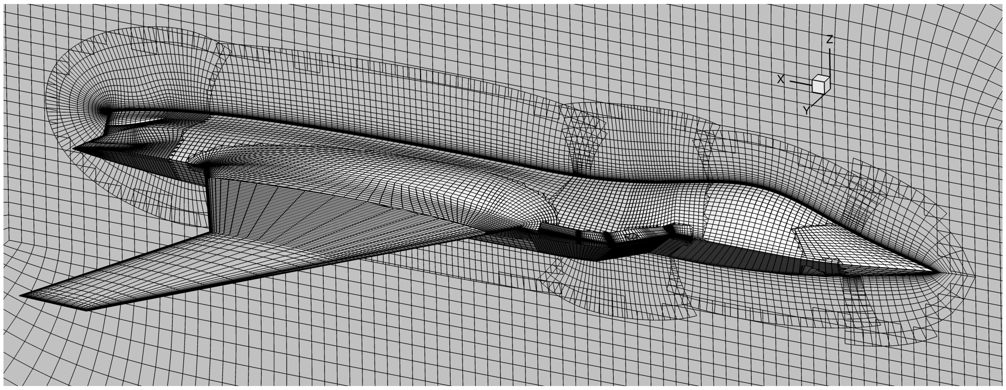


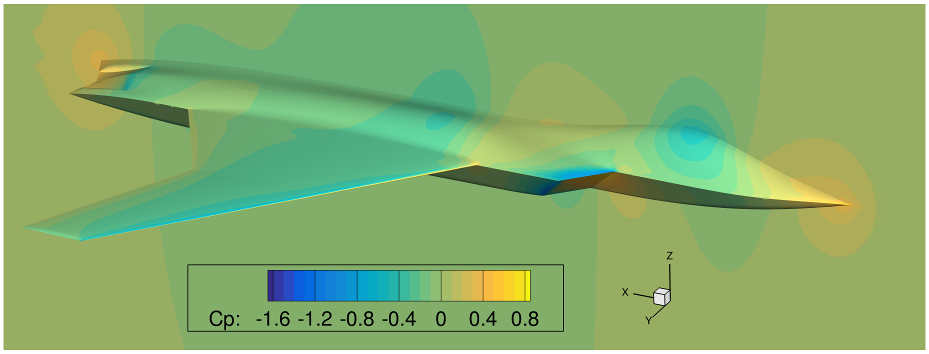
Figure 2.9: View of the overset mesh used to represent the ESAV geometry. Nine surface patches were created and extruded hyperbolically to create the set of volume meshes.

2.4.1 Supersonic aircraft model setup

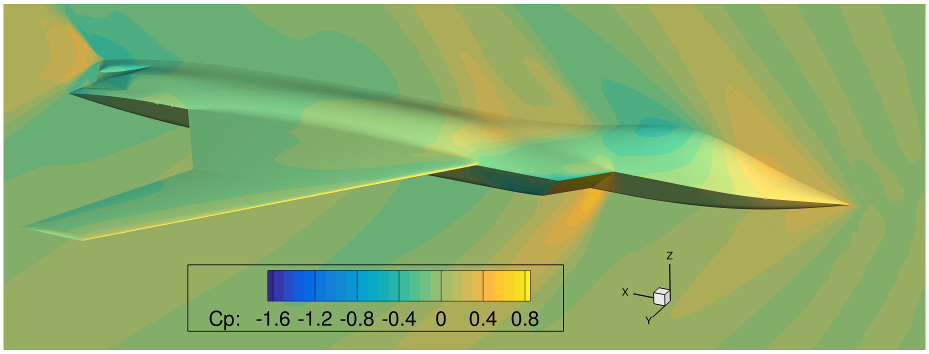
The RANS solver used in this work is ADflow [141, 142], a structured finite-volume CFD code with overset capabilities using an implicit hole cutting scheme [143]. The geometry was meshed using on overset mesh made up of 10 blocks and a total of 2,185,600 cells. This surface mesh and symmetry plane are shown in Fig. 2.9.

We converge the flow solution using the approximate Newton–Krylov (ANK) solution algorithm. We use the Spalart–Allmaras turbulence model and the fluxes are discretized with central-differencing. Two nominal flow solutions for the aircraft, one at Mach=0.8 and the other at Mach=1.5, are shown in Fig. 2.10.

Although we could use this CFD model directly in the optimization problem and perform an analysis at each point needed in the mission simulation, we do not need to do that. Instead, we construct a surrogate model based on many training points using data from the CFD analyses. We uniformly query the three-dimensional Mach-α-altitude space to populate the surrogate training points, and obtain C_L and C_D as outputs from the CFD analysis. We use 13 points in the Mach direction, from 0.2 to 1.8, clustered around the transonic region, 7 points from -8 to 8 degrees α, and 3 points in the altitude space between 0 and 20,000 meters for a total of 273 CFD evaluations.



(a) Flow solution at $\alpha = 2$ and Mach=0.8.



(b) Flow solution at $\alpha = 2$ and Mach=1.5.

Figure 2.10: Two flow solutions of the ESAV geometry at nominal flight conditions.

Table 2.4: Design parameters for the supersonic engine model

Parameter	Value	Units
Design altitude	35,000	ft
Design Mach number	0.8	
Design T_4	3200	° R
Design thrust	15,000	lbf
Extraction ratio	1.05	
Fan pressure ratio	3.3	
HPC pressure ratio	9.7	

We use this data to construct a surrogate model using SMT [95], which provides a variety of methods to represent data sets. We chose to use the RMTS method [94] because this method has inexpensive training and evaluation costs and also provides gradients of the outputs with respect to the inputs inexpensively. A contour plot of the trained surrogate with the training points overlaid is shown in Fig. 2.11. Figure 2.12 shows an angle of attack and Mach sweep of the aerodynamic model at an elevation of 30,000 feet.

We use a surrogate propulsion model based on the GE F110, an afterburning turbofan jet engine. The F110 is a two-spool design with a maximum dry thrust of 16,610 lbf and is used in aircraft such as the F-14 and F-15E [144]. We selected this engine because it is used in aircraft sized similarly to the ESAV aircraft.

To obtain the performance characteristics of this engine, we used a previously developed NPSS model as a basis for our engine model built in pyCycle. The engine design parameters are shown in Table 2.4. We ran this pyCycle model through a corridor of points in the Mach-altitude space with a set of combustor exit temperatures. The outputs we used from this model were thrust and thrust-specific fuel consumption (TSFC). We then used this data to construct a surrogate model, again using the RMTS method from SMT. The sampled points and surrogate model contours for full throttle cases are shown in Fig. 2.13.

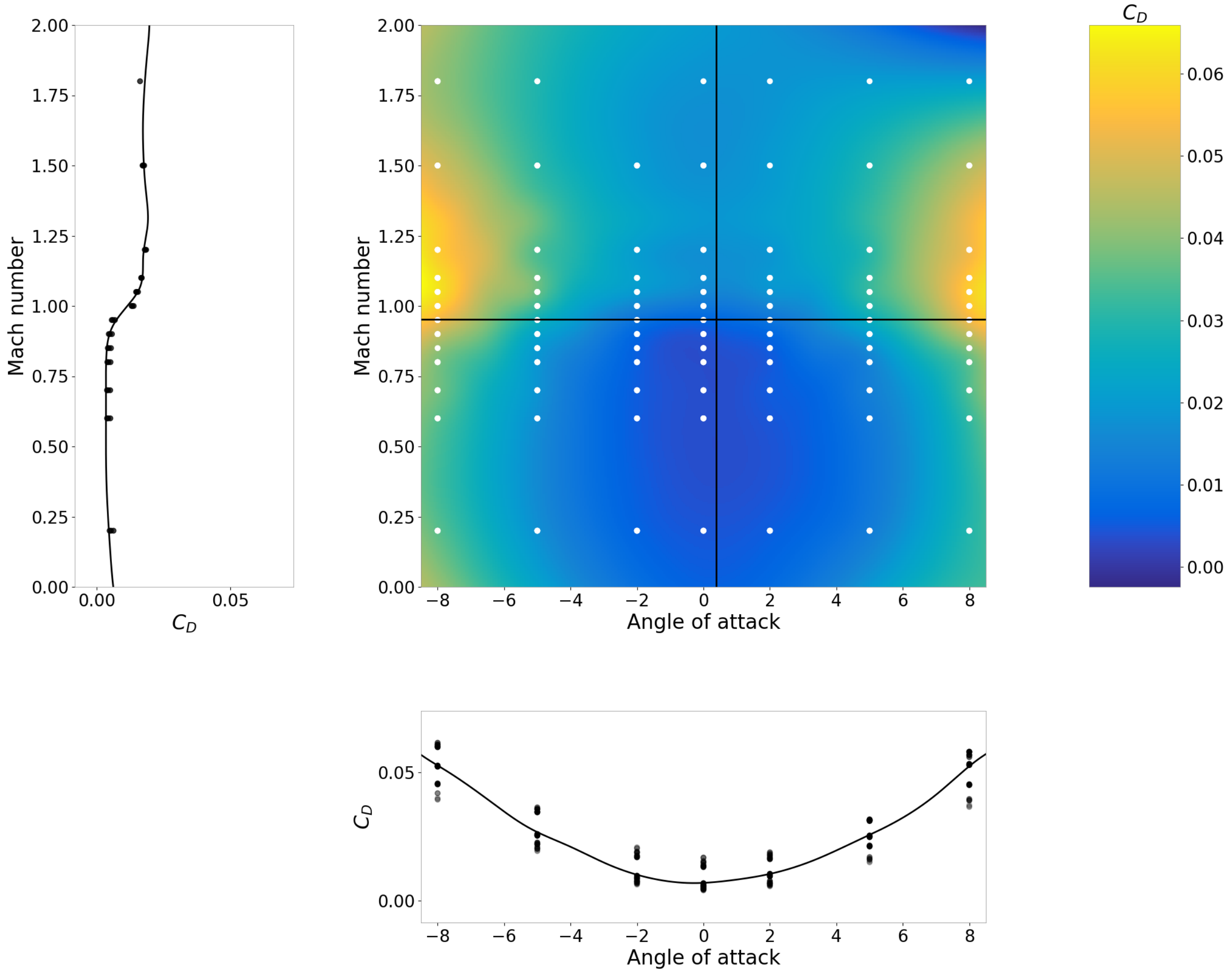


Figure 2.11: The aerodynamics surrogate model represents the training data well in a smooth and differentiable manner. This plot shows a 2D slice of the 3D input space in the contour and lines, while the points are the training data.

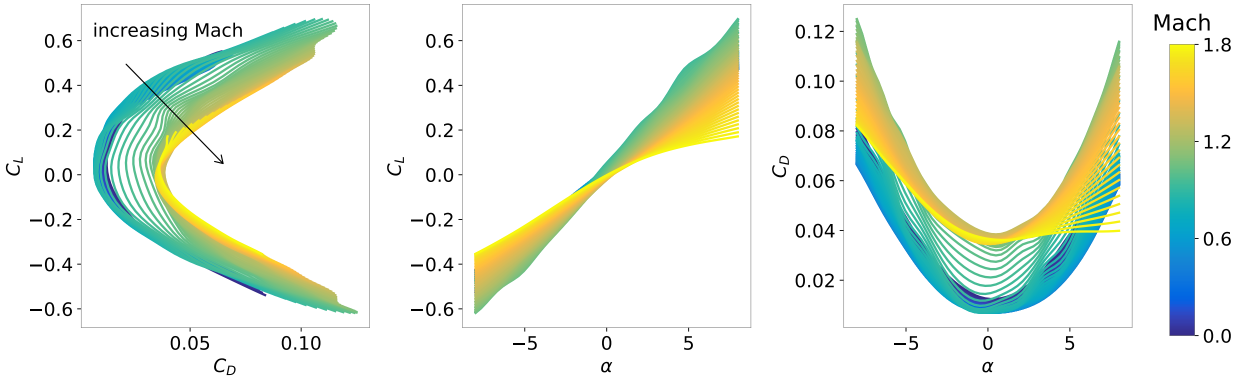


Figure 2.12: Drag polar and aerodynamic properties for the ESAV aircraft obtained from the CFD-trained surrogate model at an elevation of 30,000 feet with a Mach sweep from 0 to 1.8, and an α sweep from -8 to 8.

2.4.2 Minimum time-to-climb optimization

The minimum time-to-climb problem for a supersonic aircraft is well-studied and has become a canonical problem in trajectory optimization [17, 44, 46]. Because of this, we study a similar problem using the supersonic aircraft model we constructed. Although the canonical problem is for an F-4 with specific propulsion and aerodynamic models, we use the physics described in the previous subsection and qualitatively compare the optimal flight profiles for the ESAV aircraft.

We solve the unsteady aircraft equations of motion detailed in Sec. 2.2.1, specifically Equations 2.1, 2.2, 2.3, and 2.4. Based on the required thrust from solving those equations, T_4 is solved for using the differential inclusion approach. T_4 is the temperature of the engine flow leaving the burner and is an approximation of the throttle setting for the engine.

Table 2.5 shows our optimization problem which we will use for the following mission optimization studies. The size of the state and control arrays varies based on the order of the segments used in the collocation scheme.

For our first case, we find the minimum time for the aircraft to climb to 20,000 meters, reach Mach 1, and level out to a horizontal flight path angle, akin to a rendezvous mission. The controls

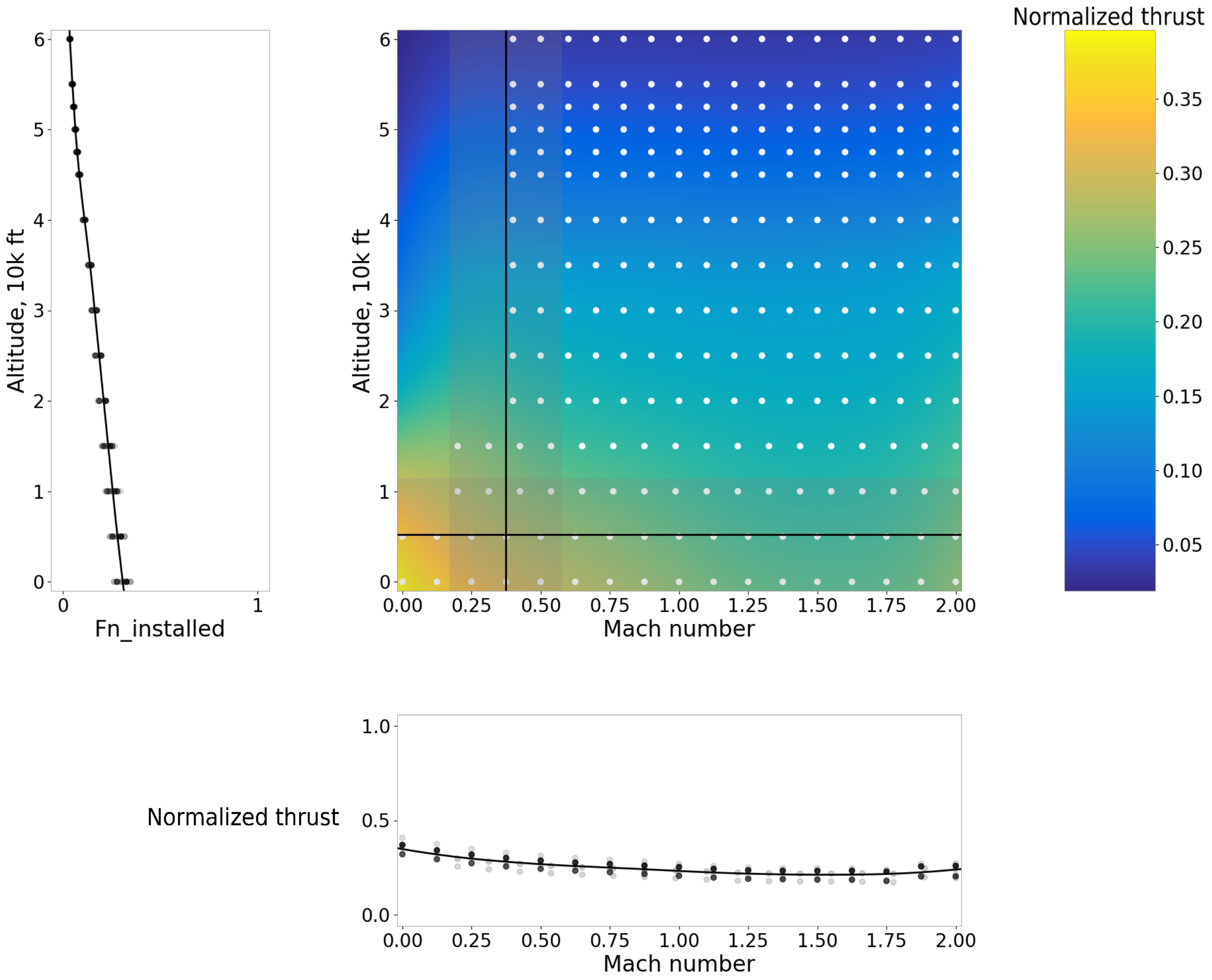


Figure 2.13: A slice of the propulsion surrogate showing normalized thrust for a combustor exit temperature of 3200 $^{\circ}R$.

Table 2.5: Nominal optimization formulation for the supersonic minimum time-to-climb problem

Category	Name	Lower	Upper	Units
Objective	fuel burn	–	–	kg
Variables	$\dot{\gamma}$	-0.1	0.1	rad/s
	$\dot{v}$	-5	5	m/s
	range	0	–	km
	altitude	0	20	km
	velocity	0	900	m/s
	γ	-0.5	0.5	rad
	mass	20,000	30,000	kg
Constraints	α	-15	15	degrees
	T_4	2000	3200	degrees R
	Mach	0.5	1.8	–
	range defects	0	0	km
	altitude defects	0	0	km
	velocity defects	0	0	m/s
	γ defects	0	0	degrees
	mass defects	0	0	kg

across the phase are parameterized using 5th order polynomials to produce smooth control profiles and increase optimization robustness. Similar polynomial controls are used throughout this dissertation.

Figure 2.14 shows the optimal flight profile and accompanying data for this problem. We see similar trends compared to the canonical problem, especially through the trade of altitude and speed increase. The aircraft initially stays at a low altitude as it accelerates, then accelerates at a slower rate as it gains altitude. Once at an altitude of around 12 km, the aircraft accelerates before zooming up to the desired 20 km altitude at a speed of Mach 1.0. The angle of attack starts high to produce enough lift for the aircraft at its initial low speed, then decreases before slowly increasing through the flight. Near the rendezvous point, the angle of attack sharply decreases to get to level flight heading. We see that T_4, effectively a measure of throttle, is near its upper limit throughout the flight profile. Right before the end of the mission, the engine is throttled back to reduce thrust so that the aircraft can deaccelerate to Mach 1.0.

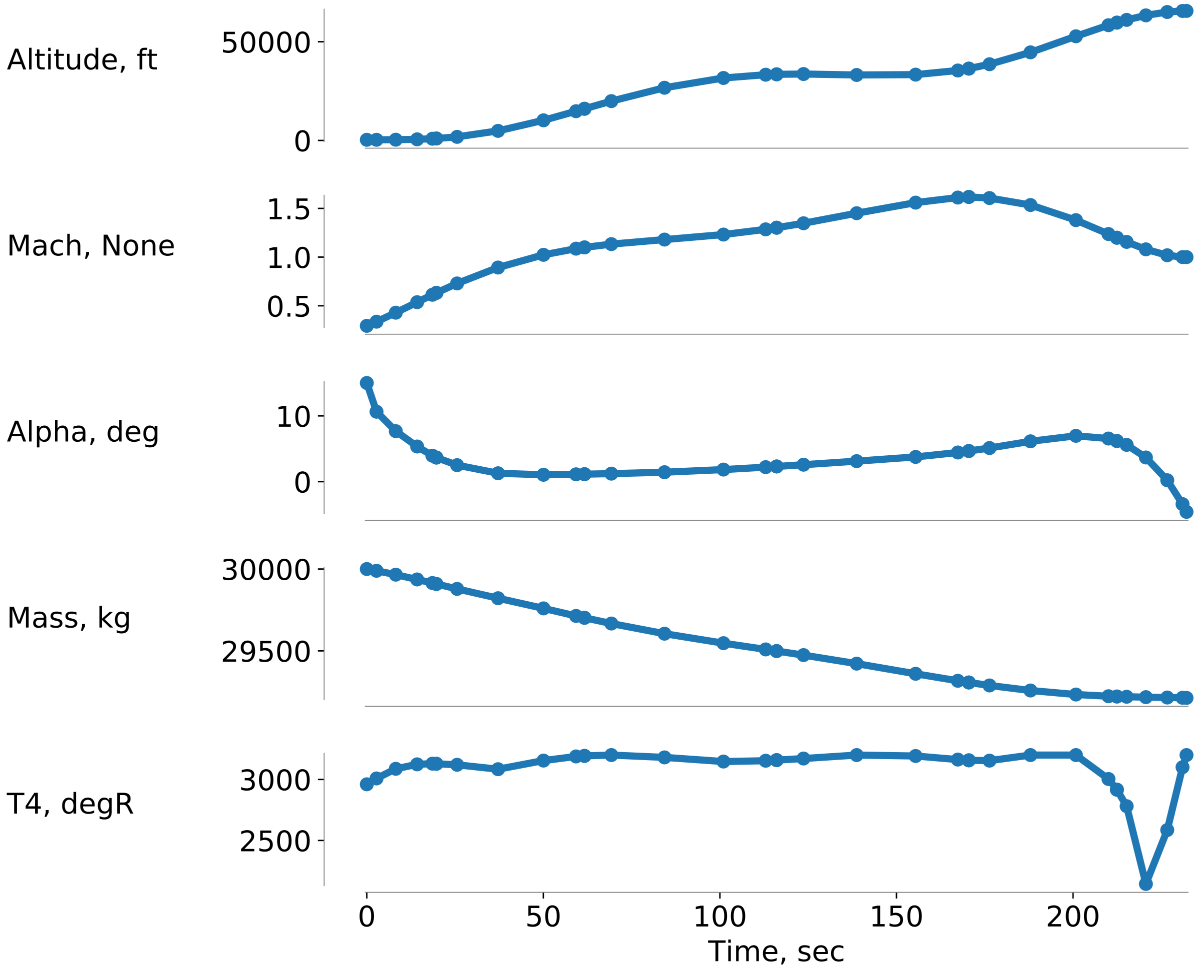


Figure 2.14: The optimal flight profile for the minimum time-to-climb mission

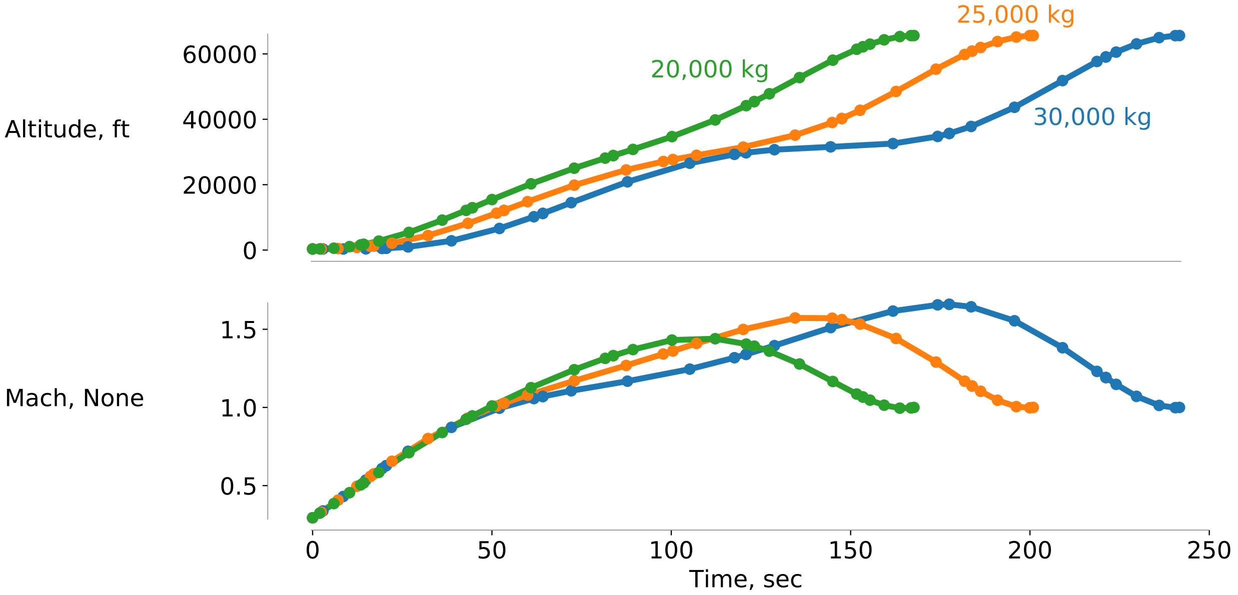


Figure 2.15: Optimal flight profiles for different initial masses

There are many studies we could do using the minimum time-to-climb problem, including examining physical effects, such as aircraft mass, thrust capabilities, or target altitude and speed, and integration effects, such as discretization, control scheme, or problem scaling. We briefly examine variants of this problem before moving on to more complex missions.

Figure 2.15 shows the optimal altitude and Mach profiles for three different initial aircraft masses. The 30,000 kg case is the same result as shown in Fig. 2.14. As the initial aircraft mass is lowered, the optimal flight profile becomes more aggressive and takes less time to reach the desired altitude and speed. We see that the 20,000 kg case does not need to accelerate then climb like the 30,000 kg case. Instead, the lighter aircraft can gain both speed and altitude due to the higher thrust-to-weight ratio. This shows that both the optimal answer *and* the characteristics of the optimal mission can vary based on physical parameters, making it challenging to generalize results for a specific aircraft or model.

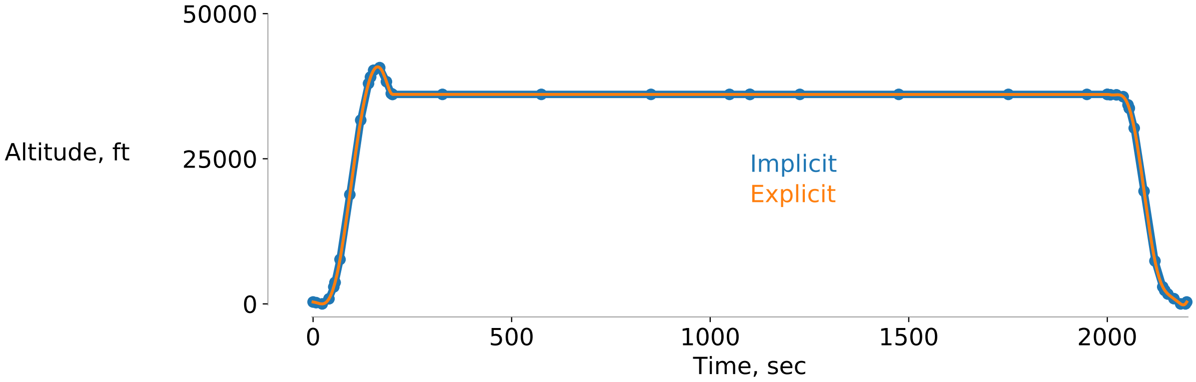


Figure 2.16: The implicit collocation scheme results are verified via explicit integration.

2.4.3 Full mission optimization of a supersonic aircraft

So far, we have only presented simplistic mission results for the supersonic aircraft. In this subsection, we extend this modeling to full missions, including multiple cruise, ascent, and descent phases.

When using implicit pseudospectral collocation methods, it is valuable to verify the resulting time-histories against explicit integration schemes. This ensures that the model physics are being adequately resolved via the collocation method. Based on the stiffness of the ODEs, as well as the number of discretization points used for the integration, it is possible for a fully converged optimization to not accurately capture the model physics. This is because stiff ODEs might have peaks or valleys that are between collocation nodes, and therefore not known to the method, leading to integration inaccuracies. Figure 2.16 shows that for a given supersonic cruise mission, the results from the implicit and explicit time integrations match well. This further reinforces our use of implicit methods to reduce optimization cost as we obtain the same results as more expensive explicit methods. This nominal mission optimization minimized fuel burn for a prescribed cruise duration and speed. The optimizer could control the ascent and descent profile, though the cruise phase was fixed.

It is important to consider the number of discretization points used to break up the mission. A small number of points will be computationally inexpensive, but may not capture the model physics well enough to reach meaningful conclusions. At the same time, a large number of points will be much more computationally expensive, but will resolve the physics much better. Additionally, a large number of points will increase the size of the optimization problem, both in terms of the costs associated with computing the total derivatives and probably the number of optimization iterations required for convergence. Using polynomial controls will somewhat reduce the number of design variables exposed to the optimizer, but all of the state values will still be considered as design variables.

Figure 2.17 shows the fuel burn-minimizing altitude profiles for the same supersonic cruise mission using different number of discretization points per phase. Each segment for all cases uses the 5th order LGR collocation method and there are three phases: ascent, cruise, and descent. From top to bottom, there are 1, 2, 3, 6, and 10 segments per phase, which translates to 16, 34, 52, 106, and 178 discretization nodes. Each phase uses 5th order polynomial controls for $\dot{v}$ and $\dot{\gamma}$ so that the optimization design freedom is similar between each case.

There are constraints on cruise Mach = 1.4 and a final point Mach = 0.5, which causes the increase in altitude before descending noticeable in the bottom few results. The aircraft has excess speed at the end of cruise, increases altitude to lose some of that speed, and coasts down from there slightly above engine idle. This shows how much of an effect the optimization problem formulation and constraints have on optimal results.

Table 2.6 compares the resulting aircraft mass and optimization cost for a subset of these cases. We see an increase in fuel burn as we use a finer discretization, suggesting the coarser discretizations are not modeling the actual physics as well. The coarsest discretization takes 23.2 seconds to optimize whereas the finest discretization takes 605 seconds. Depending on the cost evaluating the ODE and the desired accuracy, any one of these discretizations may be the best choice for a given application and study.

No matter the discretization or control scheme used, optimal results must be examined in de-

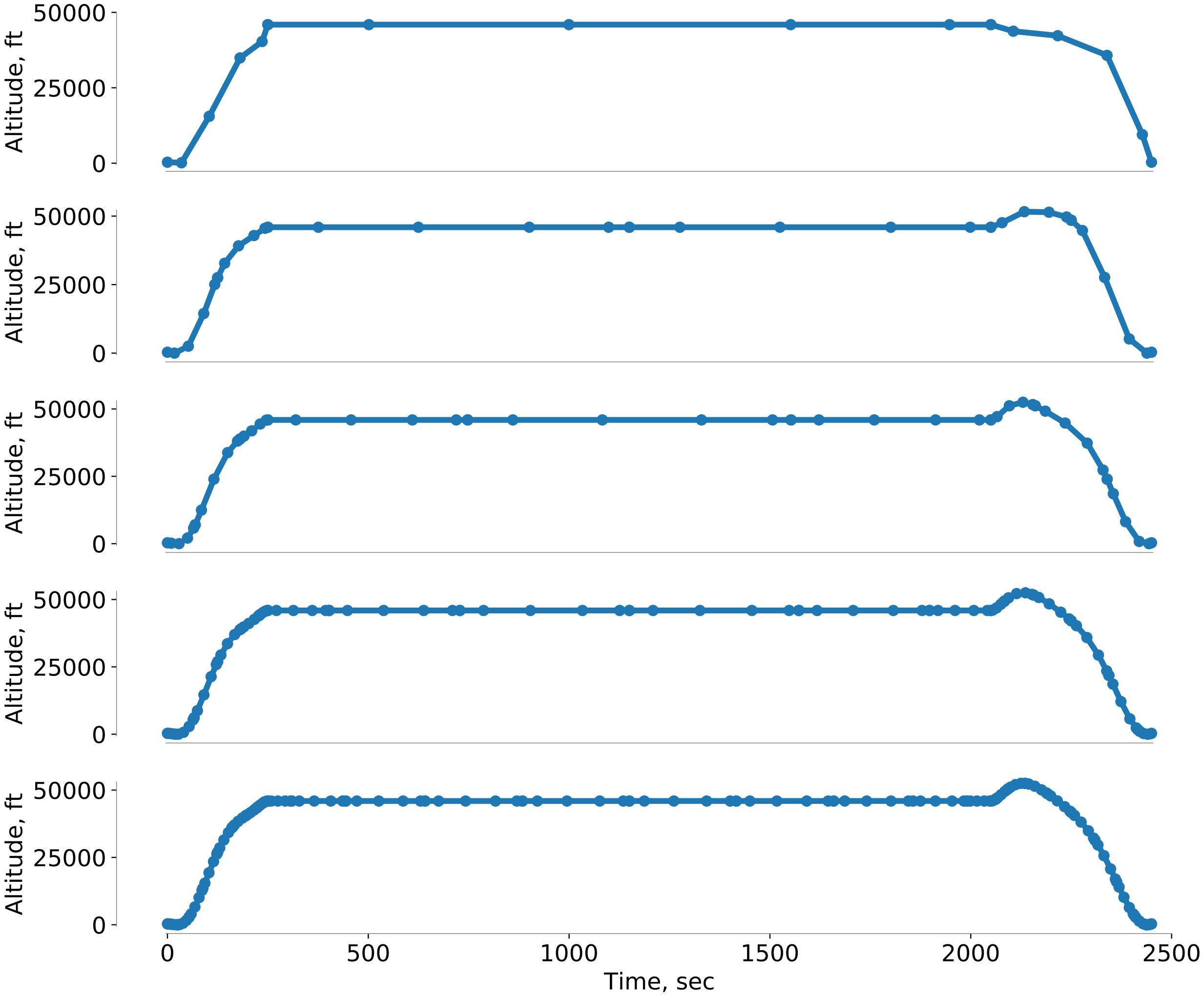


Figure 2.17: There is a balance between optimization cost and time integration accuracy based on the number of nodes used to discretize the mission problem.

Table 2.6: Optimal results and timings from the discretization study

Figure placement	Number of nodes	Final aircraft mass, kg	Optimization time, secs
Top	16	26,652	23.2
Middle	52	26,538	356
Bottom	178	26,529	605

tail with disciplinary expertise to ensure they make physical sense. The optimizer could exploit numerics for an edge case or a poorly formulated problem and produce a non-meaningful result. Not interpreting optimization results adequately is a common pitfall for many researchers. With that being said, this dissertation focuses on both presenting and interpreting the optimal results of complex multidisciplinary physical systems.

We now move to a more complete mission profile based on a supersonic strike mission, partially adapted from Simmons [145] and Clark et al. [37]. Simply, the aircraft must complete three separate cruise segments: a supersonic cruise to a target area, then a low-altitude strike cruise, then a supersonic cruise back to its takeoff location. In between the prescribed cruise segments, there are ascent and descent phases. For this setup, the time to complete each phase is fixed, so the optimizer cannot vary the phase length. The optimizer is minimizing fuel use for this mission, which could then allow more armaments on board.

There are many formulations we could use, such as considering effectiveness-based design [37] for an objective function, but we minimize fuel burn here since we are focusing on the optimal mission and control profiles. Table 2.7 shows the full optimization problem for this three-cruise mission and table 2.8 shows the mission requirements and phase constraints. Although this optimization problem is roughly square and thus neither the direct or adjoint method is markedly more efficient than the other, we use OpenMDAO's graph coloring capabilities [146] to decrease the effective Jacobian size. This problem introduces the use of an air brake, which allows the aircraft to slow down quickly in air by adding a large amount of drag. In this implementation, the optimizer can choose to increase the C_D of the aircraft for the descent phases to quickly lose energy. The number of design variables and constraints is determined by the order of the segments and the number of segments used in the collocation scheme. The control profiles are again determined via 5th order polynomials.

Figure 2.18 shows the optimal altitude, Mach, mass, and T_4 profiles for this complex mission. Because we prescribe the Mach number in the cruise segments, the aircraft has to dissipate energy

Table 2.7: Nominal formulation for the supersonic strike mission problem

Category	Name	Quantity	Lower	Upper	Units
Objective	fuel burn	1	–	–	kg
Variables	$\dot{\gamma}$	15	-0.1	0.1	rad/s
	$\dot{v}$	15	-5.	5.	m/s
	CD_{brake}	10	0	0.05	–
	range	55	0	–	km
	altitude	55	0	16	km
	velocity	55	10	900	m/s
	γ	55	-0.5	0.5	rad
	mass	55	20,000	30,000	kg
	Total	**315**			
Constraints	α	55	-15.	15.	degrees
	T_4	55	2000	3200	degrees R
	altitude	15	1	14	km
	Mach	17	0.5	1.8	–
	range defects	55	0	0	km
	altitude defects	55	0	0	km
	velocity defects	55	0	0	m/s
	γ defects	55	0	0	degrees
	mass defects	55	0	0	kg
	Total	**417**			

Table 2.8: Additional mission details and constraints for the supersonic strike mission. These values are fixed and do not change during the optimization.

Phase parameter name	value	units
1st ascent starting altitude		km
1st ascent starting speed		km
1st ascent duration	200	seconds
1st cruise altitude	11	km
1st cruise Mach number	1.4	
1st cruise duration	1700	seconds
1st descent duration	200	seconds
2nd cruise altitude	1	km
2nd cruise Mach number	1.2	
2nd cruise duration	500	seconds
2nd ascent duration	200	seconds
3rd cruise altitude	14	km
3rd cruise Mach number	1.8	
3rd cruise duration	1700	seconds
2nd descent final altitude	100	m
2nd descent final Mach number	0.5	
2nd descent duration	500	seconds

by trading between elevation and speed, which causes the bumps in altitude between cruise segments. The objective was to minimize fuel burn within a fixed time, which causes the optimizer to choose conservative ascent and descent flight profiles. T_4 appears relatively discontinuous because it is being solved for using the differential inclusion method. It could have been posed as a control, but because it was being implicitly solved for from a set of smooth controls, the resulting T_4 solution can be discontinuous. In reality, there would be some hysteresis in the T_4 value because it cannot change instantaneously, but in this work we are considering each node to be a quasi-steady point and do not account for transients on that time scale.

If we used a single mission phase instead of seven, we would not be able to easily constrain the speed and altitude across different flight segments. Using three constrained cruise phases reduces the freedom that the optimizer has to tailor performance, but results in a more stable optimization problem. Further studies into the effects of cruise constraints and optimal performance without

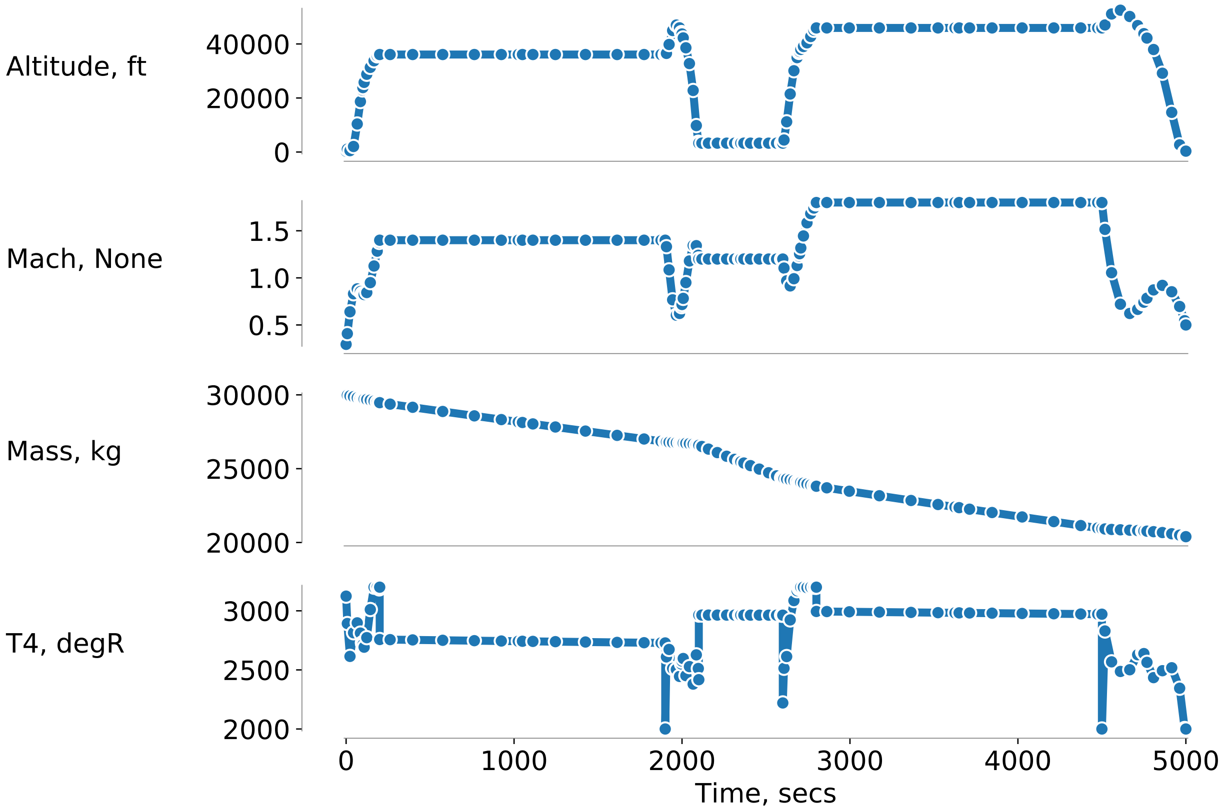


Figure 2.18: Fuel burn-minimized flight profiles for the supersonic aircraft with three cruise segments

those constraints would help show how operational mission constraints affect aircraft performance.

This last optimization problem closes out this chapter on mission optimization. With the capabilities detailed in this chapter, we have the necessary foundation to add other disciplines and models to the mission optimization problem. Although we could delve into much more detail concerning only mission optimization, we aim to optimize large-scale multidisciplinary problems and thus focus on other disciplines and the multidisciplinary integration in the following chapters.

CHAPTER 3

Thermal Systems Optimization

3.1 Review of thermal systems in path-dependent problems

Already an important multidisciplinary consideration for current aircraft, future survivable military aircraft will have heat loads three to five times that of legacy aircraft [43]. Additionally, future aircraft will have subsystems designed to minimize mass and volume, which decreases potential avenues for heat sinks and dissipation. To correctly analyze the performance of future military aircraft, designers must consider the energy of the entire system throughout the entire flight envelope, as emphasized by AFRL and Lockheed Martin [43, 147].

However, to correctly determine the thermal performance of an aircraft, many of the subsystems must be designed so that their mass, thermal loads, and potential dissipation capabilities can be considered. This leads to an iteratively coupled design loop that can be solved using MDO or humans-in-the-loop. As aircraft systems become even more complex and we introduce new physical subsystems, the use of automated MDO to analyze the performance becomes paramount. The impact of thermal requirements is apparent in the current generation of fighter aircraft. For example, the F-35 joint strike fighter suffers from thermal management challenges that impact its performance and can require changes to mission profiles to keep temperatures within acceptable ranges [148].

Future military aircraft may need to dissipate heat from extreme sources, such as directed energy weapons (DEW). Figure 3.1 shows the trends of heat and power dissipation needed for current and future aircraft. Thermal system management for aircraft with DEW systems has been publicly been a research interest since 1983, when President Ronald Reagan proposed the Strategic Defense Initiative. As laser technology matured, the US Air Force developed a 1MW testbed using a modified Boeing 747 [149]. Laser weapons continue to get more efficient and portable, leading to further research on their use in smaller fighter-scale aircraft [42, 149, 150]. The thermal loads from these DEW systems are performance critical, thus the thermal systems and architectures needed to handle the extreme conditions have been studied extensively, notably by Gvozdich et al. [42].

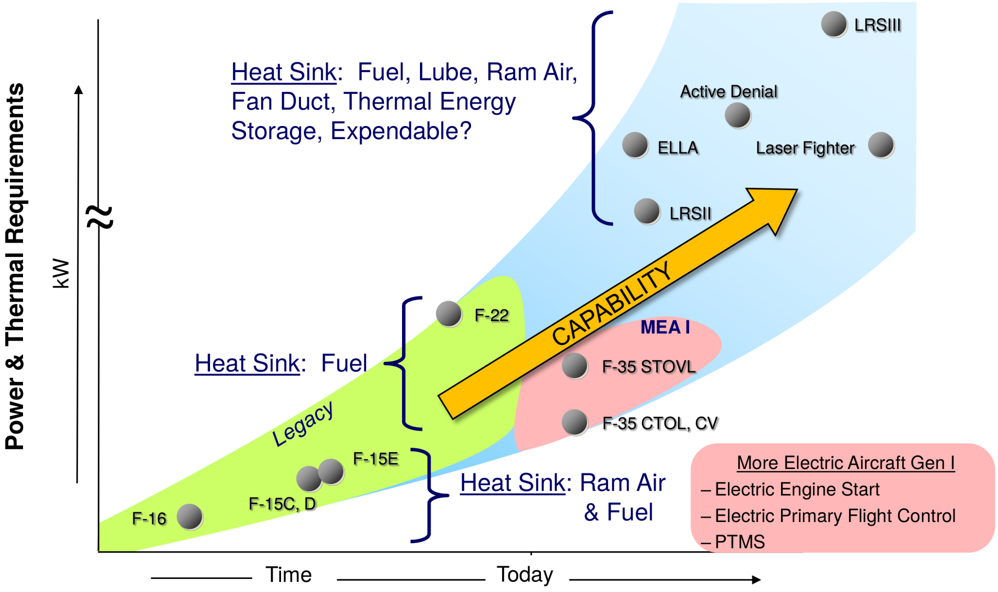


Figure 3.1: Future aircraft will have much larger thermal loads to manage. Adapted from Wolff [43].

Although DEW systems are of interest to military aircraft designers, we will not study them in this dissertation due to their complexity and oft-classified properties. Additionally, the short-scale transient effects of cycling the laser are on the order of seconds, whereas the long-scale mission transients are on the order of minutes and hours. The disparate appropriate time scales make coupled analysis and optimization intractably expensive. Instead, we focus on the thermal system with a nominal cooling setup and loads from other heat sources, such as avionics, actuators, and cockpit air conditioning.

There are a variety of methods to dissipate heat on an aircraft, including FTMS, ram air coolers, bypass heat exchangers, and ablative materials [43]. Many operational constraints, such as radar observability and weight limits, reduce the available ways to dissipate heat. With this in mind, we detail some of the different cooling methods available to supersonic military aircraft.

One option for the cooling system is to use fuel-to-air heat exchangers to dissipate heat [151]. This solution tends to interfere with the stealth properties of the airframe and thus may not be a feasible option in modern fighter design. Other solutions have been proposed using FTMS to reject waste heat from the aircraft while still maintaining a lower infrared signature [152]. A simple but low efficiency FTMS concept would be to simply dump fuel overboard after it has been used to pick up waste heat, but Alyanak and Allison [39] showed that this would result in a massive increase in take-off gross weight. More refined concepts include using feed tanks and valves to control the fuel cooling loops and then recirculate the hot fuel back into the fuel tank [153]. This approach works as long as the fuel temperature in the tank does not exceed a certain threshold, but for longer missions the temperature of the fuel in the tank can become a limiting factor.

Heat can be rejected via airstreams within the engine without drastically increasing radar signature. Because of this, interactions between the propulsion and thermal systems have been studied extensively for military aircraft [38, 39, 75, 89, 154]. The effect of heat exchangers within engine bypass ducts is a specific area of interest. Bypass heat exchangers are useful in both transonic and supersonic aircraft. Prior work has studied bypass heat exchangers using wind-tunnel experiments [155, 156] and computational methods [75, 154–156]. Bypass airflow heat exchangers are also useful in transonic commercial airlines, including the Boeing 787, which has a surface cooler to reject engine oil heat and generator heat [157], showing the wide applicability of this research.

Many thermal sources produce low quality heat or need to be kept below a certain temperature.

The ability for the heat exchanger to transfer thermal energy into the bypass stream is dependent on its surface area, the fluid properties at both the hot and cold sides of the heat exchanger, as well as the temperature difference between the two fluids. One way to increase the temperature difference is to use an air-cycle machine (ACM) or thermal lifting system, a type of heat pump which uses mechanical work to transfer heat from a cooler fluid to a warmer fluid [75]. Although more total heat is produced due to thermodynamic cycle inefficiency, the heat exchanger transfers heat faster due to the larger temperature differential. In the transient case, this can be beneficial to reject a large amount of heat quickly when conditions allow, such as when the external air is cooler. To power this thermal lifting system, energy can be used from batteries, bleed air, or shaft power offtakes from the engine.

The benefit from using a thermal lifting system may not outweigh the associated weight and complexity penalty. Additionally, the energy used to lift the thermal load will impact other subsystems within the aircraft. Recent studies have found that shaft power offtake from the engine is more efficient than using bleed air [158], as shown in Fig. 3.2.

Figure 3.2: Normalized fuel burn cost to provide 30 kW of forced air cooling based upon different energy sources. Data adapted from Williams [158].

Another way to dissipate thermal energy is to modify the aircraft's trajectory to reach cooler air at higher altitudes where more effective heat rejection is possible through the larger difference between ambient and aircraft temperatures. This has the benefit of improving the effectiveness of the FTMS, but it could also mean that the aircraft is forced to climb faster than it would otherwise have, and hence use more fuel to fly its mission. Additionally, the decrease in air density at higher altitudes will affect the performance of a ducted heat exchanger.

Regardless of what heat dissipation system is chosen, the performance of the aircraft becomes a fundamentally path-dependent problem when aircraft designers are unable to add sufficient heat rejection capabilities into the airframe. This is because the aircraft's full flight path history impacts the thermal performance at any given time. Although path-dependent performance is not typically analyzed as part of early aircraft design, some have considered thermal effects of engines across a flight path, FTMS of aircraft, or thermal history of high-altitude balloons [38, 159].

This dissertation aims to push the usage of path-dependent optimization earlier in the thermal system design process so that we can accurately capture the coupled trade-offs of a new aircraft. This chapter focuses on the construction and analysis of a representative thermal system for use in gradient-based optimization. Evaluating and optimizing FTMS performance using gradient-based methods has not been done before.

3.2 Fuel thermal management systems

3.2.1 Simplified fuel thermal management system

For our first implementation of a thermal system model, we solve for a series of equations representing the fuel recirculation FTMS as initially discussed by Alyanak and Allison [39]. This model consists of a single fuel tank, two heat exchangers, and an engine feed line as shown in Figure 3.3. The aircraft can dissipate heat generated by on-board electronics or other loads by using the thermal capacitance of the relatively cooler fuel. This FTMS allows heat rejection by burning the heated fuel or by recirculating it back into the fuel tank. The heat removed by the second heat exchanger is represented by $\dot{Q}_{\text{out}}$, but actual aircraft often do not have such an outlet for heat because they are designed to have a minimized thermal signature. Because of that, we consider cases where $\dot{Q}_{\text{out}} = 0$, as well as when we have positive value of $\dot{Q}_{\text{out}}$. There are more complex FTMS architectures to dissipate heat, such as expelling fuel out of the aircraft or having a feed tank, but we only use the recirculation-based method here.

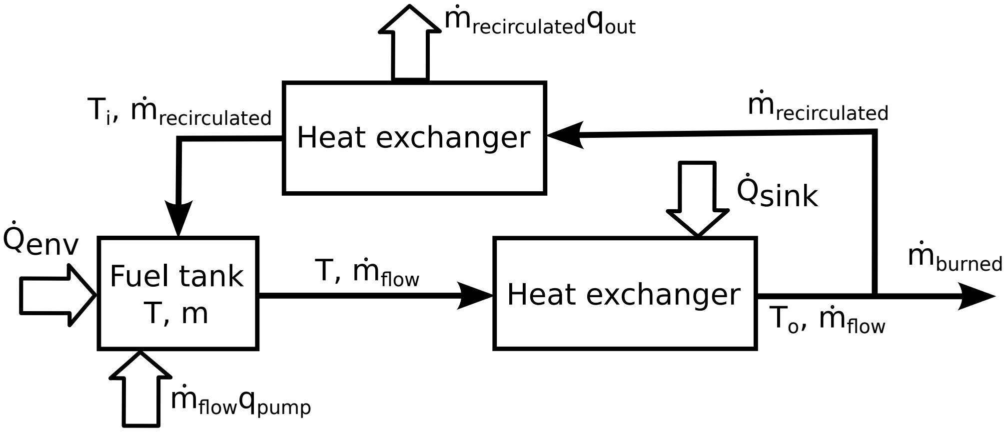


Figure 3.3: Fuel thermal management system with recirculation used for part of this work

For the purposes of this work, we do not consider the length of tubing between thermal components or the time needed for the heat to propagate to the fuel. That is, we assume that heat is instantaneously added and removed from the fuel at the prescribed rates. This allows us to greatly simplify the equations used to describe the state of heat distribution throughout the system.

The original equation relating the rate of temperature change to the heat loads and fuel flow, as presented by Alyanak and Allison [39], is

$$mc_v \frac{dT}{dt} = \dot{Q}_{\text{env}} + \left[1 - \frac{\dot{m}_{\text{burned}}}{\dot{m}_{\text{flow}}}\right] \dot{Q}_{\text{sink}} - \dot{Q}_{\text{out}}, \tag{3.1}$$

where m is the mass of fuel in the tank, c_v is the specific heat at constant volume, T is the fuel temperature in the tank, $\dot{Q}_{\text{env}}$ is the heat into the tank from the environment, $\dot{Q}_{\text{sink}}$ is the heat from the cooled systems of the aircraft, $\dot{Q}_{\text{out}}$ is the heat removed from the system via heat rejection systems, $\dot{m}_{\text{burned}}$ is the rate of fuel burned, and $\dot{m}_{\text{flow}}$ is the fuel flow rate from the tank.

We use a modified version of this equation that replaces the $\dot{Q}_{\text{out}}$ term with $(\dot{m}_{\text{flow}} - \dot{m}_{\text{burned}})q_{\text{out}}$ to account for the ability for the heat exchanger to remove more heat as the fuel mass flow rate increases. This means we can circulate more fuel to cool the tank at the cost of added pump heat and energy. Heat can only be removed from this system by burning it or by recirculating through the second heat exchanger. Additionally, we add another heat source that is dependent on the total fuel flow rate in the system. This source is representative of any heating that occurs as a function of fuel flow, thrust generated, or throttle setting, such as waste heat from gearbox bearings and shafts, oil pumps, or geared lift fans. The final equation used to obtain the temperature rate of the tank is

$$\frac{dT}{dt} = \frac{\dot{Q}_{\text{env}} + \dot{m}_{\text{flow}} q_{\text{pump}} + \left[1 - \frac{\dot{m}_{\text{burned}}}{\dot{m}_{\text{flow}}}\right] \dot{Q}_{\text{sink}} - (\dot{m}_{\text{flow}} - \dot{m}_{\text{burned}}) q_{\text{out}}}{m c_v}, \tag{3.2}$$

where $\dot{m}_{\text{flow}}$ is the total mass fuel flow in the system and q_{pump} is a coefficient that dictates how much heat is added as a function of that total mass flow.

We later refer to this single-tank system as the simplified single-tank FTMS. The next subsection introduces a more advanced architecture built using a modular thermal system framework featuring dual fuel tanks.

3.2.2 Modular fuel thermal management system framework

Although the simplified implementation presented in the previous subsection works well for that specific FTMS, using a modular thermal systems framework will allow designers to quickly prototype new architectures without needing to solve for all the ODEs of interest. In general, designers do not know *a priori* what the best architecture will be for a given aircraft application because thermal loads comes from many sources at different temperatures within an aircraft. The order of the coolant tanks, heat exchangers, sources, and sinks directly affects the thermal system, its efficacy, and weight. Also, using a different architecture is a discrete design choice, so it is challenging to examine this using gradient-based optimization. The thermal system designer must use their expertise to select architectures to study.

To model the modular thermal system, we use OpenConcept[1], which is an aircraft "conceptual design and optimization toolkit" built on the OpenMDAO framework, and includes simple, conceptual-level models of systems components [59]. OpenConcept can model conventional and electric propulsion components and recent development has extended OpenConcept's capabilities to include thermal management systems. In this work, we use the heat exchanger, transient reservoir, mixer, splitter, and thermal lifting system groups from OpenConcept. Only the heat exchanger implementation existed before this dissertation; the transient reservoir, mixer, splitter, and thermal lifting system were created as part of this dissertation work. We briefly detail the physics and implementation of these components in the following section.

3.3 Thermal subsystem implementation

3.3.1 Heat Exchanger Description

Heat exchangers provide an interface with favorable heat transfer properties between hot and cold fluids. The steady-state heat transfer of a heat exchanger can be computed via the N_{TU}-effectiveness method as

$$\begin{aligned} q &= \epsilon \frac{UA_{\text{overall}}}{N_{\text{TU}}}\left(T_{\text{in,h}} - T_{\text{in,c}}\right), \\ N_{\text{TU}} &= \frac{UA_{\text{overall}}}{C_{\text{min}}}, \\ \epsilon &= \Phi\left(N_{\text{TU}}, \frac{C_{\text{min}}}{C_{\text{max}}}\right), \end{aligned}$$

where q is the heat transfer rate, ϵ is the heat transfer effectiveness, $T_{\text{in,h/c}}$ are the fluid inlet temperatures, N_{TU} is the number of thermal units, UA_{overall} is the overall heat transfer coefficient times

---

[1] https://github.com/mdolab/openconcept/

the corresponding heat transfer area, C_{min}, C_{max} are the maximum and minimum values of the fluid heat transfer capacity $\dot{m}\,c_p$ for the hot and cold sides, and Φ is an analytical or empirical function that depends on the flow arrangement of the heat exchanger (for example, crossflow) [160].

For this study, we modeled crossflow, plate-fin heat exchangers with offset strip fin geometry. This configuration is considered a "compact" heat exchanger because it has a large surface area to volume ratio, suitable for gas-liquid or gas-gas exchange in weight-sensitive aerospace applications [160]. The offset strip fin geometry improves convective heat transfer coefficients compared to straight fins. The heat exchanger design problem involves choosing appropriate geometry and material thickness to satisfy heat transfer, pressure loss, weight, and volume requirements. Figure 3.4 illustrates a cross section of plate-fin flow channels, including the parameters width, height, fin thickness, and plate thickness. The OpenConcept [59] heat exchanger model implements the N_{TU}-ϵ equations to compute heat transfer, and uses an empirical correlation from Manglik and Bergles [161] for convective heat transfer and friction coefficients specific to the offset strip fin geometry.

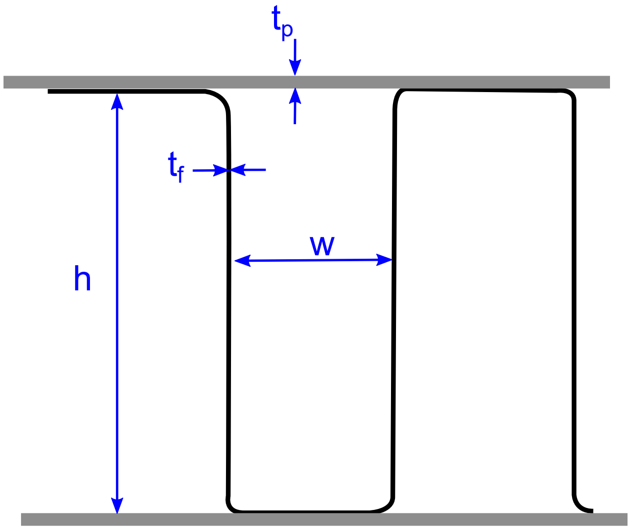


Figure 3.4: Plate-fin heat exchanger geometry

The hydraulic diameter $2wh/(w + h)$ of a typical compact heat exchanger channel is on the order of 1 mm. The overall heat exchanger area depends on the number of channels in each plate-fin layer, and the number of layers. For this study, we used values representative of a air-liquid heat exchanger, with cold-side channel width and height 1 mm, and hot-side channel width 14 mm by 1.35 mm. We did not vary the heat exchanger geometry design variables during optimization but optimizing these parameters would most likely improve system performance slightly. The heat exchanger in the bypass stream of the engine occupies the entire duct. We achieve this by setting the frontal area of the heat exchanger to be same as the bypass duct area computed by the engine design case.

3.3.2 Lifting System Description

In this work, we use shaft power from the high-pressure shaft in the engine to lift the thermal energy. This directly introduces a weak coupling between engine performance and thermal system efficiency as some energy must come off of the shaft to power the lifting system. However, the magnitude of the shaft power used in the lifting system is usually dwarfed by the total power in the engine, so this coupling is inconsequential. For a design later presented in Sec. 5.1, the lifting work taken off the shaft is about 33 hp, while the shaft is producing more than 19,000 hp, which means that the lifting system is using about 0.17% of the shaft power. The more important trade-off is between the amount of lifting work done and the heat generated during the process. Although the heat load is at a higher temperature as it is lifted, the greater total heat amount that needs to be dissipated has an effect on the optimal engine design. The increased thermal energy in the bypass airstream affects the thrust produced by the engine nozzle.

Nominally, this ACM operates as a closed Brayton cycle, where the working fluid that transmits heat is recirculated through a series of elements. In a closed Brayton cycle, the working fluid flows through a low-temperature heat exchanger and accepts heat. The fluid then enters a compressor and contracts, leading to a higher temperature working fluid. This heated fluid then flows through a high-temperature heat exchanger, where energy is dissipated. The fluid then goes through a turbine and expands before interacting with the low-temperature heat exchanger as the cycle continues. In our model, the low-temperature heat exchanger interfaces with the electronics load on the hot side and the Brayton cycle on the cold side. The high-temperature heat exchanger takes heat from the Brayton cycle on the hot side and transfers it to the bypass duct airstream. This realistically accounts for the losses in heat transfer between the fluids, since we do not assume instantaneous and lossless heat transfer in the lifting system.

Instead of explicitly modeling the Brayton cycle using pyCycle, we model it using a system of equations based on assumed friction losses and performance efficiencies to capture the relevant physics without adding unnecessary complexity to the model. These equations are formulated to take in the amount of lifting work provided by the engine shaft, and compute the heat loads that must be transferred at both the electronics and duct heat exchangers. We solve for the heat load from the lifting system that must be dissipated using the duct heat exchanger, and get

$$Q_h = \frac{W'}{1 - T_c/T_h} = \frac{\eta_p \eta_f W}{1 - T_c/T_h},$$

where Q_h is the lifted heat load, W' is the efficiency-adjusted work, T_c and T_h are the temperatures of the cooling fluid at the electronics and duct heat exchangers respectively, W is the work coming off the shaft, η_p is the shaft power transfer efficiency, and η_f is the friction loss efficiency. We can then solve for the cold-side heat load and get

$$Q_c = Q_h - (2 - \eta_f)\eta_p W.$$

Q_c is the amount of heat transferred from the actual heat load, e.g., our electronics heat output. A schematic of the work and heat flow for this simplified cycle is shown in Fig. 3.5. The losses due to friction are lost to the cold reservoir, leading to a higher Q_c value than in an ideal case. Throughout this work, we use $\eta_f = 0.95$ and $\eta_p = 0.95$ as nominal representative values. In real cases, these efficiency values might be drastically different or might vary based on operating conditions, but we use these fixed values here for model simplicity. Additionally, the simplifications made to this model are true only in certain operating conditions. These assumptions are fit for the purposes of this dissertation, but not necessarily applicable for general cases of thermal lifting systems.

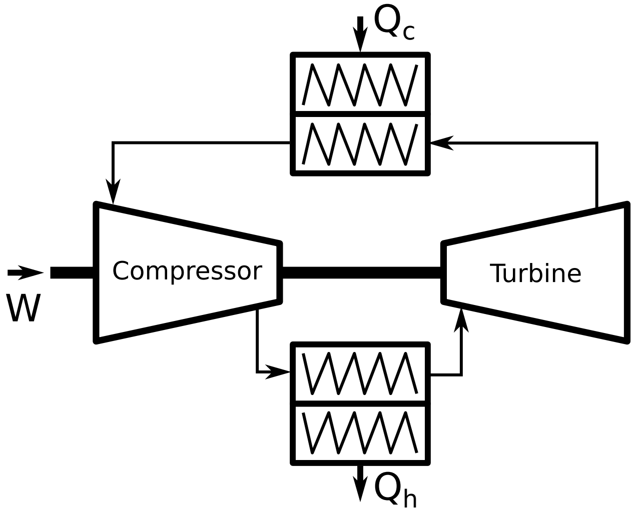


Figure 3.5: The lifting system

3.4 Construction and analysis of a modular thermal system

We construct a modular thermal system here based on a simple architecture for the ESAV aircraft, shown in Fig. 3.6. It is a dual-tank FTMS where the fuel is heated by an on-board aircraft component before either being used in the engine burner or recirculated back into the feed tank. During recirculation, the fuel goes through a ducted heat exchanger whose cold side interfaces with ducted airflow. Due to observability constraints, a ducted heat exchanger might not be possible on some stealth aircraft. Instead, the bypass air from the engine might be used to dissipate some of the heat, as we will examine in Sec. 5.1. Here, we assume that there is an inlet-duct-nozzle separate from the full engine so we do not need to model the entire engine to examine this thermal system.

The dual tank system allows the main fuel tank to not be affected by the thermal dissipation and recirculation as the recirculating fuel only enters the feed tank.

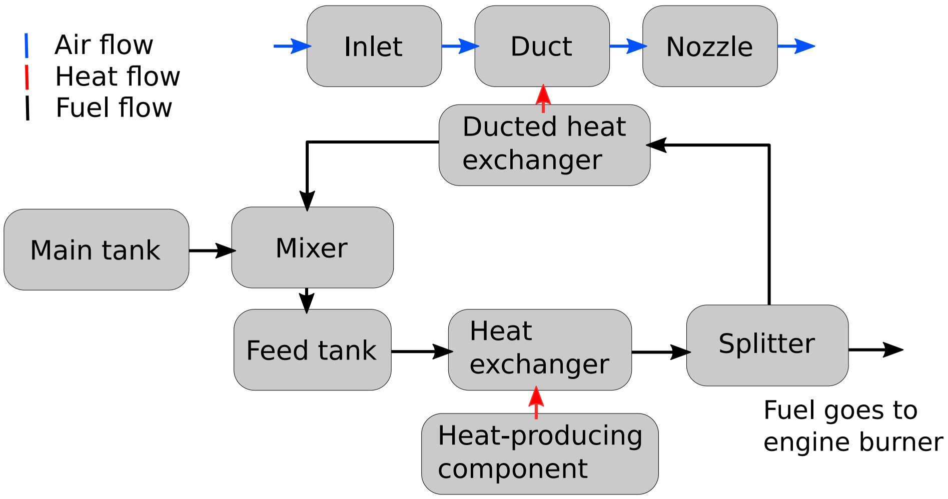


Figure 3.6: Dual tank modular FTMS used for part of this work

3.4.1 Thermal model setup

In Fig. 3.6, the inlet, duct, and nozzle are modeled using pyCycle and the rest of the thermal elements are modeled using OpenConcept. The tanks, mixer, and splitter elements were added to OpenConcept as part of this dissertation. The elements in this thermal system are able to interface directly because both pyCycle and OpenConcept are implemented in OpenMDAO. In general, the computational cost of the OpenConcept elements is much smaller than that of the pyCycle elements. The coolant loop only deals with liquid fuel, whereas the duct passageway concerns potentially supersonic air with full chemical equilibrium analysis. We could simplify the inlet, duct, and nozzle elements, but by not doing so we are able to use the pyCycle elements off-the-shelf.

To ensure this model is correctly assembled and that components are passing data correctly, we use the OpenMDAO N2 diagram. Figure 3.7 shows a zoomed view of this N2 diagram with all of

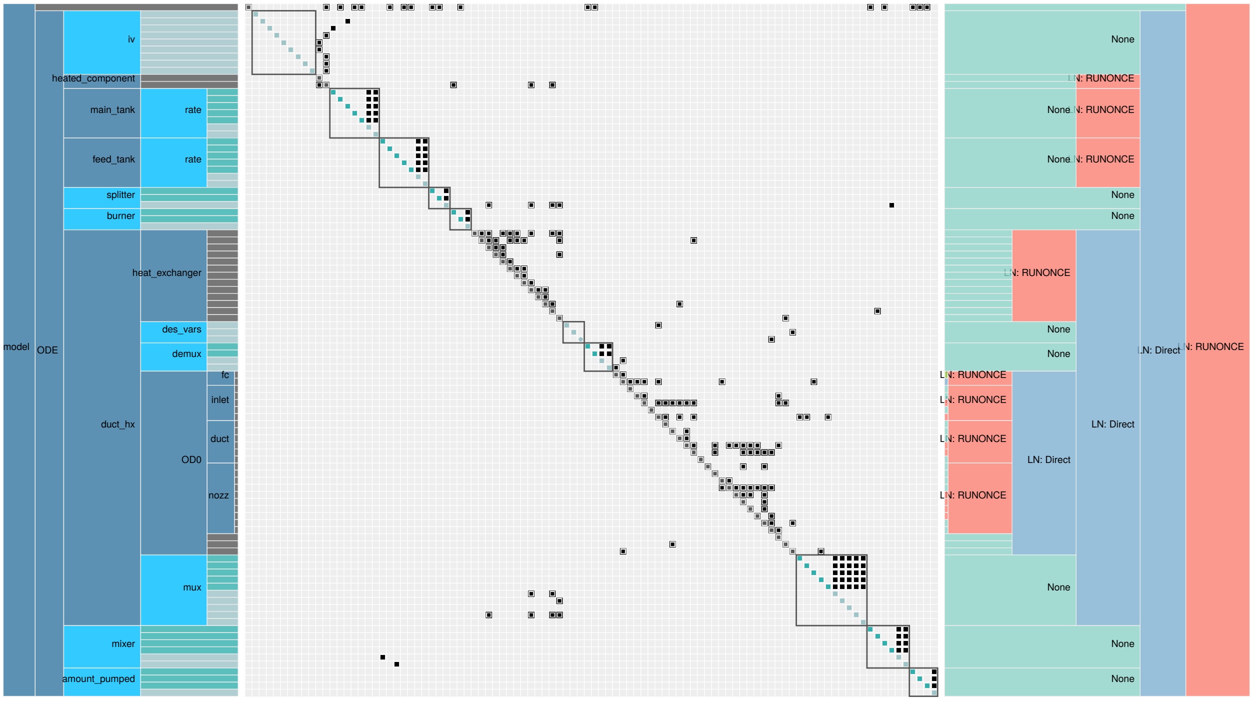


Figure 3.7: N2 diagram generated by OpenMDAO of the thermal model shown schematically in Fig. 3.6

the thermal elements shown. Although OpenConcept is vectorized so that it can handle multiple analysis points in each element, pyCycle is not. This means that we must stamp out copies of the pyCycle elements and use mux and demux components to facilitate data transfer between elements.

This thermal model has a large number of possible inputs and settings that will affect the cooling performance of the FTMS. The tank temperatures, fuel mass, heat-producing component temperatures, thermal limits, fuel flow rates through the loop and feed pipes, must all be considered when evaluating the system effectiveness. Given a real aircraft, we could setup the thermal system to match that configuration and then provide thermal data and inputs for different flight conditions. However, to keep this work available in the public domain, we use nominal values loosely based on thermal loads that a current-gen supersonic jet might see.

Many factors influence the heat transfer through the bypass duct for a fixed flight condition, including inlet and nozzle duct areas, the temperature of the coolant entering the heat exchanger, and the heat exchanger physical dimensions. The flow path areas affect how air is compressed through the bypass duct, which in turn affects the flow temperature. Higher fuel temperatures increase the amount of heat dissipated through the heat exchanger. The number of cross-flow channels, the fin thickness, height, and width, on both the hot and cold sides of the heat exchanger affects the heat dissipation ability. However, for the studies presented in this dissertation, we hold most of the physical heat exchanger design parameters fixed.

3.4.2 Analysis sweeps of thermal design space

To better understand this dual tank thermal model, this section contains parametric analysis studies across many flight conditions and fuel flows. As mentioned before, there are many parameters that must be set within the model. The relevant values used in these studies are provided in Table 3.1. For the heat exchangers, we assume aluminum fins and casings and obtain fin and channel dimensions from a representative aircraft air-liquid heat exchanger. The coolant used throughout this thermal work and in the engine burner is Jet-A fuel. We assume standard atmospheric air enters the inlet and flows through the bypass duct. Most of these values are also used later in Chapter 5 for the fully coupled path-dependent problem.

We could vary any one of these parameters to better study the design space. Some of interest would be the Mach and altitude of the flight condition for the ducted heat exchanger, the fuel flow through the heat exchangers, the heat exchanger sizing, and the component thermal load. The main goal in establishing this thermal system is to use it in a large-scale mission optimization. This means the efficacy of the thermal system at different flight conditions as well as at different fuel flow rates will be important to understand. Studying the thermal system alone and gaining an intuition for its trends will allow us to better interpret the results presented in Chapter 5.

This section presents three parametric sweeps and discusses their trends. Specifically, we vary Mach, altitude, and fuel flow rate through the recirculatory heat exchanger. In each case, all the other parameters are held fixed and only the parameter of interest varies. The nominal fixed values are those shown in Table 3.1. For each of the sweeps, we show the values listed in Table 3.2. All gas temperatures shown are the static temperature, not the total temperature.

Figure 3.8 shows the altitude sweep for this thermal system. At low altitudes, the ducted heat exchanger cannot remove thermal energy from the fuel because the temperature of the air in the duct is too high. As the freestream temperature decreases, the heat exchanger efficacy increases. However, at around 37,000 feet, the freestream temperature becomes constant with altitude while air density continues to decrease. This produces the kink in most of the outputs, and it negatively affects the ducted heat exchanger's performance. We see $\dot{Q}$ begins to decrease after that change in temperature.

Figure 3.9 shows a similar sweep for varying Mach, or speed of air entering the duct inlet. Again, all other parameters are held constant. As Mach increases, $\dot{Q}$ decreases due to the increased duct temperature due to the compressed air. The air velocity in the duct actually increases, which intuitively might lead to better heat exchange, but the increased temperature outweighs that effect. In the full problem, the optimizer might prefer to fly at a faster speed to complete the mission more quickly, but the heat dissipation ability decreases with increased Mach. These kinds of trade-offs are non-intuitive and would not be immediately evident without examining each discipline's design space in depth before moving on to the coupled problem.

The last sweep we present is shown in Fig. 3.10, where we vary the amount of fuel mass flow rate through the recirculatory heat exchanger. Intuitively, increased fuel flow through the heat exchanger should increase $\dot{Q}$, and this is seen, however the heat transfer amount quickly levels off as we increase $\dot{m}_{\text{recirculated}}$. This is because the heat exchanger geometry is fixed, and that geometry has limits to the amount of heat that can be transferred, regardless of the flow rate.

In the first two sweeps, the trends for ΔT_{feed} and $T_{\text{out,fuel}}$ closely followed each other, but that is not true for Fig. 3.10. As the flow rate increases, the total thermal energy added to the feed tank increases, even though the temperature of the fuel exiting the heat exchanger remains relatively constant. So, there could be disadvantages to increasing the fuel flow rate through the ducted heat

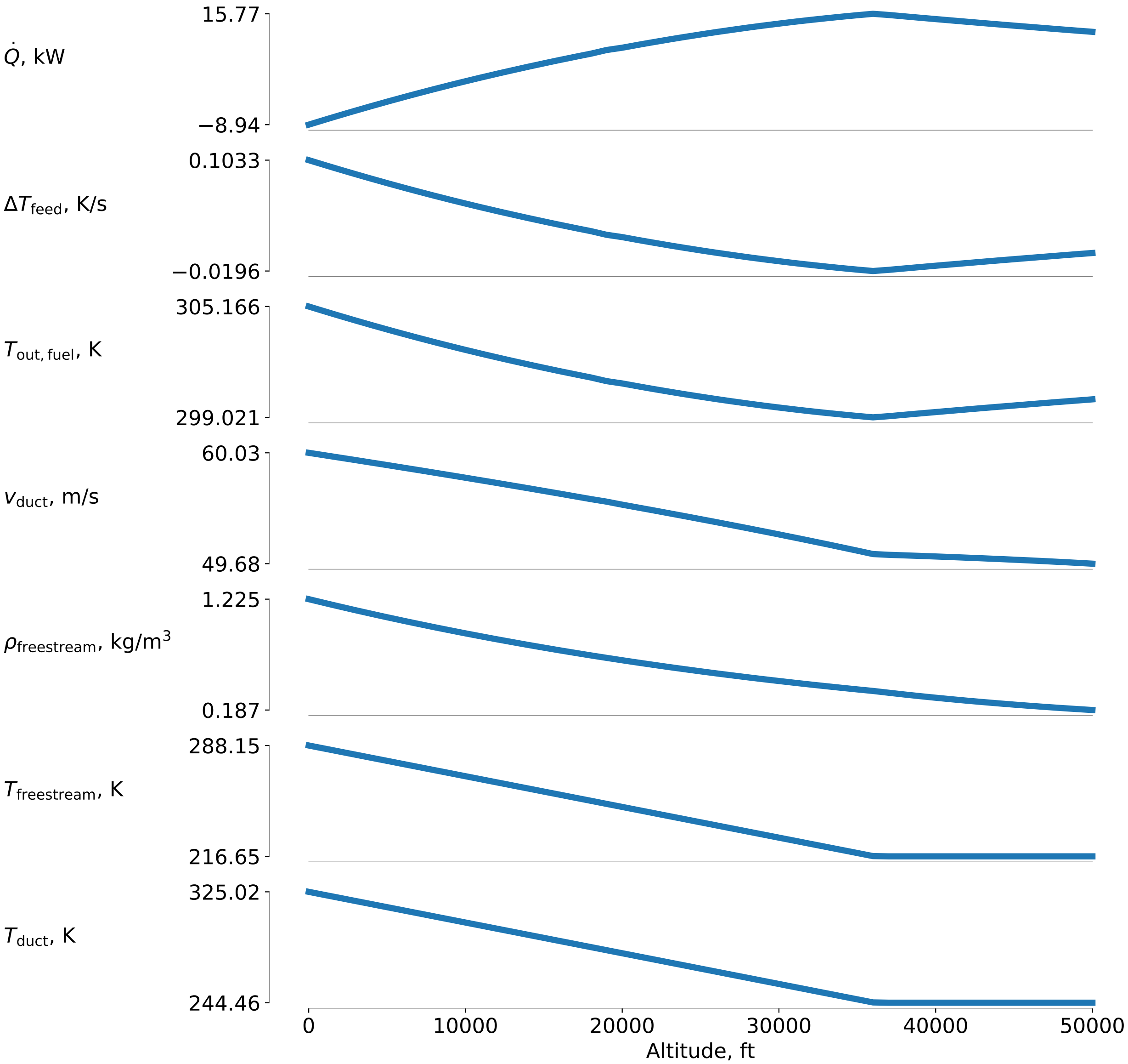


Figure 3.8: Parametric sweep through altitude for the dual tank thermal system

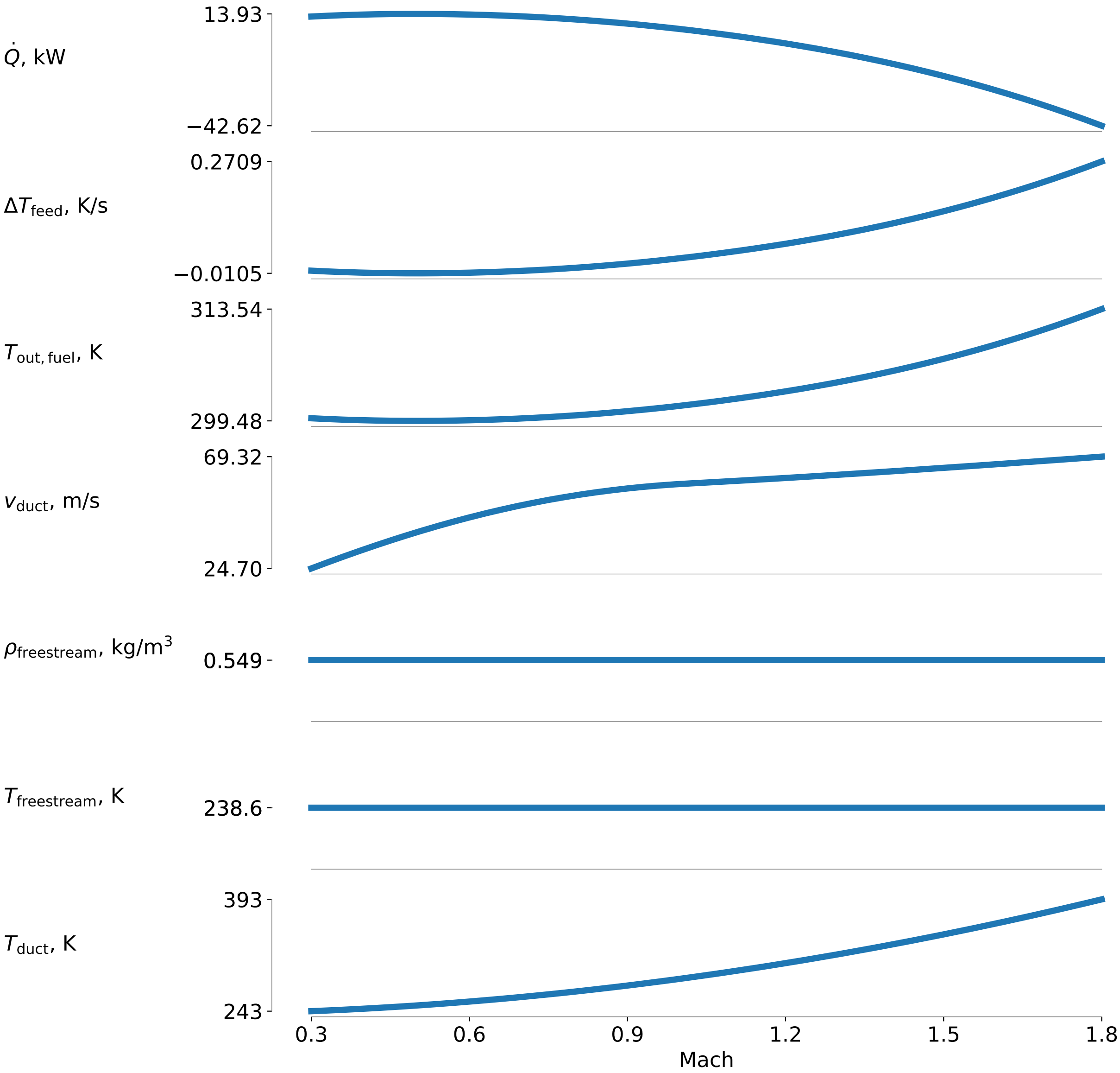


Figure 3.9: Parametric sweep through Mach for the dual tank thermal system

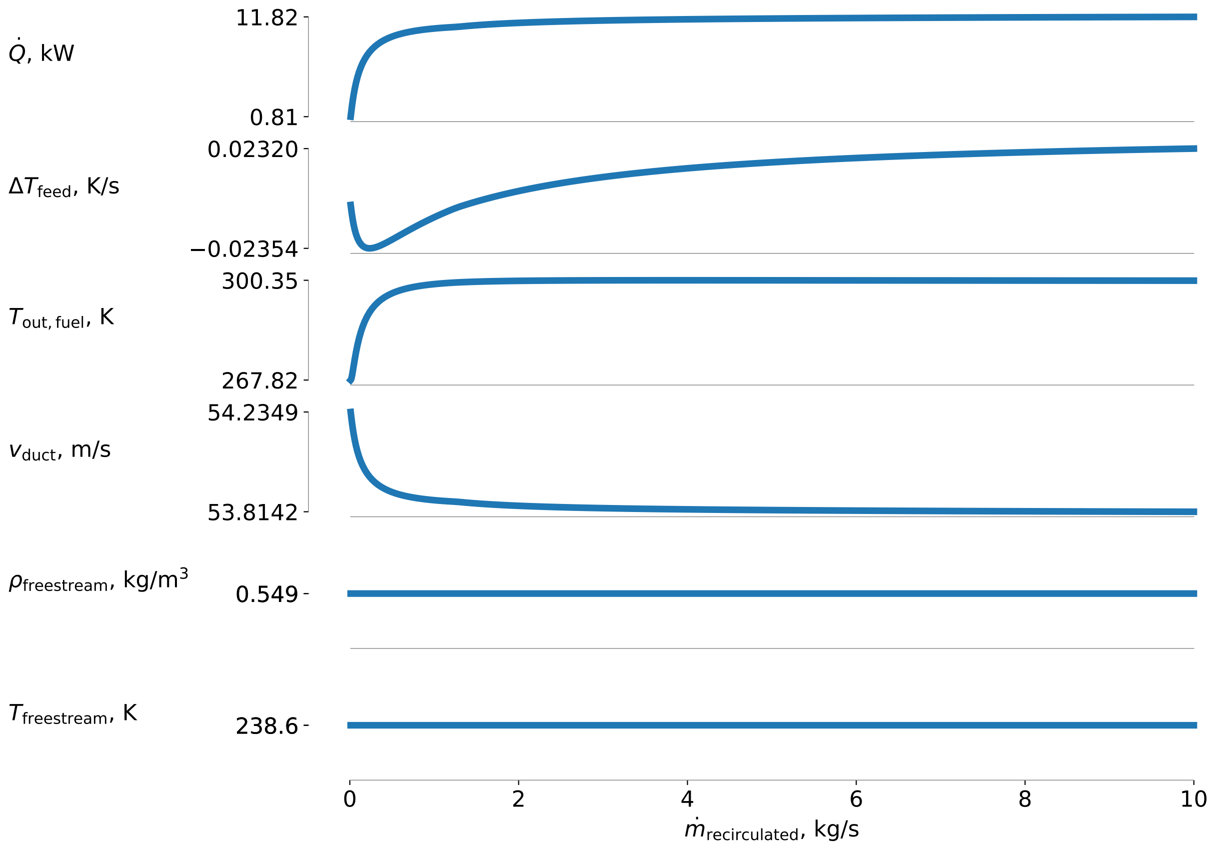


Figure 3.10: Parametric sweep through $\dot{m}_{\text{recirculated}}$ for the dual tank thermal system

exchanger. Based on the problem formulation, design variables, and flight conditions, there is a sweet spot for the amount of fuel pumped through the recirculatory heat exchanger.

Table 3.1: Specifications and parameter values for the dual tank thermal system.

Parameter	Value	Units	Comments
Component temperature	350	K	
Initial tank temperatures	300	K	
Initial main tank fuel mass	1000	kg	
Initial feed tank fuel mass	100	kg	
$\dot{m}_{\text{transfer}}$	3.	kg/s	
$\dot{m}_{\text{recirculated}}$	1.	kg/s	
$\dot{m}_{\text{burned}}$	1.	kg/s	
Mach	0.8	–	low-altitude cruise
Altitude	25,000	feet	low-altitude cruise
Specific heat of component	921	J/kg/K	aluminum
Specific heat of fuel	2010	J/kg/K	Jet-A [162]
Fuel density	800.	kg/m^3	Jet-A [162]
Fuel thermal conductivity	0.110	W/m/K	Jet-A [162]
Fuel viscosity	0.000704	kg/m/s	Jet-A [162]
Component mass	12	kg	small avionics package
Component $\dot{Q}$	20	kW	small avionics package
Channel width	1.0	mm	
Channel height	20.	mm	
Channel length	0.2	mm	
Case thickness	2.0	mm	
Fin thickness	0.102	mm	
Plate thickness	0.2	mm	
Material thermal conductivity	190	W/m/K	aluminum
Material density	2700	kg/m^3	aluminum
Number of fins long, cold side	3	–	
Number of channels wide, cold side	200	–	
Number of hot/cold stacks	15	–	
Channel height, cold side	14	mm	
Channel width, cold side	1.35	mm	
Fin length, cold side	6	mm	
Specific heat, cold side	1005	J/kg/K	air
Thermal conductivity, cold side	0.02596	W/m/K	air
Viscosity, cold side	0.00001789	kg/m/s	air
Channel height, hot side	1	mm	
Channel width, hot side	1	mm	
Fin length, hot side	6	mm	
Specific heat, hot side	2010	J/kg/K	Jet-A [162]
Thermal conductivity, hot side	0.11	J/kg/K	Jet-A [162]
Duct inlet area	0.0645	m^2	
Nozzle throat area	0.0194	m^2	

Table 3.2: Parameters shown in the thermal system sweeps.

Parameter	Units	Comments
Altitude	feet	used for the duct inlet air
Mach	–	used for the duct inlet air
$\dot{m}_{\text{recirculated}}$	kg/s	fuel flow rate through the recirculatory heat exchanger
$\dot{Q}$	kW	heat transfer rate in recirculatory heat exchanger, positive removes heat
ΔT_{feed}	K/s	rate of change of temperature of fuel in the feed tank
$T_{\text{out,fuel}}$	K	temperature of the fuel leaving the recirculatory heat exchanger
v_{duct}	m/s	speed of air through the duct for the recirculatory heat exchanger
$\rho_{\text{freestream}}$	kg/m^3	freestream air density
$T_{\text{freestream}}$	K	freestream air temperature
T_{duct}	K	air temperature entering the bypass duct

CHAPTER 4

Engine Optimization

4.1 Review of engine optimization

Since the invention of the gas turbine engine in the 1930s, researchers have been developing cycle analysis tools [163–165]. At first, these calculations were done by hand, but as engines became more complex and computing power increased, engine cycle analysis codes became more popular. Hendricks and Gray [87] detail the history of cycle analysis tools used to design engines from the 1940s to now. Their paper includes a figure, reproduced in Fig. 4.1, that shows which cycle analysis tools were prominent throughout history. This dissertation focuses mostly on the "Vehicle System Multidisciplinary Tool" pyCycle, with some comparisons to the "Engine System Multidisciplinary Tool" NPSS, but the history of cycle analysis is important to consider for added context and appreciation for the capabilities we have today. In this section, we will first discuss cycle analysis, then expand that discussion to include engine design and optimization.

To computationally model engines, methods range from using empirically fit rubber engine models used in preliminary aircraft design to performing unsteady CFD on engine components. These engine models are often verified and validated using experimental results, and some researchers explored multifidelity models that combined multiple analysis types for increased accuracy [166–168]. A popular middle-ground is one-dimensional methods, which are used in the industry standard NPSS and NASA's pyCycle. These methods model engine elements as lumped

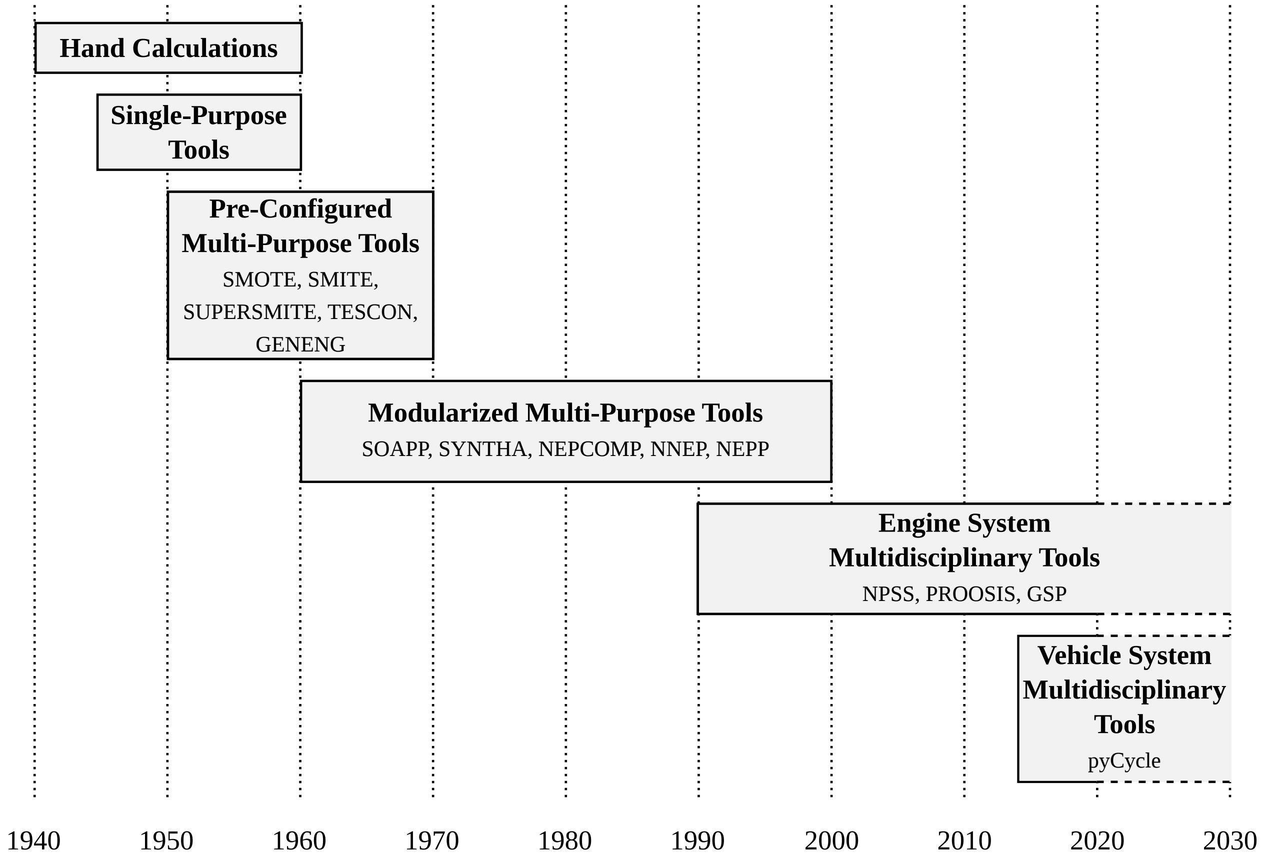


Figure 4.1: History of gas turbine cycle analysis eras with example tools, adapted from Hendricks and Gray [87]

components with no internal axial dimensionality with element properties coming from "maps" that relate flow properties to efficiency.

Only recently in the 2000s have propulsion systems been optimized in an MDO context. Grönstedt and Wallin [169] compared both gradient-free and gradient-based optimization methods for a turbofan engine design, but only looked at the engine and not other aircraft systems. Simmons [145] examined a complex variable cycle 3-stream engine in detail using gradient-free optimizers, but only examined engine modeling and not other aircraft systems. Allison et al. [75] performed design studies of the propulsion system considering thermal dissipation ability, though they sampled the design space by hand without using an optimizer.

Researchers at NASA Glenn Research Center have produced a large amount of work on engine and air vehicle design, especially using gradient-based optimization. Heath et al. [170] optimized the inlet shape on a supersonic aircraft using adjoint gradient-based optimization. Hendricks [171] exhaustively examined the multi-level multi-design point approach for design of gas turbine engines. He used a gradient-free optimizer in OpenMDAO with multiple different models to compare optimal designs. Hearn et al. [88] published the first paper on pyCycle, which detailed its analytic derivatives and gradient-based capabilities. Gray et al. [81–83, 172] have presented multiple studies on the design of boundary layer ingestion propulsors, which are coupled aero-propulsive systems. More recently, Hendricks et al. [47] modeled the gas turbine engine for a tiltwing commuter aircraft in a large-scale MDO problem.

Although there are many published studies on engine design, there is a large amount of information that is proprietary. Because of this, some portions of engine design involve word-of-mouth and practical knowledge. Major engine designers and manufacturers, like GE, Rolls Royce, and Pratt and Whitney, keep important information internal to the company, so much so that collaborators and purchasers do not get direct access to all performance and testing information. This is especially true for supersonic engines due to their use in military applications.

In this chapter, we detail our approach to modeling engines and improvements to the process as a result of this dissertation. We close with results from a multipoint supersonic engine optimization problem.

4.2 Engine modeling tool: pyCycle

The state-of-the-art propulsion modeling tool, NPSS, solves a large nonlinear system of equations that represent the physics from 1-D propulsion models. Despite not providing derivatives, NPSS has been used for gradient-based optimization by finite differencing across the model. However, researchers have noted accuracy problems when using finite differences across NPSS [87].

The desire to perform multidisciplinary gradient-based optimization of engines drove the development of pyCycle [87, 173]. pyCycle is a NASA tool that models the same physics as NPSS, but provides analytic derivatives throughout the entire process. Additionally, pyCycle is written in OpenMDAO, which helps support modular model construction and integration into larger MDO problems. pyCycle uses chemical equilibrium analysis [86] for its underlying thermodynamic library and uses a system of nested Newton solvers to converge complex engine cycles efficiently. pyCycle has been used for propulsion-mission optimization [44] and boundary layer ingestion studies [81, 83, 172]. An overview of the optimization problems that can be formulated by pyCycle is shown in Fig. 4.2.

pyCycle has been verified against the industry standard tool, NPSS, in a series of papers [86, 87]. They used a systematic process of evaluating the pyCycle code, starting first in small unit-test sized chunks and comparing those results to NPSS at each step. Gray et al. [86] first verified the computation of thermodynamic properties required at flow stations throughout the cycle by comparing them to the chemical equilibrium analysis (CEA) thermodynamic calculations used in NPSS. They found that the mean error across all cases tested was 0.03 % and the maximum error was 0.52 % [86]. They went on to say this “demonstrates a strong agreement between CEA and the new code, and it verifies its predictions for cycle analysis applications” [86]. Next, Hendricks

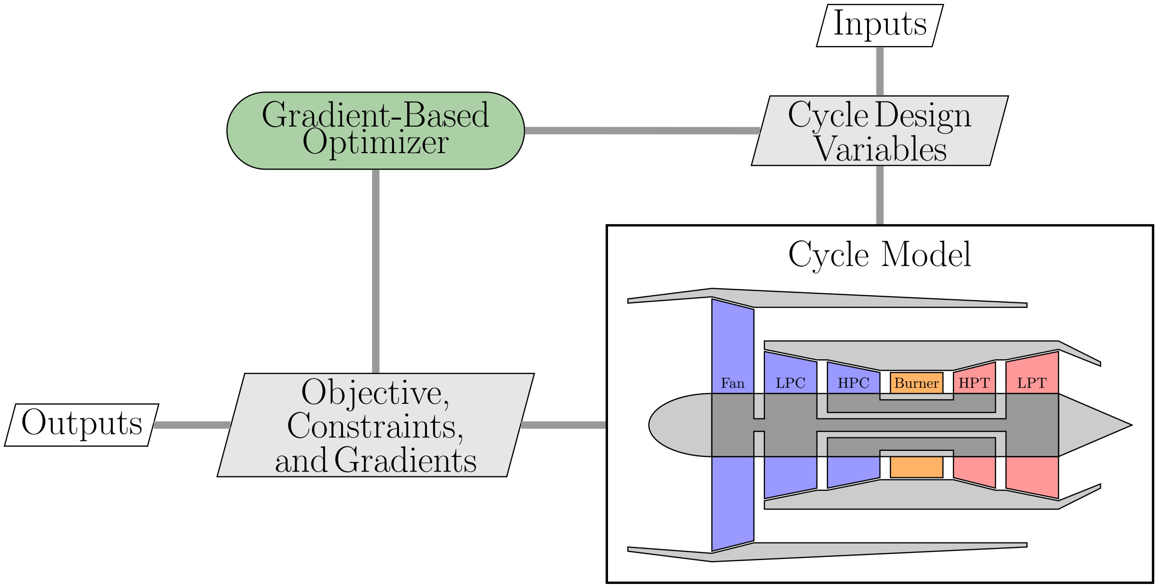


Figure 4.2: Schematic view of a pyCycle optimization, adapted from Hendricks and Gray [87].

and Gray developed and verified physical engine components, such as inlets, fans, burners, and compressors, against NPSS, and found good agreement between the two codes [87].

To verify the full capability of pyCycle, Hendricks and Gray combined the individual cycle elements together to create a complete engine model to compare to NPSS [87]. They constructed the N+3 engine, a NASA-developed high-bypass turbofan engine representative of 2035 technology [174]. To perform this engine model verification test, the pyCycle model was run with the same inputs, states, and residuals as the model originally developed by Jones [175] in NPSS. Using these identical setups, pyCycle and NPSS results showed excellent agreement. Specifically, they compared the thermodynamic properties at each flow station in the engine. The maximum relative error for any output was approximately 0.03%, indicating good agreement. Hendricks and Gray also compared overall engine performance metrics, and found that the largest relative error was less than 0.02%.

In addition to verifying the thermodynamic properties at each flow station for this test, Hendricks and Gray [87] also compared the overall performance metrics computed by the pyCycle and NPSS models. They found good agreement across all four design points, with a relative error of less than 0.02% for most values. They did find a larger relative error of 0.15 % for the sea-level static (SLS) ram drag, which shows that there are some minor differences between the codes when operating at very low Mach numbers, such as Mach 0.0001. They determined that given the solver tolerances within the models and the engineering precision of the models, these levels of error were acceptable. Directly quoting their paper, they say “this level of error provides provide strong evidence to verify that pyCycle components and overall engine models properly model thermodynamic cycles” [87]. They went on to say that “the pyCycle code was deemed sufficiently verified against the current state of the art code NPSS to proceed with evaluating the analytic derivative calculations and the use of pyCycle for design optimization” [87].

Because pyCycle has been rigorously verified against NPSS, we do not compare between the codes here. Instead, we focus on constructing and optimizing models using pyCycle in novel and robust manner. The challenges we faced, our implemented solutions, and then the optimized results are detailed in the immediately following sections.

4.3 Descriptions of engine models used in this work

Throughout this dissertation, multiple engine models are presented. This section details the similarities and differences between those models to make it clear which physics are being considered for a given application case. We could have used a single engine model, but this would require a complex and sophisticated model to study each of the trade-offs presented here. Although possible, that would greatly add to the developer time cost of setting up proper optimization problems in each application case.

All of the engine models used in this work are supersonic-capable two-shaft turbofan engines, with different additional physics considered. Table 4.1 details what physical features are included

Table 4.1: Different engine models used in this work

	Sec. 4.6	Sec. 5.1	Sec. 2.4, Sec. 5.3.3
Two-shaft turbofan	X	X	X
Mixed-flow	X		X
Installation effects		X	X
Double bypass		X	
Heat exchanger		X	
Variable cycle	X		

with each of the three engine models, and we now describe what these physical features entail.

If the engine is mixed-flow, it means that the flow exiting the core and the bypass stream are combined in a mixer before leaving the engine through a single nozzle. The alternative setup is called split-flow, which has the core and bypass flows separated and exiting the engine through two nozzles. Installation effects refers to the inclusion of an inexpensive model that captures the drag contributions from the inlet. The theory and implementation of this model are detailed in Sec. 4.5. Double bypass means the engine has three total streams: the core, one bypass stream, and an additional bypass stream. The existence of a third, separate stream is motivated by the inclusion of a bypass-duct heat exchanger, which allows the bypass flow to accept waste heat from the aircraft. Lastly, variable cycle refers to the engine's ability to tailor performance at different flight conditions by varying internal geometry within the engine, such as the inlet guide vanes (IGV)'s angles or a variable area bypass injector (VABI).

More details about the implementation of these engine models will be provided in their corresponding sections.

4.4 Solutions to challenges faced when performing engine optimization

Engine modeling is notoriously difficult. Much prior work has discussed the difficulties they experienced while constructing and analyzing numerical engine models [37, 75, 169, 171]. When performing gradient-based engine optimization, these difficulties are magnified.

We have identified the two most important challenges that impact engine optimization as:

- The complex physics within an engine model require highly accurate convergence
- Engine models are challenging to converge without good initial guesses

Both of these challenges greatly hinders the efficacy of gradient-based engine optimization, especially if they are not addressed. Because we are changing the design of the engine at each optimization iteration, we need to be able to converge a new engine model reliably and across different flight conditions. Additionally, we need the design space to be smooth so the optimizer does not get stuck in function noise, so we need to converge the model to a tight tolerance.

We develop and implement a combination of solutions to address these challenges and enabled gradient-based engine optimization:

- We use a nested gradient-based solver structure with line searches
- We train and query a surrogate model of the engine properties at different flight conditions to provide good guesses for subsequent analyses

The use of nested solvers to converge the engine model was initially developed at NASA simultaneously as the work in this dissertation was being completed, whereas the surrogate model usage to provide good initial conditions was developed specifically as part of this dissertation. We will now go into more detail about each one of these challenges and our solutions in the following subsections.

4.4.1 Nested solver hierarchy and line search technique

In physical terms, an engine is a complex combination of elements interacting together. This leads to a large number of implicit relationships, as each element affects one another, both up and downstream within the engine. Because of this, we have a highly coupled model where information must be passed considering both the feed-forward and feedback coupling. One of the main features of OpenMDAO is the ability to easily construct and solve highly complex models by connecting the groups and components together, regardless of the direction of coupling. pyCycle models are perfect examples of complex systems that OpenMDAO handles well whereas other frameworks would not be as flexible or powerful.

With the engine model constructed and variables linked together, we need to prescribe the solver structure so that the analysis can efficiently converge considering all of the coupled interactions. To do that, we intelligently select where to place nonlinear solvers within the model to decrease computational cost while increasing robustness. For example, there is a Newton solver to converge the static flow properties at the output of the mixer, then these converged properties are used to converge an impulse balance for the mixer element in an intermediate level Newton solver, then those converged values are used in the top level Newton solver for the entire engine model. This is just one specific instance of the nested solver setup within the engine model.

On each nonlinear solver, we can choose to use a line search method to increase robustness of the solver while also increasing computational cost. The line search modifies the Newton method by not always taking a full step in the step direction. Instead, it queries the nonlinear system at points in the line created by the step direction to find the next iteration in the nonlinear solver. OpenMDAO includes a few line search methods, but we used two in this work. The first, 'BoundsEnforceLS' within OpenMDAO, ensures that the variable bounds are respected as the nonlinear system converges. This can help prevent the Newton solver from navigating into a divergent or non-physical region. Specifically, the line search would backtrack the states to an allowed region and query the system again. The second method, 'ArmijoGoldsteinLS', enforces the bounds as well, but also backtracks until a termination criteria is met. This criteria can be either the Armijo-Goldstein condition, which checks for a sufficient decrease from the initial point by measuring the slope, or an iteration limit for the line search. The Armijo-Goldstein line search is useful for converging stiff systems or when you do not have a great initial guess yet.

Sometimes even using a line search is not enough to converge a nonlinear system. In the event of convergence failure, the model designer needs to choose how the system and optimizer should proceed. For the cases presented here, all solvers below the top level do not throw an error if they do not converge. This is so if a subpart of the model is not converged, the nonlinear solver one level up may be able to converge the larger system. If the top level nonlinear solver fails to converge, it passes a failure flag up to the optimizer so the optimizer can backtrack.

The solver and line search hierarchy detailed here is different than what is in NPSS. NPSS uses a hybrid Newton-Broyden solver and employs a pseudo-line search method that prevents steps larger than 10 % from the current design for any individual variable. Additionally, NPSS uses a reduced-space Newton method instead of the hierarchical nested solvers that pyCycle uses. All of this contributes to pyCycle's greater suitability for robust gradient-based engine optimization as compared to NPSS.

4.4.2 Using surrogate models to increase engine analysis robustness

We need to give the engine model a good set of initial guesses for its state values because Newton-based solvers are extremely sensitive to the solver starting point. If we simply ran the engine model without feeding it reasonable initial guesses, it would diverge on the first Newton iteration. We could hardcode guesses into the model, but that would not account for the differences in converged states of the engine at different flight conditions, let alone if we change the engine design. To partially alleviate this problem, we worked to make the engine model robust across a range of flight conditions, as discussed in Sec. 4.4.1. We also implemented surrogate-enhanced guess tables, which we now detail in this subsection.

To evaluate an engine model across a variety of flight conditions, normally it requires an iterative process. The engine designer first converges the model at one Mach-altitude-throttle point, then uses the results from that as the initial guess for a nearby point. Usually this requires stepping across the flight condition space in relatively small steps, on the order of 3000 feet or Mach 0.1 at a time. This is because the converged engine state for a subsonic low-altitude case is radically different than that of a high-altitude supersonic case, for instance. Without stepping to the next evaluation point from a successfully converged one, engine designers normally cannot analyze another point far in the flight condition space.

To remedy this, we developed a process to generate surrogate-enhanced guess tables that can provide good initial guesses to the engine model that are independent of the model's previous state. The process is outlined below for a given engine design:

1. Run the engine at the fixed design point and an off-design point
2. Save the resulting states and performance data
3. Train the surrogate model using the saved states data
4. Repeat steps 1-3 for all points in the flight envelope by varying the off-design analysis point flight conditions

After going through this process across the range of conditions in the flight envelope, we now have a surrogate model that we can query to give good initial guesses to the engine model. This allows us to query the engine model at any point in the flight envelope easily, which is especially important if we include engine analysis in a mission optimization problem. Even if we change the static design of the engine, this enhanced guess table gives reasonably good initial guesses, allowing us to also optimize the engine.

Figure 4.3 shows an example set of points that we ran in the Mach-altitude space to construct a surrogate model. Depending on the engine model, some areas of the flight envelope may require

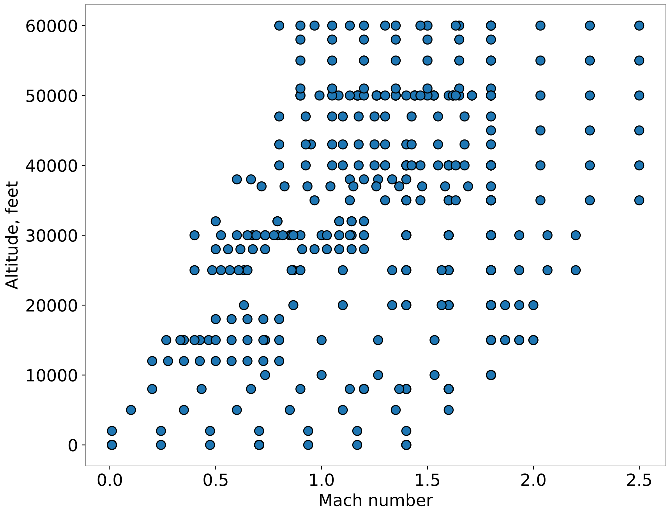


Figure 4.3: We ran a nominal model at a set design thrust across the Mach-altitude space to determine good initial guesses for the model presented in Sec. 4.6.

more training points to get a reasonable representation of the states there. This is especially true when the analysis point pushes the engine close to the edge of an element map. Once an engine model is attempting to query element performance off of the map, the results are no longer valid as we lack element information there. Thus, these maps are partially what define the limits that an engine model can compute. Figure 4.3 also shows a reasonable flight envelope for a supersonic engine based on the extents of the training points. This particular guess table is used to help enable the multipoint variable cycle engine (VCE) engine optimization presented in Sec. 4.6.

This enhanced guess table actually serves dual purposes in this dissertation. In addition to providing good initial guesses for the engine model, the surrogate model also provides performance information for the engine based on the flight condition and throttle. This allows us to query the surrogate model instead of the actual engine model when performing mission analysis and optimization. Because the surrogate model and the actual engine model have the same inputs and outputs for the mission problem, we can choose to use either method within a given model with ease.

4.5 Inexpensively modeling engine installation drag

The performance of an engine in a test stand differs greatly than when installed on an actual aircraft. Because of this, it is important to model installation effects when designing an engine. In the best-case scenario, we are able to model the entire aircraft geometry and the engine performance simultaneously, along with other relevant aircraft subsystems. However, doing that is extremely computationally expensive. Some researchers have modeled and designed the inlet geometry of a supersonic engine using CFD [170, 176], though using CFD greatly increases the computational cost.

Inlet effects are especially important for supersonic aircraft due to drag contributions from bleed, bypass, and spillage effects. Supersonic engine inlets can have fixed or variable geometry, further complicating the design process. Instead of explicitly modeling the inlet geometry, we can gather data from other sources, such as CFD or experiments. We can then use this data and obtain the performance effects using much less computational power, which is detailed in this section.

Here, we introduce a novel method to obtain performance data of a fixed-geometry supersonic inlet. The method presented here is adapted from internal NASA Glenn Research Center design studies. They previously performed a design of experiments study to quantify the drag losses of a supersonic inlet at different Mach numbers and air capture areas. We then adapted this data and constructed multiple surrogate models that output the bleed, spillage, and bypass drag contributions. Using this surrogate model, we can inexpensively include installation effects in our engine and mission optimization problems. Other researchers, such as Muller and Gasko [177] and Slater [178] have characterized or designed supersonic inlets, but did not also design the engine simultaneously, as we are doing here.

4.5.1 Implementation of inexpensive installation effects model

The initial data set from NASA Glenn Research Center came from a series of CFD runs on a fixed-geometry supersonic inlet, the "NVSTOL" Mach 2.0. This inlet is an axisymmetric, external compression nozzle with a design Mach number of 2.0 [179]. They ran this geometry through many flights conditions in a full factorial design of experiments (DOE), with Mach ranging from 0.0 to 2.0, and sweeps of capture areas A_o/A_c depending on Mach. At each of these points, they obtained $C_{D,\text{spill}}$, $C_{D,\text{bypass}}$, and $C_{D,\text{bleed}}$, to obtain the total drag contribution from the inlet. Figure 4.4 shows the raw data from the NASA analyses. In this subsection, we detail the novel implementation of a surrogate-based method to use this data in engine design optimization.

Figure 4.5 shows the N2 diagram for the supersonic installation drag computation group whereas Fig. 4.6 shows the physical representation of the inlet geometry. The inputs to the installation group are Mach number, air density, freestream velocity, the capture area of the inlet A_c, and the inlet flow available at the engine face A_{eng}. Using these inputs and the computed inlet data, we can obtain the drag force terms due to the inlet. Throughout this process, most of the area information for the inlet is expressed a ratio of A_c. This normalization allows the drag coefficient terms to be generally valid for any A_c, which might be a design variable, especially if considering the propulsion and aerodynamic shape simultaneously.

Using the data from the NASA analyses, we construct three surrogate models to represent the C_D trends. Each surrogate model has two inputs, the Mach number and A_o/A_c, and outputs the C_D value due to spillage, bleed, and bypass. This allows us to determine the drag contributions in a continuous and differentiable manner. Some portions of the raw data show C_1 discontinuities in Fig. 4.4, which also motivates our use of surrogate modeling to enable gradient-based optimization.

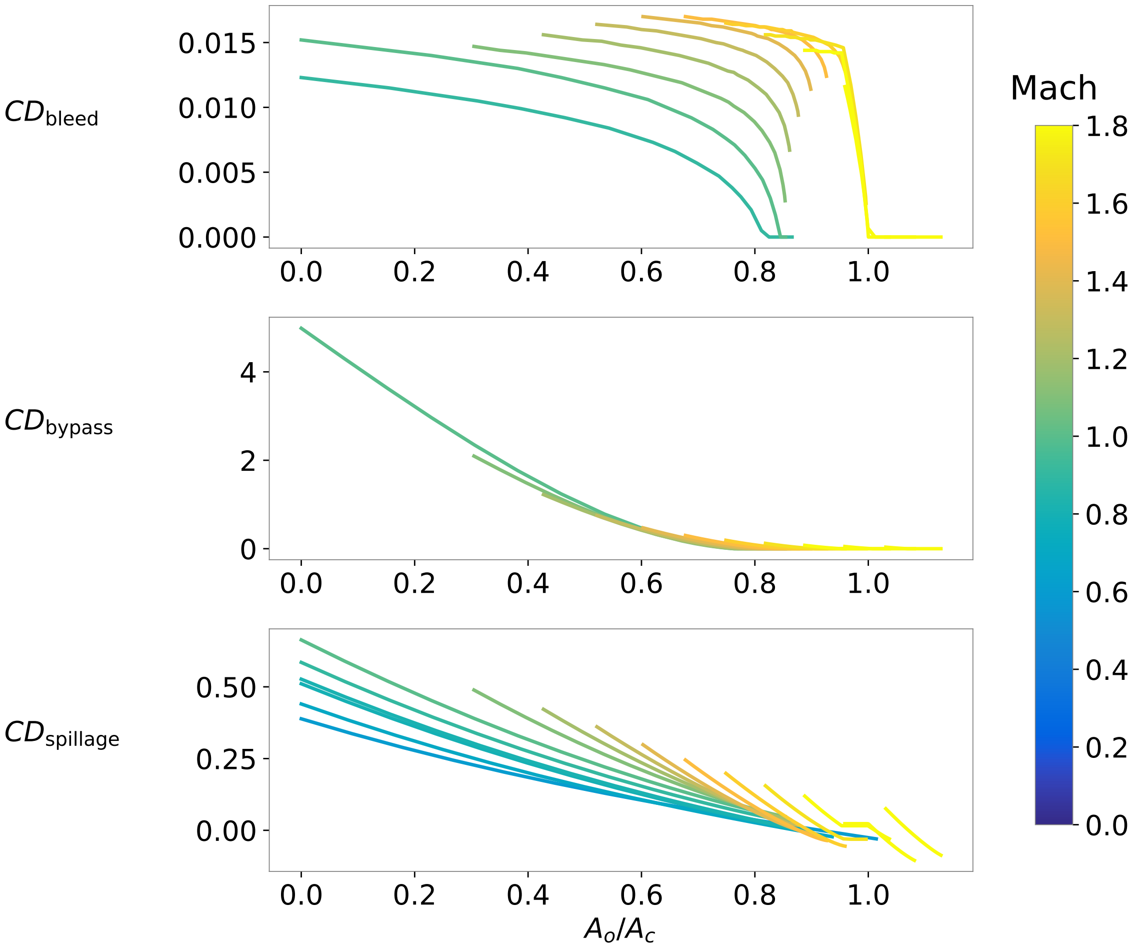


Figure 4.4: Raw C_D data from NASA inlet analyses. Values for Mach numbers where most entries have zero drag are not shown to reduce visual complexity.

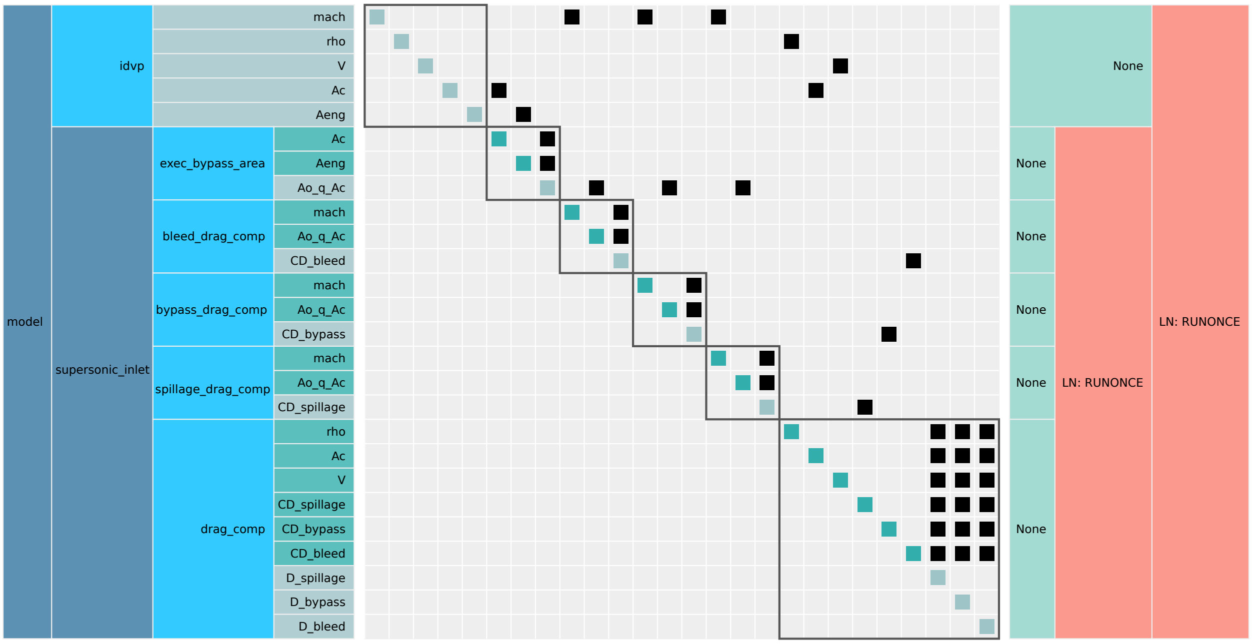


Figure 4.5: The N2 visualization of the supersonic installation implementation

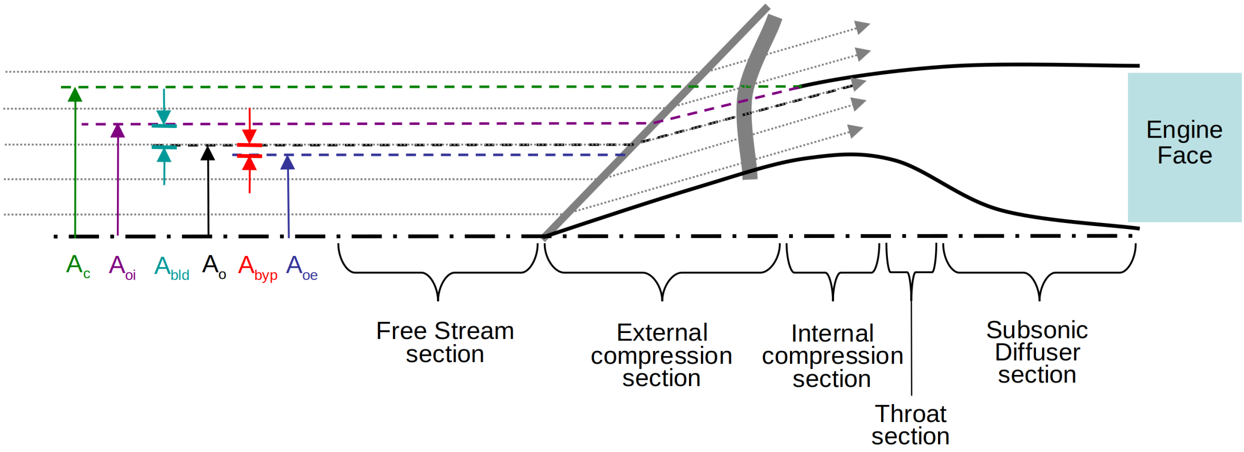


Figure 4.6: Physical representation of the inlet geometry

4.5.2 Parametric studies of installation effects

With the surrogate models constructed, parametric studies can show how the installation effects vary across flight conditions. Figure 4.7 shows drag trends across a Mach number sweep for the inlet installation group attached to a mixed-flow turbofan engine model. For these analyses, we use the supersonic mixed-flow turbofan engine model with parameters described in Table 2.4 and study the inlet performance at 40,000 ft. We see that the drag contributions at subsonic Mach numbers are essentially 0 and increase as Mach increases. There is a small amount of spillage drag in the transonic region around Mach 0.8. The contributions from spillage and bypass losses have a much larger effect at Mach numbers greater than 1.5 as compared to the bleed losses. Although Mach number is the only value being directly controlled in Fig. 4.7, others vary as a result of converging the engine flow. The ratio between the engine and capture Mach numbers is shown for this inlet with a design Mach of 2.0.

4.6 Multipoint optimization of a mixed-flow turbofan

One way to greatly increase effectiveness is to improve engine performance, which lowers fuel consumption and decreases the aircraft weight for a given mission. To help make the engines more efficient across many flight conditions, researchers previously developed and evaluated VCEs to help tailor performance across the flight envelope [180–184]. In this section, we construct a mixed-flow VCE engine model, perform multipoint optimization both with and without VCE capability, and discuss the results.

4.6.1 Engine model and description

Turbofan engines are used in multiple different classes of aircraft, including commercial airliners of all sizes and many supersonic jets. High-bypass turbofan engines are used in commercial

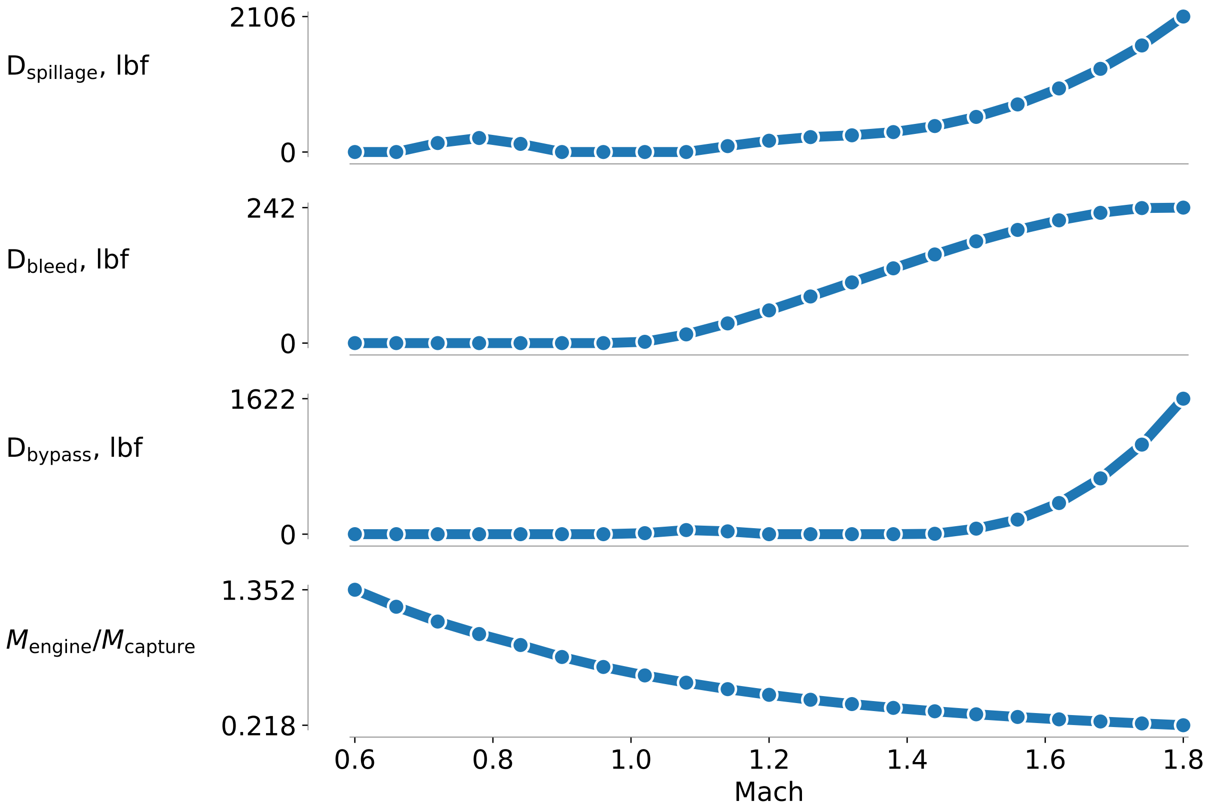


Figure 4.7: Drag forces across a Mach number sweep for the inlet installation effects using a mixed-flow turbofan engine model. The ratio between the engine and capture Mach numbers is also shown.

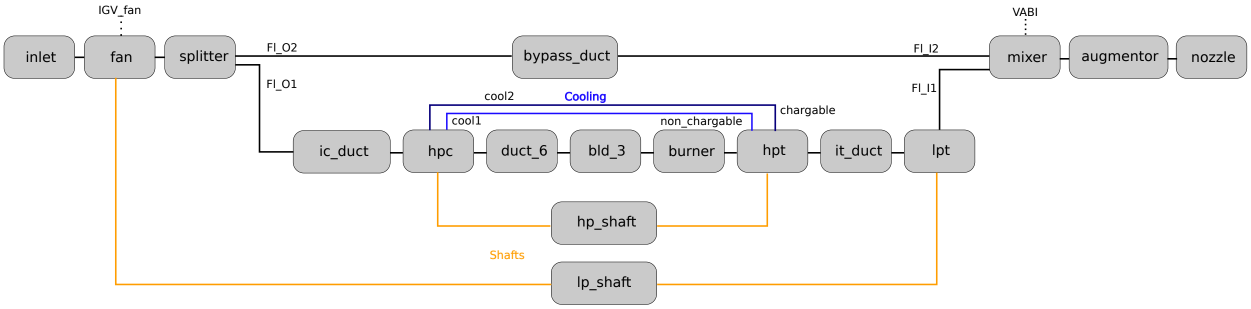


Figure 4.8: The elements in the mixed-flow turbofan engine model are shown with the VCE parameters labeled as IGV_fan and VABI.

aircraft due to their low TSFC, while low-bypass turbofan engines are used in supersonic aircraft due to their ability to produce a large amount of specific thrust. We have constructed a two-spool mixed-flow turbofan engine model within pyCycle which consists of 21 subparts, or 'elements' from NPSS terminology. This model is based on the GE F110 engine, but only loosely because we vary the design parameters throughout this study. We use the same element configuration as a GE F110 model to create a generic supersonic engine where we vary the static design. We added VCE parameters, where we can control the IGV and VABI to affect the engine performance. Figure 4.8 shows how these elements are laid out and connected.

Throughout this section we are considering the uninstalled performance of the engine. This is a first step before considering a more complete model where we include installation effects which will give more accurate performance data for the engine. Additionally, the model presented here does not include the use of an afterburner. We are designing the engine to meet all of the thrust requirements without an afterburner.

We use performance maps for the compressors and turbines to obtain relevant performance parameters for these elements. Specifically, these maps relate the pressure ratio, corrected mass flow rate, corrected engine speed, and component efficiency for the element [185, 186]. These maps are then scaled to meet requirements set on the design condition, and these scalars are passed to the off-design conditions where the scaled maps are used to obtain relevant performance parameters.

As alluded to in the previous paragraph, the setup of the problem and how we solve for the states in the engine differ in design and off-design modes. An engine is conventionally designed at a single operating point, sometimes the SLS condition, but we are interested in its performance at other flight conditions as well. We can vary the fuel flow through the engine to control the thrust, and then solve for the shaft speeds to ensure the engine performance can be calculated correctly. The mass flow rates, net power output, pressure rises, and rotational speeds of the shafts and flows for each compressor and turbine pair must match [186]. This leads to a complex solver hierarchy within this model due to the way we ensure the engine is correctly balanced. We drive the difference between the shaft powers and speeds to 0 using nested Newton systems within the model.

4.6.2 Parameter studies

We include five physical variables to control the design of the engine and we will now detail those that we use in this section. overall pressure ratio (OPR) is the total pressure ratio of both compressor stages, which is the ratio of the stagnation pressure at the front to the fan to the stagnation pressure at the back of the high-pressure compressor (HPC). fan pressure ratio (FPR) is the pressure ratio just across the fan. OPR is equivalent to the product of HPC pressure ratio and FPR. extraction ratio (ER) is the ratio of the total pressure levels in the bypass to the core exhaust ducts. Another design parameter is the desired thrust of the engine at the design point, or $\mathrm{F}_{\mathrm{net,SLS}}$. This desired thrust effectively sizes the engine by requiring a certain mass flow rate and fuel-to-air ratio (FAR), which in turn scales the performance maps, and changes performance based on the design thrust. FAR influences how much fuel is added in the burner, which is the main driver for thrust production from the engine. Although FAR is a design variable in this section, it is effectively constrained by T_4, the temperature of the flow leaving the burner. Increasing FAR means more fuel is burned, which means the flow leaving the burner is at a higher temperature. T_4 is limited by the available material technology level for the high-pressure turbine blades.

Based on these physical design variables, we can obtain an engine design that can be used at off-design points to obtain its performance. These variables are also called static design variables, because they cannot vary at different flight conditions and are fixed for a given engine design. For non-VCE engines, only these physical variables can be varied to change the geometry of the engine.

We include two operational variables that vary the geometry of the engine model so it can be controlled at individual flight conditions. For the fan, we can control the IGV, which can be thought of as varying the stator vane angle within the compressor. Varying these vane angles mostly affects the performance of the corresponding compressor element, though only for some specific flow parameters. The other operational variable in this model is the VABI, which effectively controls the bypass ratio of the engine by changing the flow areas at the mixer element. This has a large effect on the performance of the entire engine. The combination of these two variables allows the bypass ratio (BPR), stall margins, and mass flow rates to be tailored for each flight condition queried using the engine model.

We now discuss how these VCE parameters are implemented in the engine model. The IGV control is achieved by querying the compressor maps accounting for the vane angles. Specifically, we supply the corrected speed, Rline value, and IGV value to the compressor map to obtain the corrected flow, efficiency, and pressure ratio. In essence, the ability to control the IGV comes from having the correct maps that have this information tabulated. On the other hand, the VABI control does not deal with any elements maps. Instead, we directly vary the areas in the mixer element, using the design case areas as the baseline. As the VABI control is increased, the area for the core flow in the mixer is increased and the area for the bypass flow is decreased. As the VABI control is decreased, the opposite is true. The total area in the mixer remains fixed and does not change based on the VABI setting.

In addition to these two VCE variables, we can also vary FAR at each of the off-design points to hit a thrust target. Although FAR is not considered a VCE parameter, it is an operational variable

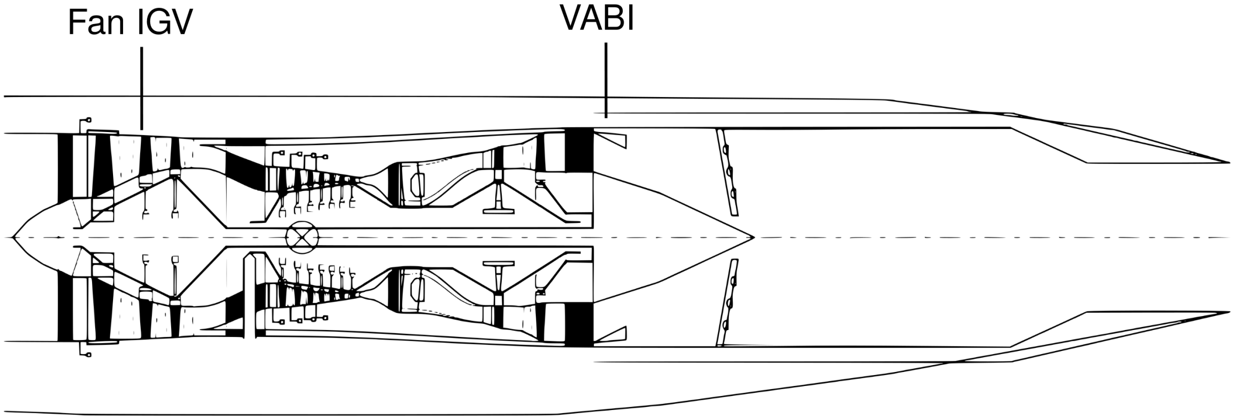


Figure 4.9: A cutout view of a representative engine with the IGV and VABI controls highlighted. Adapted from Heath et al. [170].

that can be changed to improve the performance at each flight condition.

With this model constructed, we performed sweeps on some of the operational variables to better understand the design space. This lets us see the effects of the changing the IGV and VABI controls to see how the performance can be tuned at different flight conditions. The effects of these operational variables depend on the physical design parameters and where the engine is operating in Mach-altitude space. These are complex relationships and difficult to generalize, but here we present two different parameter sweeps to examine a subset of the trends we expect to see. Both sweeps are at the same physical design and Mach-altitude flight condition. Specifically, here the engine is sized at 17,000 lbf for the SLS design case and we are performing the sweeps at Mach 1.6 and an altitude of 50000 ft. Each of the y-axes in the plots shown in this subsection are normalized by the baseline value. For IGV, this is the value when the control is set to 0.0, while for the VABI this corresponds to when the control is set to 1.0. This is done to show the relative variation instead of dimensional variation in each of these outputs to make these results easier to generalize for other flight conditions.

Figure 4.10 show the effect of sweeping through the fan IGV parameter from its lower to upper bounds. We see negligible changes in mass flow rate, but larger changes in corrected speed. This

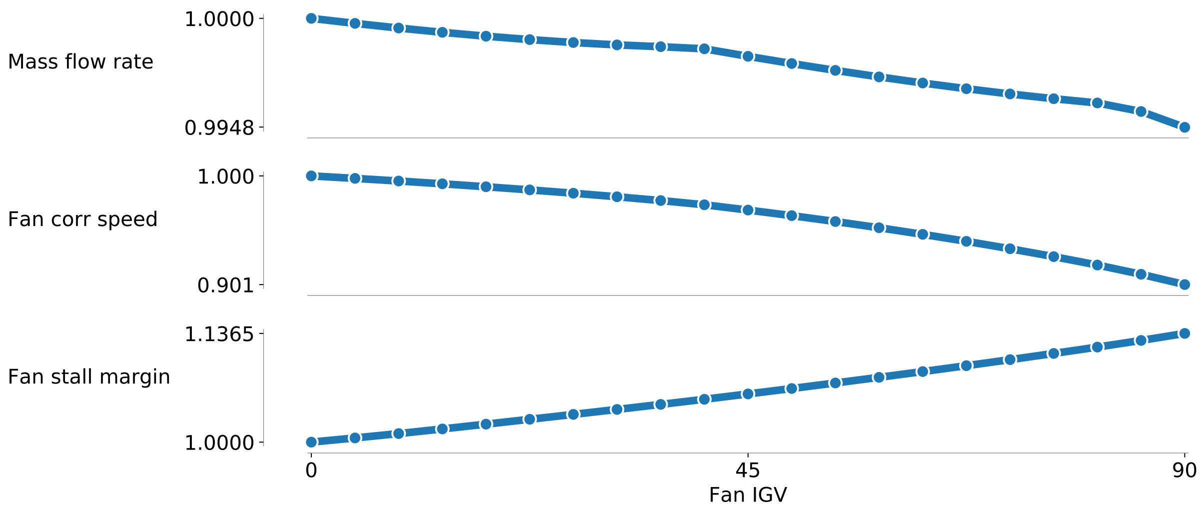


Figure 4.10: The fan IGV controls the stall margin, but we also see a change in corrected speed.

means that the IGV parameter varies where the compressor is operating on its performance map. Due to the IGV affecting where on the map the element is operating, we see a relatively change in the stall margin. If the physical engine design puts one of the operating points on the corrected speed or stall margin constraint, we could change the corresponding IGV parameter to move the operating point off that constraint and into the feasible region. To a first order, varying the IGV allows control of the stall margin and corrected speed. The small kinks in the mass flow rate curve come from the linear interpolation used when reading in the performance maps for the elements and has a negligible effect on the engine.

Unlike the IGV, the VABI has a larger affect on the overall performance of the engine. As we increase VABI, we are increasing the area in the mixer corresponding to the core flow while we lessen the area for the bypass flow. Thus, the total area remains the same as VABI changes. Looking at Fig. 4.11, as VABI increases, we see BPR, net force, and mass flow rate increase noticeably, while FAR decreases slightly.

The fact that BPR increases as we increase VABI is counterintuitive, and we now investigate the physical reasons behind this trend. As we increase the area in the core, the back-pressure decreases, which decreases the pressure drop across the turbine. This lower pressure ratio means that the high-pressure turbine puts less power on the shaft, which effectively slows down the shaft. However, this also means there was more energy left in the flow when it encounters the low-pressure turbine, which causes that turbine shaft to increase in speed, as we see in the bottom portions of Fig. 4.11. This increase in fan shaft speed increases the amount of flow that goes through the bypass duct, which increases BPR.

Whereas the IGV mostly control some aspects of the flow and engine performance near their respective elements, the VABI can be seen as an overall engine performance parameter. We expect the optimizer to take advantage of these coupled effects between operational parameters where a human designer would have trouble intuitively understanding the optimal settings for each of the flight conditions due to the complexity of interactions.

After examining how the VABI affects performance at a single flight condition, we also examined the performance across three Mach-altitude combinations. Figure 4.12 shows the fan map with points from three different VABI sweeps overlaid on the map. In general, as we increase Mach number and altitude, the fan moves to a lower corrected speed and lower pressure ratio. As we increase the VABI, we see that the points may become choked depending on the flight condition. Additionally, the efficiency of the fan decreases as we increase VABI here. This decrease in efficiency may result in better overall engine performance depending on the flight conditions and constraints imposed. We expect the optimizer to take advantage of this freedom by tailoring the VABI parameter at each off-design condition.

4.6.3 Optimization problem formulations and results

Now that we have constructed a model, briefly explored its design space, and have the ability to compute gradients through the entire system, we can formulate the optimization problems we want to solve. We first determine the flight conditions to use in this multipoint engine design problem, then we detail how we chose the design variables, constraints, and objectives for the optimization

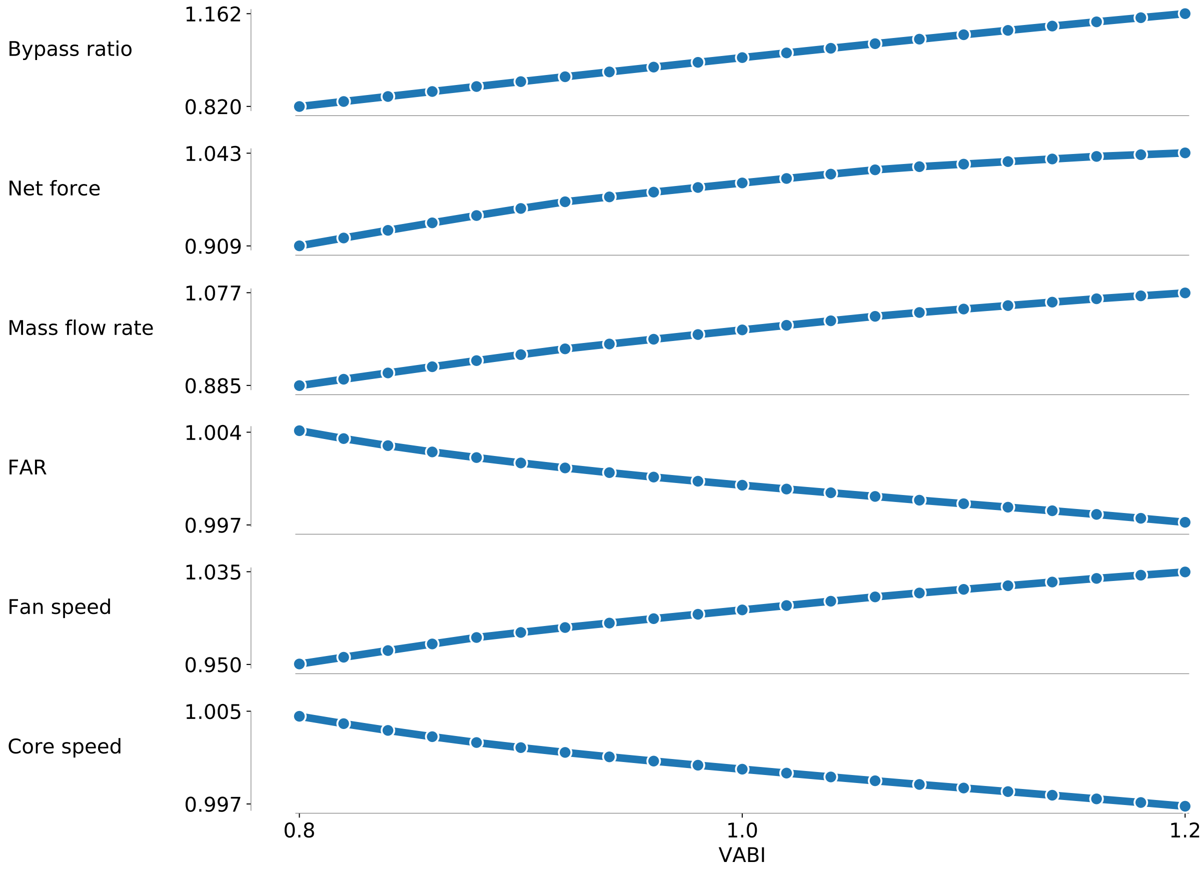


Figure 4.11: We see a greater effect on performance from the VABI compared to the IGV.

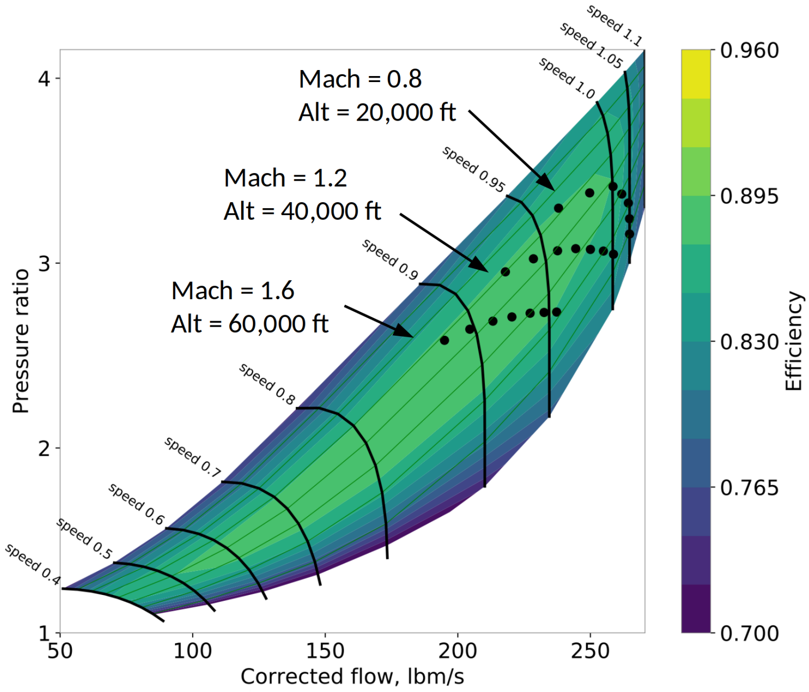


Figure 4.12: We see the fan efficiency vary greatly based on the flight condition and VABI setting. The VABI setting sweeps from 0.8 for the leftmost points to 1.2 for the rightmost.

problems before presenting results.

Given that we do not wish to analyze every point along the mission profile, we need to find the critical points in the flight envelope where the engine is constrained in some way. The physical reason for these constraints might primarily come from aircraft, engine, or mission requirements. For example, if our aircraft is at its maximum takeoff weight (MTOW) and is on a short runway, we would need to produce a large amount of thrust to accelerate the heavy aircraft quickly enough to safely takeoff. Another source for performance limits comes from the maximum temperature allowed at the exit of the burner, or T_4 as it is commonly known. This is generally limited by the material of the high-pressure turbine blades, which would begin to degrade or melt if exposed to too great of a temperature. Another possible limiting case would be if the desired mission requires a supersonic cruise at high altitude. Nominally, it is challenging to get enough air into the aircraft at high altitudes to produce enough thrust to sustain supersonic flight. On most modern fighter jets, supercruise is possible without the use of afterburners, though afterburners are still used when maximum thrust output is needed.

Although there are many possible representative points, we select rolling takeoff (RTO), top of climb (TOC), subsonic (SC), and after ordinance drop (AOD) as the four off-design conditions where we are trying to optimize engine performance. These points are shown on a supersonic ordinance drop mission in Fig. 4.13. RTO requires a large amount of thrust at a low altitude and Mach number when the aircraft is fully loaded. TOC needs to be able to produce adequate thrust for a close-to-MTOW aircraft at a moderately high altitude and supersonic Mach number. SC requires a small amount of thrust at a transonic Mach number, but we want the TSFC at this point to be low to increase the range of the aircraft. AOD needs a large amount of thrust to produce enough acceleration to quickly disengage after an ordinance drop at low altitude. The aircraft would perform optimally at each one of these flight conditions with a certain engine design, but we want to find the best single design that satisfies all of these constraints while minimizing a certain metric, such as a summation of fuel burn.

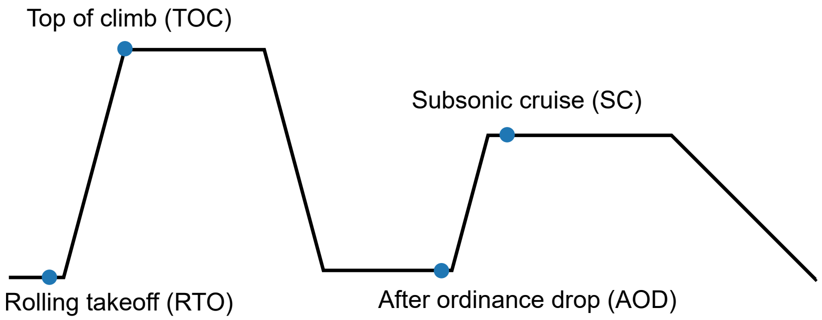


Figure 4.13: The altitude profile from a nominal mission shows the four flight conditions of interest.

Given a certain aircraft weight and aerodynamic performance, we can compute the thrust needed at each one of these flight conditions. We used aerodynamic performance data from a previous study by Jasa et al. [89] which was obtained by performing RANS CFD on the Efficient Supersonic Air Vehicle aircraft. In that previous work, we ran the model at 273 different flight conditions in the Mach-altitude-α design space to construct a 3D surrogate model. Here, we query that model to obtain the lift and drag of the aircraft at our flight conditions of interest.

We use the unsteady aircraft equations of motion to determine the thrust needed at each condition. To do this, we look at the four main force sources on the aircraft — thrust, weight, lift, and drag — and ensure that the vector sum of these forces results in the desired net force on the aircraft to achieve the prescribed motion. In the SC case, we prescribe a net force of 0 on the aircraft, while in the RTO and AOD cases, we prescribe a minimum acceleration. For TOC, we enforce a minimum climb rate related to performance requirements at the aircraft's service ceiling. We then solve these equations by varying thrust and angle of attack to satisfy these constraints and obtain the minimum thrust values needed for each condition. Explicitly, we use these equations of motion,

Table 4.2: Tabulated data for each flight condition, with the thrusts obtained from solving the nonlinear set of equations.

Flight condition name	Mach	Alt., ft	Weight, lb	Acc., ft/s^2	Climb rate, fpm	Thrust, lbf
RTO	0.2	0.1	68,000	19.9	0.	28,000
AOD	0.6	1000	50,000	24.6	0.	22,450
SC	0.8	35,000	66,000	0.	0.	2060
TOC	1.4	50,000	50,000	0.	800.	7600

$$
\begin{aligned}
\dot{v} &= \frac{T\cos(\alpha) - D}{m} - g\sin(\gamma) \\
\dot{\gamma} &= \frac{T\sin(\alpha) + L}{mv} - \frac{g\cos(\gamma)}{v} \\
\dot{h} &= v\sin(\gamma) \\
\dot{r} &= v\cos(\gamma),
\end{aligned}
$$

where v is the aircraft velocity, T is the thrust, α is the angle of attack, D is the drag, L is the lift, m is the aircraft mass, g is the acceleration due to gravity, r is the aircraft range, and γ is the flight path angle.

The resulting minimum thrust values are shown in Table 4.2. We use two engines to power the aircraft and the thrust values shown are for a single engine. We see that the RTO and AOD cases require the most thrust because we are prescribing minimum accelerations. Additionally, the TOC case requires a large amount of thrust for its altitude. We do not know *a priori* which case will be the most limiting for the engine design.

We parameterized this model to have five physical design variables that control the size and performance of the engine. These design variables are applied at the on-design condition, SLS, and the resulting design is then passed to each of the off-design cases. We allow the optimizer to vary the net thrust, the mixer ER, FPR, OPR, and FAR in the core. We selected these design variables to produce a well-posed optimization problem that gives enough freedom to the optimizer to control the overall engine performance. In addition to these physical variables, we also give the optimizer freedom to vary the operational variables at each flight condition. This includes the fan

IGV, VABI, and FAR at each point.

We add constraints to the optimization problem to produce a realistic design. We constrain the area of the inlet to limit the overall size of the engine. Because we are not designing the entire aircraft, this area constraint is in place to limit the size of the engine so it could fit within an existing airframe. A more complete optimization problem would at least consider the supersonic inlet, if not size the aircraft at the same time, but that is outside the scope of the current work. We also constrain the flow exit temperature of the burner at all flight conditions by an upper bound due to the material of the high-pressure turbine blades. At each flight condition we constrain the corrected speed of the fan to not exceed the maximum speed allowed by the engine assembly. We also add minimum thrust targets for each of the off-design points.

We could select an objective function from a few quantities of interest, including fuel burn, MTOW, maximum speed, maximum altitude, etc. Given that we are not designing the entire aircraft in this section, we chose to minimize an equally-weighted sum of TSFC values at the four off-design points to obtain a representative measure of the performance of the engine. This is not necessarily a realistic design objective, but we use it here to illustrate how the model and optimization method consider performance across multiple flight conditions simultaneously. A more advanced optimization problem might consider varying the aerodynamic properties, the internal structure, or the mission profile of the aircraft simultaneously as we design the engine.

Table 5.1 shows the complete optimization problem formulation for two cases. In the first case, we give the optimizer freedom to vary all operational variables, whereas in the second case, we remove the VCE parameters. This allows us to see the effect that the VCE parameters have on the optimal performance of the engine. Although FAR is considered an operational variable, we include it in both problem formulations because non-VCE cycles can vary FAR at each off-design point.

The first five variables in Table 4.3 correspond to the physical design variables for the engine, specifically at the SLS case we selected as our design point. The rest of the variables are the

Table 4.3: This shows the multipoint optimization problem formulations for both the full multipoint problem considering all operational variables as well as the multipoint problem where we do not vary the VCE parameters.

Category	Name	Quantity – VCE	Quantity – No VCE	Lower	Upper	Units
Objective	$\sum$ TSFC	1	1	–	–	$\frac{\text{lbm}}{\text{hr lbf}}$
Variables	FPR	1	1	2.5	4.0	–
	ER_{mixer}	1	1	0.9	1.1	–
	OPR	1	1	20.	35.	–
	$\text{F}_{\text{net,SLS}}$	1	1	15,000	35,000	lbf
	FAR_{SLS}	1	1	0.015	0.028	–
	FAR	4	4	0.0135	0.028	–
	IGV_{fan}	4	0	0.	90.	–
	VABI	4	0	0.95	1.05	–
	Total	**18**	**10**			
Constraints	Inlet area	1	1	1000.	2000.	inches^2
	T_4	5	5	2000.	3200.	°R
	Fan corr. speed	5	5	0.4	1.05	–
	$F_{\text{net,RTO}}$	1	1	28,000	–	lbf
	$F_{\text{net,TOC}}$	1	1	7600	–	lbf
	$F_{\text{net,SC}}$	1	1	2060	–	lbf
	$F_{\text{net,AOD}}$	1	1	22,450	–	lbf
	Total	**15**	**15**			

operational variables for each off-design point, including the two VCE parameters as well as FAR. We then have the inlet area, burner exit temperature, and corrected speed constraints, which help keep the engine within a realistic physical limits. the minimum thrust targets for the off-design points fill out the bottom of the table. This gives us a total of 18 design variables and 15 constraints for the full problem.

We performed six different individual optimizations using the VCE cycle, each one with a different maximum allowed inlet area, and repeated these six optimizations with the non-VCE cycle. By varying the area constraint and optimizing the physical and operational variables, we can directly see the effect that this constraint has on the optimal design and performance of the engine. Recall that all analysis done here is considering the uninstalled performance of the engine and we do include an afterburner. Because we do not have a measure of the effects of the inlet

Table 4.4: Optimized design variables and performance obtained from each of the six optimization problems for the VCE cycle.

Inlet area, in^2	$F_{net,SLS}$, lbf	ER_{mixer}	FPR	$TSFC_{SC}$, $\frac{lbm}{hr\ lbf}$	$TSFC_{TOC}$ $\frac{lbm}{hr\ lbf}$	$T_{4,RTO}$, °R
2000	33,435	0.942	2.855	0.703	0.882	3070.7
1900	32,873	0.945	2.969	0.712	0.895	3002.5
1800	32,301	0.947	3.103	0.722	0.907	3013.5
1700	31,685	0.954	3.239	0.733	0.921	3024.4
1600	30,818	0.956	3.369	0.745	0.939	3029.9
1500	30,650	0.963	3.630	0.761	0.955	3052.5

geometry, we use the inlet area constraint as a stand-in.

Results from these optimizations are shown tabularly in Table 4.4 and graphically in Fig. 4.14. For each case shown, the optimal values for OPR and $TSFC_{TOC}$ converged to 35 and 3200 °R respectively, so they are not shown in the table. As we make the inlet area constraint more restrictive, we see that the optimizer decreases the design case net thrust. However, since we are still enforcing the minimum thrust constraints on the off-design points, the optimizer increases FPR and decreases BPR to produce more thrust from the core to make up the decreased ability to produce thrust using the bypass stream. As one would expect, as we limit the design space through more restrictive inlet area constraints, the TSFC values increase. In each of the optimization cases, T_4 is at its upper limit only for the TOC flight condition, whereas the other flight conditions are at partial throttle or are below the T_4 limit.

At each optimal point, the VCE-capable cycle outperforms the non-VCE cycle. This is as expected, since we are giving the optimizer more freedom to vary the engine design at each flight condition when using VCE parameters. The non-VCE optimization corresponding to an inlet area of 1500 in^2 did not converge because the optimizer could not find a feasible point, so we do not show results for that case.

The operational parameters are shown in Table 4.5. Although they varied during the optimization process, the IGV and VABI settings are all on a bound at the optimal design. This is partially due to our objective function, which seeks to minimize TSFC across all points, which encourages

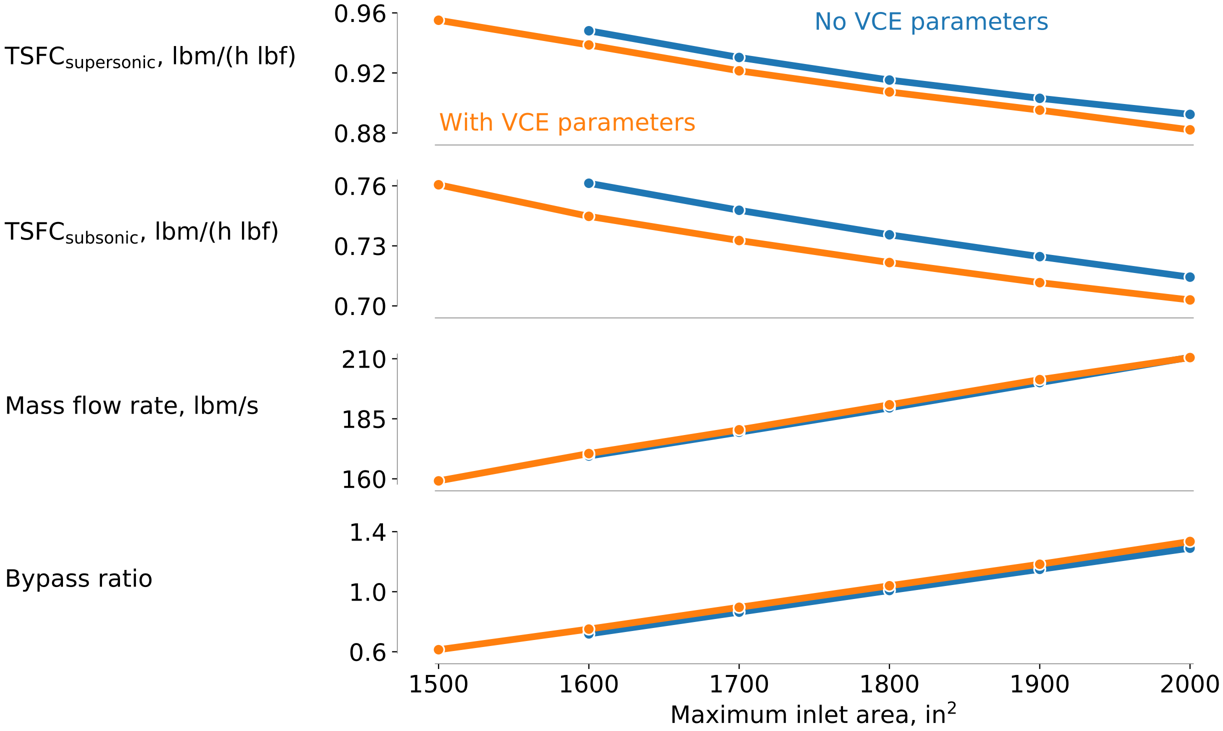


Figure 4.14: As we decrease the allowable inlet area, TSFC increases and mass flow rate and bypass ratio decrease.

Table 4.5: Operational variables at the optimal points for each of the six optimization problems, looking at only the minimum and maximum values across the four flight conditions.

	IGV_{Fan}		VABI		FAR	
Inlet area, in^2	Min	Max	Min	Max	Min	Max
2000	0.000	0.031	1.049	1.050	0.0178	0.0264
1900	0.000	0.023	1.050	1.050	0.0173	0.0262
1800	0.000	0.028	1.050	1.050	0.0168	0.0262
1700	0.000	0.083	1.050	1.050	0.0161	0.0262
1600	0.000	0.026	1.050	1.050	0.0159	0.0260
1500	0.000	0.072	1.050	1.050	0.0158	0.0262

the optimizer to maximize the VABI setting because we are not considering installed effects. Because we did not include compressor stall margin constraints, the IGV are on their lower bounds at the optimal points, which minimizes TSFC. We would expect these optimal operational parameters to move off the bounds if we had a more realistic problem formulation with additional constraints.

We also performed four single-point optimizations to compare against the multipoint optimized result. All optimizations for this comparison used an inlet area constraint of 1800 in^2. In each single-point optimization, we ran only the on-design and one off-design flight condition without considering the performance of any other flight condition. We gave the optimizer control over all of the physical and operational variables corresponding to that single flight condition of interest and retained the same constraints on that flight condition as in the multipoint problem. The objective function for each of these single-point optimizations was the TSFC of that individual point.

Figure 4.15 shows the TSFC values obtained from the VCE-capable multipoint optimization as well as four individual single-point optimization. We see that the single-point TSFC values outperform the multipoint results across all four flight conditions, which allows us to quantify the sacrifice in fuel consumption efficiency necessary to meet performance demands at multiple flight conditions. However, in reality, we desire good performance from a variety of flight conditions. The multipoint optimum is better overall because the single-point optimums have poor performance for flight conditions different than their design condition.

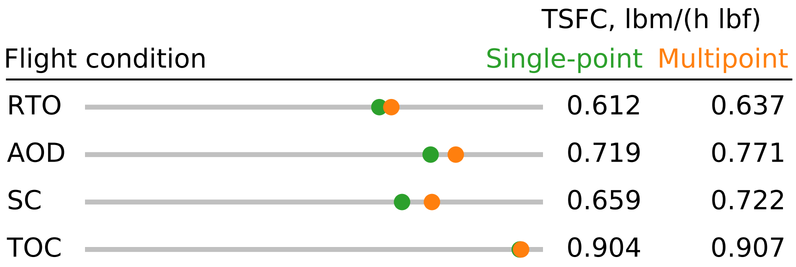


Figure 4.15: The single-point optimizations result in TSFC values that are 0.3–8.6% lower than those from the multipoint optimization, showing the decreased fuel efficiency at individual flight conditions when we optimize considering multiple flight conditions.

CHAPTER 5

Aero-thermal-propulsive-mission Optimization

With multiple disciplinary models constructed and examined in optimization problems in previous chapters, we combine these models into fully-coupled multidisciplinary problems and present optimization results in this chapter. There is no shortage of combinations and studies that can performed by considering subsets of the aero-thermal-propulsive-mission problem. We focus first on an engine-thermal optimization where we consider the performance of an engine with a bypass duct heat exchanger. We then optimize the minimum time-to-climb mission considering thermal constraints from the FTMS. Lastly, we examine the optimal flight profile of a supersonic reconnaissance mission considering thermal constraints using the dual-tank FTMS.

5.1 Engine-thermal optimization

The next generation of aircraft, in both the commercial and military sectors, must meet increasingly stringent requirements, including constraints on fuel consumption, noise, emissions, heat, and cost. To meet these performance metrics, aircraft designers must innovate, from the system level to individual parts. The propulsion system has a particularly large effect on overall aircraft performance, since it propels the aircraft, provides auxiliary power, and allows an avenue for heat dissipation.

To capture the effects of bypass heat exchangers, we need to model both the engine and thermal systems simultaneously and in a fully coupled manner. This has been done previously by Allison et al. [75] and by Puterbaugh et al. [154], who found interesting trade-offs between engine performance and thermal dissipation capacity. However, few of these studies looked at changing the design of the engine or thermal system, and no previous study has performed optimization on the coupled system. To explore a large design space efficiently, we need to use gradient-based design optimization [187]. In the case of coupled engine-thermal systems, we need the derivatives of any quantity of interest (fuel burn, thrust produced, heat dissipated) with respect to all engine and thermal design variables.

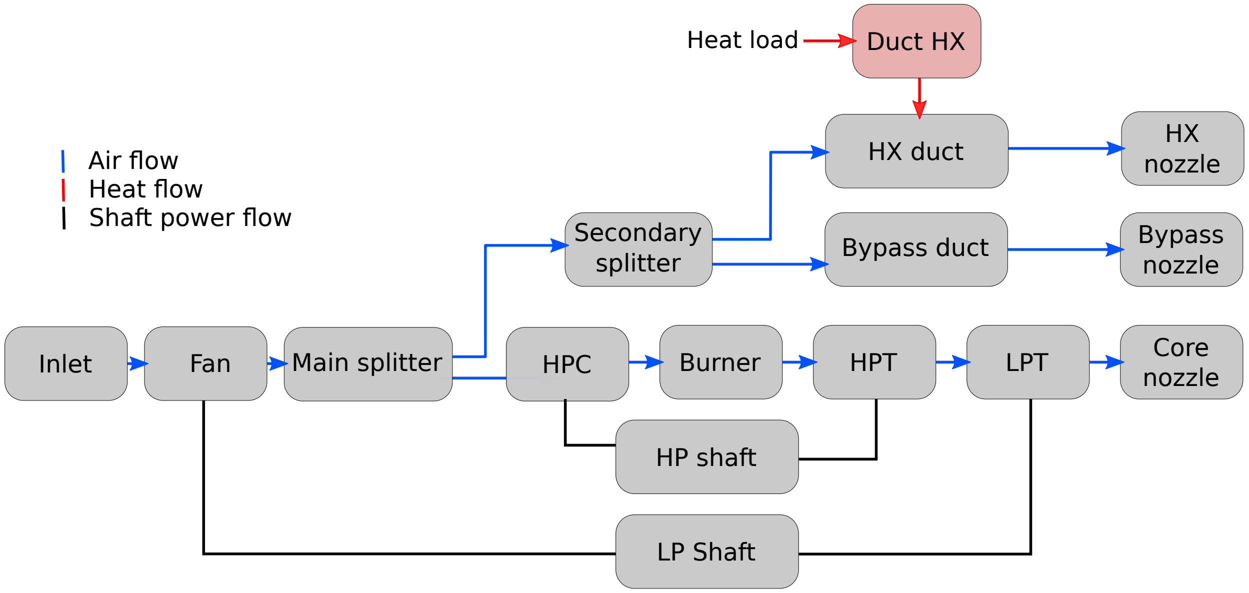
In this section, we first discuss the relevant theory behind engine and thermal system modeling. We then construct a model that couples the engine and thermal system and efficiently computes the relevant gradients. We construct a 3-stream, split-flow supersonic turbofan engine and examine different thermal system architectures, including a simple heat-lifting system. We then perform a series of optimizations to show how the engine and thermal systems interact together and find optimal designs for a variety of thermal loads and flight conditions. We find that the addition of the thermal lifting system decreases the fuel consumption for a thermally-constrained supersonic flight condition by amounts ranging from 1.5% for 2.5 kW of heat rejection, to 2.6% for 15 kW of heat rejection.

5.1.1 Model description

5.1.1.1 Propulsion Model

We use pyCycle to construct our three-stream split-flow supersonic engine model, which is shown in Fig. 5.1. We also show the engine model with the addition of the thermal lifting system in Fig. 5.2, which is explained in detail in Sec. 3.3.2. Throughout the paper, we perform single-point design optimization, though pyCycle has the ability to run multipoint optimization problems in parallel. We use a supersonic top-of-climb flight condition throughout this work, since a previous

Figure 5.1: The 3-stream engine schematic with bypass duct heat exchanger



multipoint design study found it was the most limiting condition [85].

We include installation effects using a series of surrogate models based on data from a fixed-shape supersonic inlet as described in Sec. 4.5. These drag contributions are subtracted from the net thrust of the rest of the engine cycle to obtain an installed net thrust, which is used to measure engine performance. Split-flow turbofans are realistically not used in the supersonic regime, though we use a split-flow architecture here to study engine-thermal interactions with a relatively simple engine model. For that reason, the engine architecture used here does not have an afterburner.

5.1.1.2 Fully Coupled Model

By directly coupling the propulsion and thermal system models, we can evaluate the performance of the full system while correctly accounting for the systems' interactions. An XDSM diagram [68] showing the data passing between the models and optimizer is shown in Fig. 5.3. The optimizer provides design variable values to the engine and thermal systems. pyCycle computes the bypass duct airflow properties, including air temperature, heat capacity, density, thermal con-

Figure 5.2: The 3-stream engine architecture with the thermal lifting system added

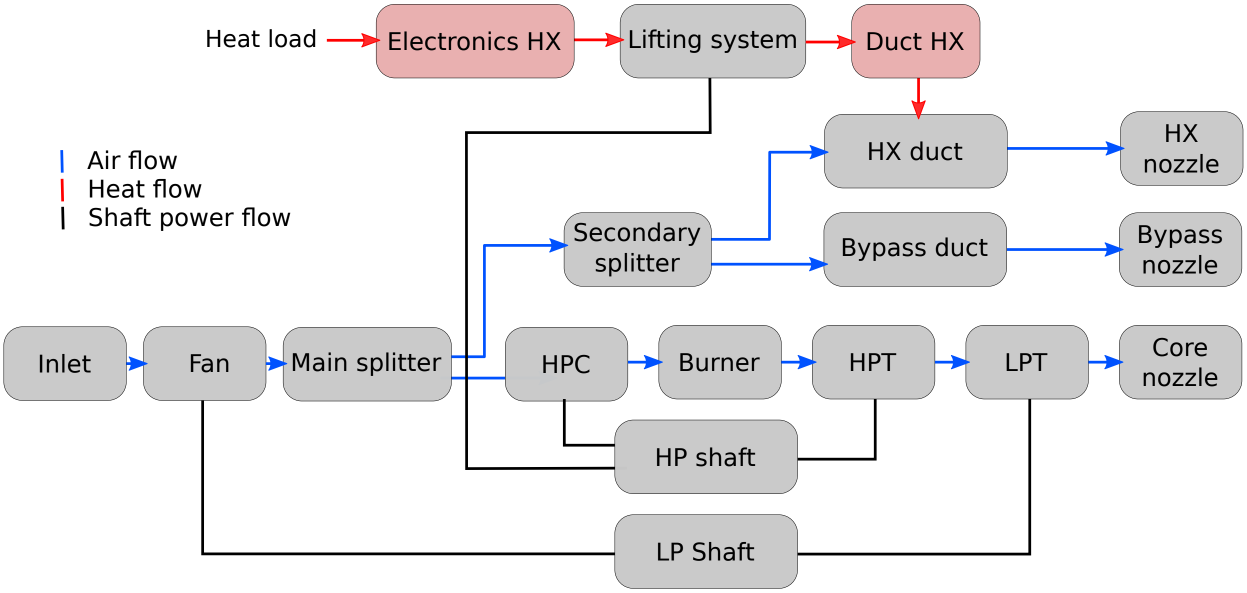
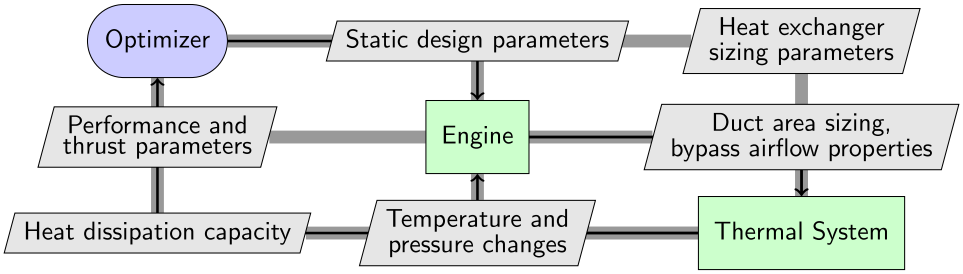


Figure 5.3: The XDSM diagram for the engine-thermal system



ductivity. These properties are then passed to the heat exchanger model, which then computes the heat transfer and pressure drop across the heat exchanger to pass back to the engine model. These coupled systems are converged simultaneously using a Newton solver with an Armijo–Goldstein linesearch. Because we constructed the models in OpenMDAO with MDO in mind, we can modularly change the system grouping and solver settings based on knowledge of how best to converge the complex system.

Throughout this work, we are interested in examining the trends and coupling between the engine and thermal systems instead of the absolute numbers from the analyses and optimizations.

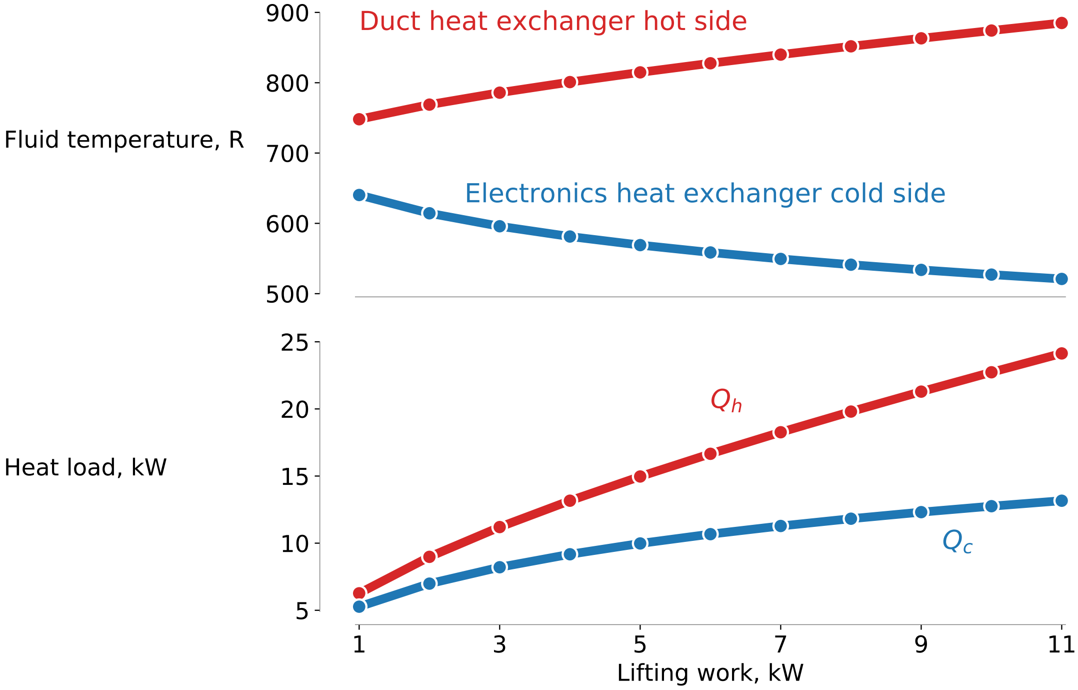
We assume a constant temperature and single-source heat load, when in reality this thermal system would have multiple heat loads at different qualities. The interactions between the disciplines are captured here without using exact efficiencies, element maps, and sizing parameters from engine manufacturers. This allows us to more publicly examine the coupled system and report on our findings.

5.1.2 Parameter studies

We can use the fully coupled model to explore multidisciplinary trade-offs of interest to both engine and thermal system designers. We can vary engine design parameters, such as bypass ratios, design Mach numbers, pressure ratios, element maps, and efficiencies. On the thermal system side, we can control the heat exchanger geometric variables, such as fin length, width, spacing, and offset distance. For the thermal lifting system, we can vary the mass flow through the ACM and the heat exchanger variables that interface with the ACM. In this work, we examine a small subset of these design parameters while holding the others constant to highlight how these variations affect overall system performance.

To gain an intuitive understanding for how the thermal lifting process affects optimal engine performance, we analyzed the coupled model across a sweep of lifting work values. We used the same fixed engine design used in the BPR study and did not perform any design optimization here. Figure 5.4 shows the temperatures and heat loads for the hot and cold sides of the lifting system for 11 different nominal lifting work values. As the amount of lifting work done increases, the temperature difference in the ACM increases, which means the heat exchangers can dissipate more thermal energy due to the larger temperature difference between the hot and cold sides. However, the work used to lift the thermal energy must also be dissipated by the heat exchangers, which means more thermal energy is put into the bypass airstream. The optimal amount of lifting work, accounting for the increased temperature difference and larger thermal loads, depends on the bypass airflow properties and heat exchanger designs. Depending on the vehicle flight conditions,

Figure 5.4: As more lifting work is added to the thermal lifting system, the amount of heat that must be dissipated increases, as does the temperature of the air within the ACM. Q_h is greater than Q_c due to the added heat from the lifting work.



different lifting work amounts will be optimal based on the amount of heat dissipation required.

5.1.3 Optimization results

As discussed previously, there are a large number of potential optimization problems that could be solved to study the design space of this model. We first compare optimal engine performance for the non-lifting and lifting thermal systems across a range of heat transfer requirements at a single flight condition. For all of these optimizations, we hold the heat exchanger designs fixed, except for the frontal area of the heat exchanger in the bypass duct, which we vary to occupy the entire duct. Table 5.1 shows the optimization problem formulations for both thermal system types. The lifting work design variable is only present in the engine-thermal model that includes the lifting system.

Table 5.1: The optimization problems for both coupled system types. The lifting work variable is only in the lifting system problems.

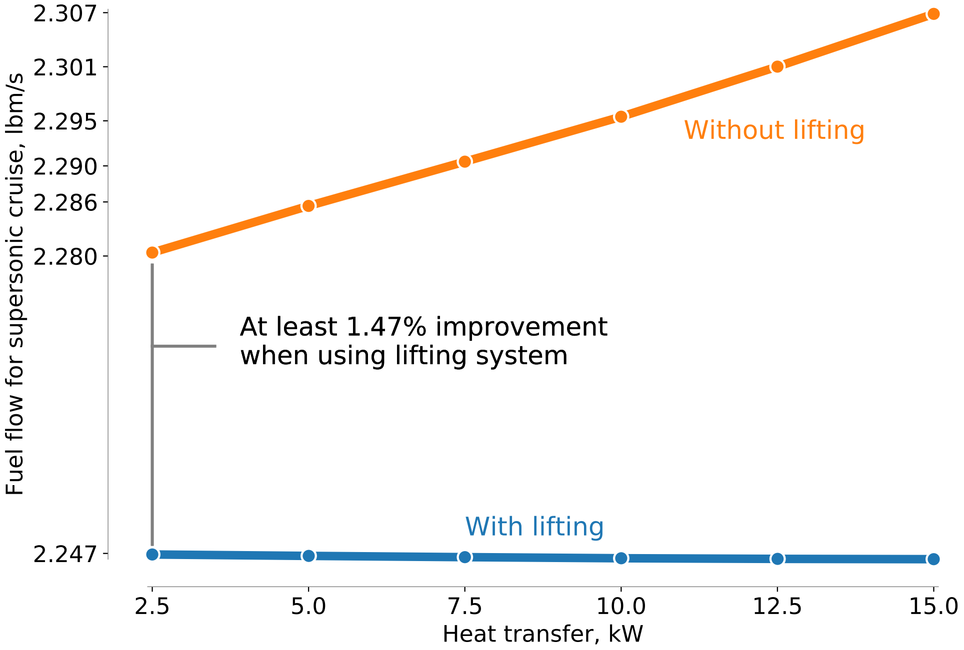
Category	Name	Lower	Upper	Units
Objective	Fuel flow	–	–	lbm/s
Variables	FPR	1.5	5.0	–
	OPR	15.	35.	–
	$F_{\text{net,SLS}}$	22,000	35,000	lbf
	Ac_{des}	600	1200	in^2
	Main BPR	0.05	–	–
	Lifting work	0.5	–	kW
Constraints	Capture area ratio	1.	1.	–
	$F_{\text{net,TOC}}$	7600	–	lbf
	Heat transfer	2.5	15	kW

We are minimizing the fuel flow to the combustor while meeting a heat transfer requirement at a supersonic flight condition. The optimizer has control over FPR, OPR, the main BPR, and the design sea level static thrust for all cases, as well as the lifting work for the lifting system cases. The Ac_{des} variable controls the capture area of the inlet, which we constrain to be equal to the capture area of the supersonic case to accurately compute installation drag. We sweep through heat transfer requirements from 2.5 kW to 15 kW for both thermal system architectures and perform optimizations at each point, prescribing the heat transfer amount using an equality constraint. For these optimizations, we hold the secondary BPR fixed to 1.0 to simplify the optimization problem.

Figure 5.5 shows the optimized results for the two types of coupled systems. We see that by using a lifting system, the optimal fuel flow at the supersonic flight condition is improved between 1.5% to 2.6%. The only ways for the non-lifting system to meet the heat transfer requirement are by changing the bypass airflow or duct size, whereas the lifting system can also change the quality and quantity of the thermal load.

Tables 5.2 and 5.3 show the optimal design variable values. Examining the optimal designs, as the heat transfer requirement increases for the non-lifting system, the optimizer chooses to decrease FPR, which lowers the temperature of the air in the bypass duct, allowing for more heat transfer with the heat exchanger. On the other hand, the lifting system can have a higher FPR

Figure 5.5: The lifting system allows the optimizer to find a lower fuel burn across all heat transfer requirements.



while still meeting the designated heat transfer, leading to a more efficient engine. The addition of the lifting system allows the optimizer to find the thermodynamically optimal engine design for each heat transfer requirement. This is partially possible because we are only considering engine performance at a single flight condition.

The lifting system introduces additional thermal energy into the bypass duct, which causes the bypass nozzle to create more thrust than if the airflow was not heated. This slightly positive net

Table 5.2: Optimal design variable values for the non-lifting system.

Heat transfer, kW	Design thrust, lbf	OPR	FPR	Main BPR	Fuel flow, lbm/s
2.5	31584	35.0	3.050	0.598	2.280
5.0	31548	35.0	2.787	0.589	2.286
7.5	31504	35.0	2.540	0.581	2.290
10.0	31464	35.0	2.309	0.571	2.295
12.5	31448	35.0	2.095	0.555	2.301
15.0	31436	35.0	1.895	0.533	2.307

Table 5.3: Optimal design variable values for the lifting system.

Heat transfer, kW	Design thrust, lbf	OPR	FPR	Main BPR	Fuel flow, lbm/s	Lifting work, kW
2.5	31651	35.0	5.000	0.643	2.247	0.62
5.0	31651	35.0	5.000	0.643	2.247	1.94
7.5	31651	35.0	5.000	0.643	2.247	4.24
10.0	31653	35.0	5.000	0.644	2.246	8.00
12.5	31660	35.0	5.000	0.644	2.246	14.18
15.0	31669	35.0	5.000	0.644	2.246	24.67

thrust effect, coupled with the larger temperature difference possible through the lifting system, counteracts some of the negative effects associated with needing to dissipate more thermal energy. The optimal lifting work values go from 0.62 kW to 24.67 kW across this range of heat transfer requirements. Since we require more heat to be dissipated, much more lifting work is required at each optimal point.

The coupled model does not include many of the realistic considerations that are necessary when measuring total system performance. In particular, we do not account for the weight of the heat exchangers or thermal system components, which is not negligible. Additionally, the actual performance of the ACM differ from what is modeled here, because we are using an analytic Brayton cycle with assumed efficiencies. We are considering steady-state thermal effects, though heat flow is an inherently transient problem. We would have to integrate this engine-thermal system across multiple time points to accurately evaluate the transient performance.

5.1.4 Conclusions

In this section, we constructed a coupled engine-thermal system that models a 3-stream split-flow supersonic engine with a heat exchanger in the secondary bypass duct and an optional heat lifting system. This model was constructed in a modular way to more easily integrate it into a larger vehicle- or mission-level optimization problem.

We first performed analysis sweeps to examine how the model behaves and to ensure that the trends matched physical reasoning. We then performed a series of optimizations across a range of heat transfer requirements. We compared the performance of the engine with and without the thermal lifting system and found that thermal lifting enabled fuel burn reductions of 1.5% for 2.5 kW of heat rejection, to 2.6% for 15 kW of heat rejection. The results presented here are limited in scope, and this model could be used for a variety of other optimization studies.

5.2 Simplified aero-thermal-propulsive-mission optimization of a supersonic aircraft

As discussed in previous sections, the impact of thermal requirements is already apparent in the current generation of fighter aircraft. In this section, we construct an aero-thermal-mission coupled model and solve a series of MDO problems to find the optimal minimum time-to-climb trajectory subject to thermal constraints under various heat loadings. We first explain the details of the coupled system, then present seven different optimization cases and compare their results.

5.2.1 Optimization problem formulation

The aircraft studied here is the ESAV vehicle described in Sec. 1.5.2. The fully-coupled system used here consists of the simplified FTMS described in Sec. 3.2.1, a surrogate-based propulsion model of the GE F110 engine, and the aerodynamic surrogate introduced in Sec. 1.5.2.

For each result in this section, we use the same basic optimization problem formulation with modifications for each case. Table 5.4 shows a nominal optimization problem, where we find the minimum time for the aircraft to climb to 20,000 meters, reach Mach 1, and a horizontal flight path angle. The aero-thermal-mission problem is set up so we can easily change the discretization, choose to optimize the recirculation or not, vary the heat inputs, and many other combinations of settings.

Table 5.4: Nominal optimization problem formulation

Category	Name	Quantity		Lower	Upper	Units
		No recirculation	Recirculation			
Objective	time	1	1	–	–	seconds
Variables	α	50	50	-8	8	degrees
	throttle	50	50	0	1	–
	$\dot{m}_{\text{recirculated}}$	0	50	0	–	kg/s
	range	25	25	0	1000	km
	altitude	25	25	0	20	km
	velocity	25	25	10	–	m/s
	γ	25	25	-86	86	degrees
	mass	25	25	15,000	80,000	kg
	temperature	25	25	–	–	K
	time	1	1	0	–	seconds
	Total	**251**	**301**			
Constraints	final altitude	1	1	20,000	20,000	m
	final Mach	1	1	1.0	1.0	–
	final γ	1	1	0	0	degrees
	Mach	75	75	0.1	1.8	–
	temperature	75	75	–	312	K
	$\ddot{m}_{\text{recirculated}}$	0	24	0	–	1/s^2
	$\dot{m}_{\text{flow}}$	0	24	0	50	kg/s
	range defects	25	25	0	0	km
	altitude defects	25	25	0	0	km
	velocity defects	25	25	0	0	m/s
	γ defects	25	25	0	0	degrees
	mass defects	25	25	0	0	kg
	temperature defects	25	25	0	0	K
	$\dot{\alpha}$ defects	24	24	0	0	degrees/s
	throttle rate defects	24	24	0	0	1/s
	Total	**348**	**396**			

The thermal loads used in each case are based on estimates of next generation fighter jets' possible avionics, armament, and waste heat generating systems. For all cases in this section, we use 25 mission segments of transcription order 3 in an LGL scheme. We use an operating empty weight of 10,500 kg and start the mission phase after takeoff. The ambient and starting temperature for the fuel is 310 K and we set a limit of 312 K to represent the maximum temperature rise allowable for this portion of the entire mission.

Table 5.5: Heat loads for results shown in Fig. 5.6

	Color	Recirculation	$\dot{Q}_{\text{env}}$, kW	$\dot{Q}_{\text{sink}}$, kW	q_{pump}, kW/kg	q_{out}, kW/kg
Case 1	Blue	No	400	100	0	0
Case 2	Orange	No	450	100	0	0
Case 3	Green	No	500	100	0	0

5.2.2 Optimal trajectories under different heat loads

For the set of optimization cases, we do not allow the optimizer to recirculate any fuel while subjecting the system to a specified heat load through $\dot{Q}_{\text{env}}$ and $\dot{Q}_{\text{sink}}$. This means it must increase the amount of fuel in the tank to increase the thermal capacitance of the system to stay below the temperature limit. Table 5.5 shows the three separate case descriptions for the $\dot{Q}_{\text{env}}$ studies.

Figure 5.6 shows the results from three separate optimization cases, where each has the parameters as described above, but they have different $\dot{Q}_{\text{env}}$ values. The heavier aircraft cannot accelerate as quickly, which means that in the optimal trajectory the aircraft spends more time building up speed near a fixed altitude before climbing to the desired altitude. Case 3 takes 135 seconds to climb while Case 1 only takes 113 seconds. Although this is a fairly straightforward result, it confirms that the fully coupled system and optimization problem behave as expected.

5.2.3 Effect of allowing recirculation

We now compare the worst case scenario from Fig. 5.6 with a case where we allow the optimizer to control the amount of fuel being recirculated throughout the system. The two case specifications are shown in Table 5.6. Additionally, we set q_{out} to be 5000 kJ/kg, which means that we assume there is a heat exchanger that removes an amount of heat from the fuel dependent on the recirculated mass flow. We limit the total mass flow of the system so that the sum of the mass fuel burn rate and the recirculation rate must be less than or equal to 50 kg/s.

With the ability for the aircraft to recirculate fuel, the aircraft can climb at a faster rate due to its decreased mass as it needs to carry less overall fuel to cool the system. The amount of fuel

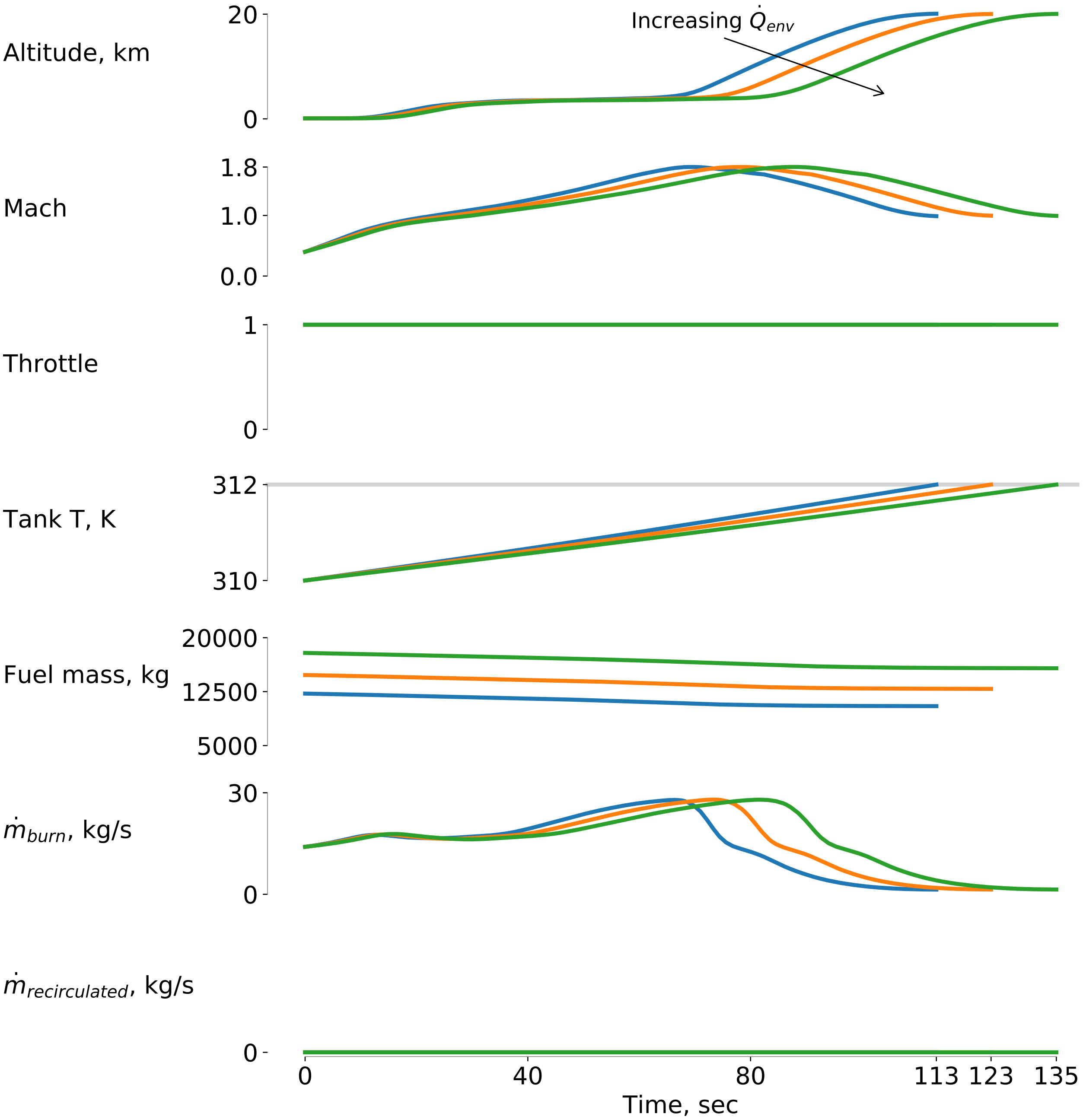


Figure 5.6: As we increase the thermal load on the fuel tank, we see the trajectories get less aggressive due to the increased aircraft mass.

Table 5.6: Heat loads for results shown in Fig. 5.7

	Color	Recirculation	$\dot{Q}_{\text{env}}$, kW	$\dot{Q}_{\text{sink}}$, kW	q_{pump}, kW/kg	q_{out}, kW/kg
Case 3	Green	No	500	100	0	0
Case 4	Red	Yes	500	100	0	5

being recirculated is such that the total fuel flow is always at the limit of 50 kg/s.

5.2.4 Effect of pump-rate-dependent heating

The last set of cases we examine here introduces a thermal term that is dependent on the mass flow rate through the main fuel pump. This heat source is a simplified representation of a variety of heat sources that increase as fuel flow or thrust increase, including waste heat from gearbox bearings and shafts, oil pumps, and geared lift fans. For these cases, we do not have any other sources of heat except the flow-dependent term, and we do not consider any fuel recirculation effects.

Essentially, the optimizer could choose to load more fuel onto the aircraft to increase thermal capacitance, or fly at lower throttle (lower fuel flow rate) to lessen the heat loads. Figure 5.8 shows the results of the three optimization cases described in Table 5.7.

We see a drastic difference in the optimal trajectories for each of these cases because the amount of heat being added to the system is dependent on the fuel flow rate, which in turn depends on throttle, altitude, and Mach. When the thermal load is at the worst case scenario, shown in pink, we see the optimizer choose to greatly increase the amount of fuel on-board, but also decrease the throttle setting during the initial portion of the mission. This combination leads to a much longer minimum time-to-climb compared to the two other cases. Each case is thermally constrained here, but the case with the lowest q_{pump} coefficient requires much less fuel to complete the mission. It starts with 7589 kg of fuel compared to the worst case scenario's 16,276 kg. This shows the large effect on sizing that thermal constraints can have, even if we are able to optimize the trajectory simultaneously.

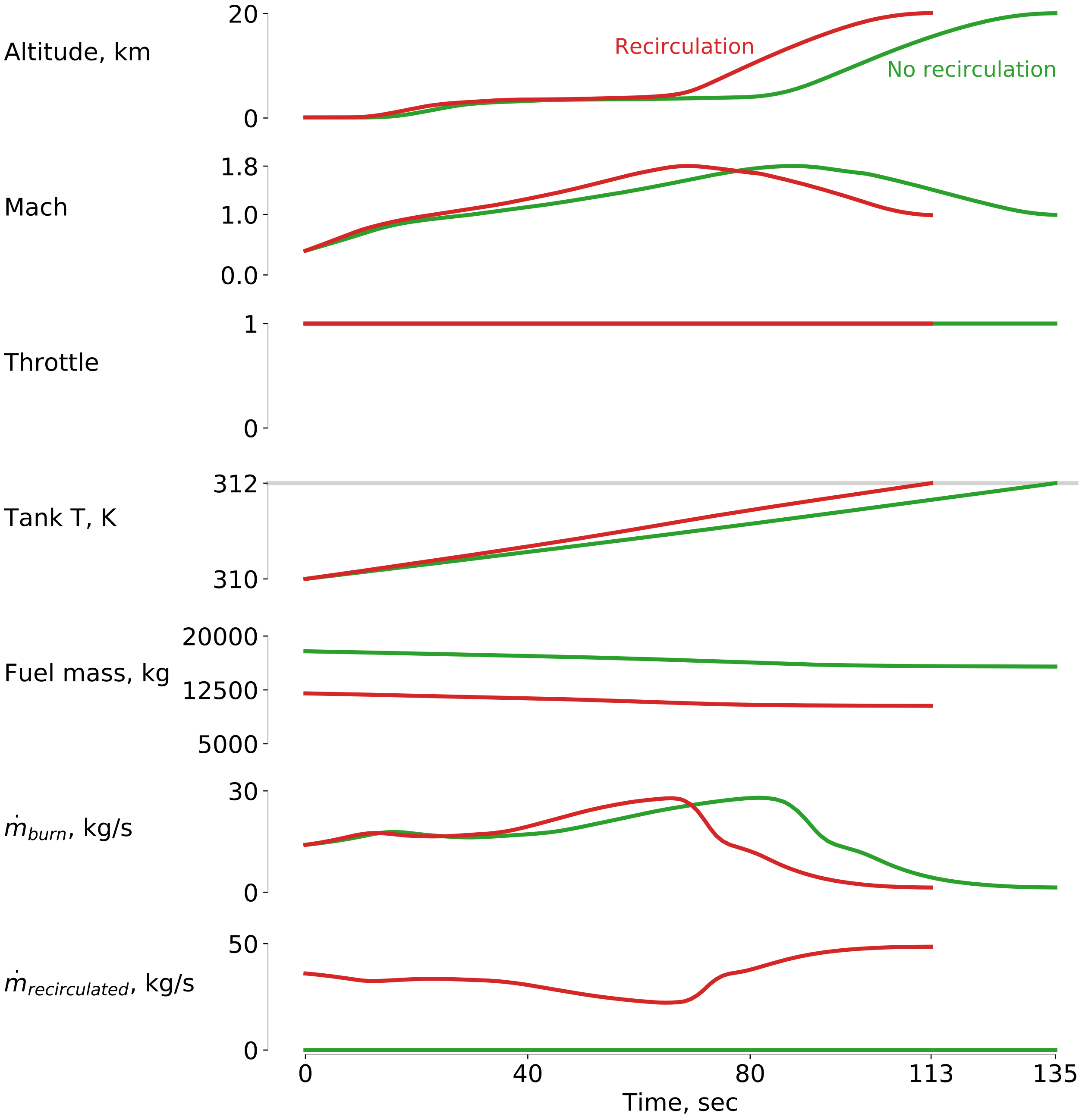


Figure 5.7: Allowing the FTMS to recirculate fuel to cool down the system yields a minimum time-to-climb that is 22 seconds less than without fuel recirculation.

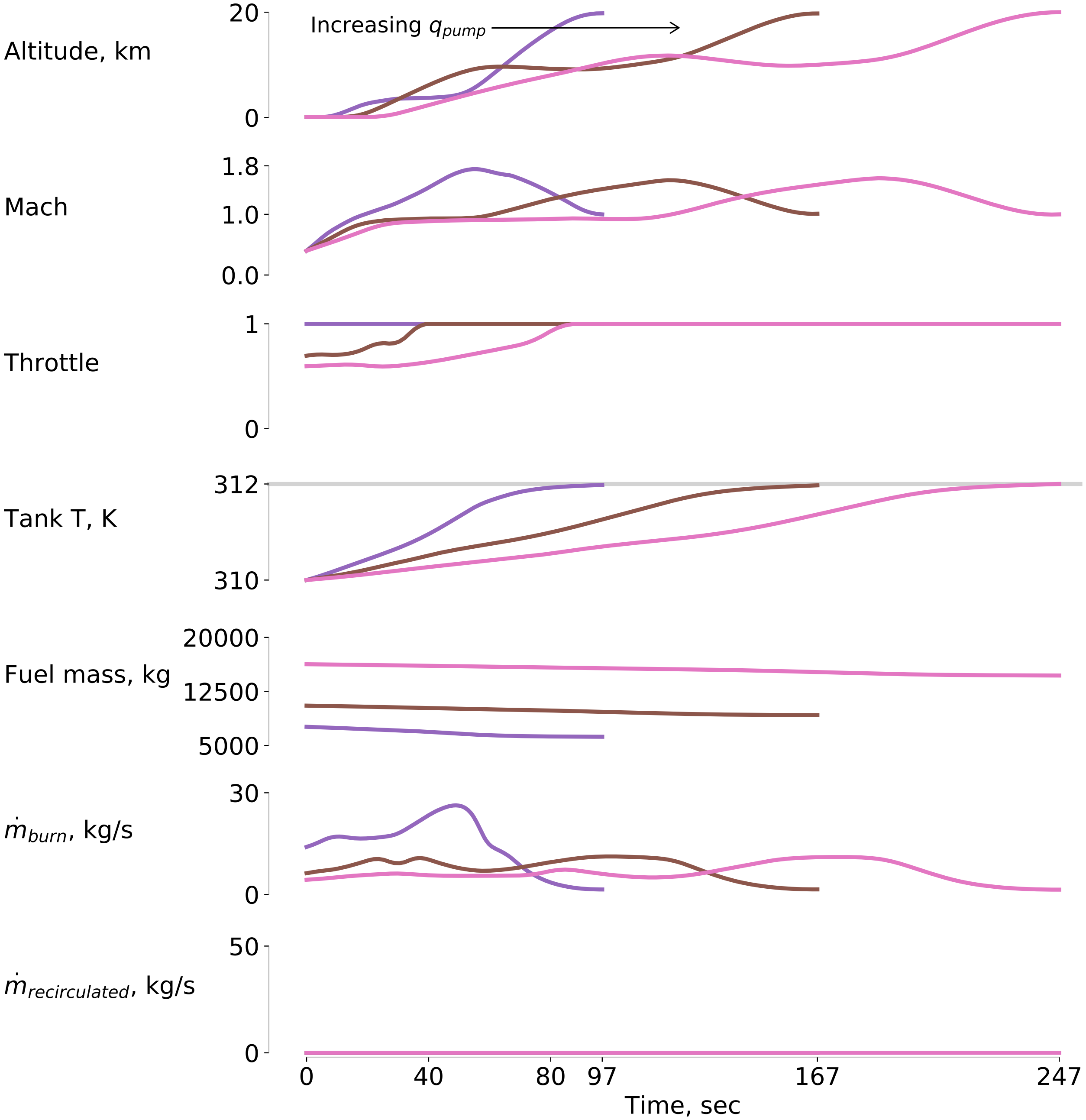


Figure 5.8: In these cases, we only have one thermal source whose magnitude is correlated to the total fuel flow rate. The optimizer both increases on-board fuel and decreases throttle in the worst-case scenario to stay within the thermal constraint.

Table 5.7: Heat loads for results shown in Fig. 5.8

	Color	Recirculation	$\dot{Q}_{\text{env}}$, kW	$\dot{Q}_{\text{sink}}$, kW	q_{pump}, kW/kg	q_{out}, kW/kg
Case 5	Purple	No	0	0	200	0
Case 6	Brown	No	0	0	300	0
Case 7	Pink	No	0	0	400	0

5.2.5 Conclusions

We constructed a fully-coupled aero-thermal-mission model to optimize the trajectory of a supersonic air vehicle subject to thermal constraints. We used a RANS CFD-based surrogate model for the aerodynamic analysis, a surrogate model based on one-dimensional thermodynamic cycle analyses for the propulsion, and a transient fuel thermal management system that tracks the temperature of the fuel in the tank across the mission. We can then perform implicit time integration to obtain the aircraft performance across its mission and optimize it based on our defined control and state parameters.

We optimized seven individual cases with varying heat loads and recirculation abilities. These results showed that the heat loads within a supersonic aircraft can affect the optimal trajectory and operating conditions when the aircraft is thermally constrained. Without advanced FTMS or heat rejection capabilities, the aircraft requires additional fuel to increase its thermal capacitance to stay with the thermal constraints, even if we optimize the trajectory and throttle controls. Introducing a heat exchanger and recirculating warmed fuel back into the tank allowed the aircraft to complete the minimum time-to-climb problem 22 seconds faster than when using a non-recirculating system. These results show that the performance of the next generation of fighter jets may be degraded if thermal effects and FTMS are not considered early enough in the design process.

5.3 Fully coupled aero-thermal-propulsive-mission optimization

The natural extension of the work presented in Sec. 5.2 is to introduce additional physical considerations that more closely model reality. In this section, we examine full mission profiles considering thermal effects from the dual-tank FTMS presented in Sec. 3.4. We again use surrogate models for the aerodynamic and propulsive performance, and thus do not design the aircraft shape or engine in this section. In the problems presented here, the optimizer can control the flight profile and the thermal system flow rates to find the optimal missions considering operational and thermal constraints.

5.3.1 Model setup

All of the models used in this section's studies have been introduced in previous sections. We use the aerodynamics and engine surrogate models presented in Sec. 2.4.1 and the dual tank thermal system from Sec. 3.4. The models used in this section differ from Sec. 5.2 in that we are using the more complex thermal system, a propulsion model that includes installation effects, and reformulated flight dynamics equations for increased robustness.

Figure 5.9 shows how these models are connected in the context of a full optimization problem, later described in Sec. 5.3.3. Although there is some backwards coupling in this model from the flight dynamics and propulsion systems being coupled together, most of the interdisciplinary interactions are resolved at the optimizer level due to the collocation scheme. For example, the coupling between the fuel burned in the engine combustor and the fuel tank mass is resolved by the optimizer.

5.3.2 Model investigations

To better understand how the flight conditions affect the thermal system, we examine the transient performance of the coupled system in steady level cruise flight at two different Mach numbers.

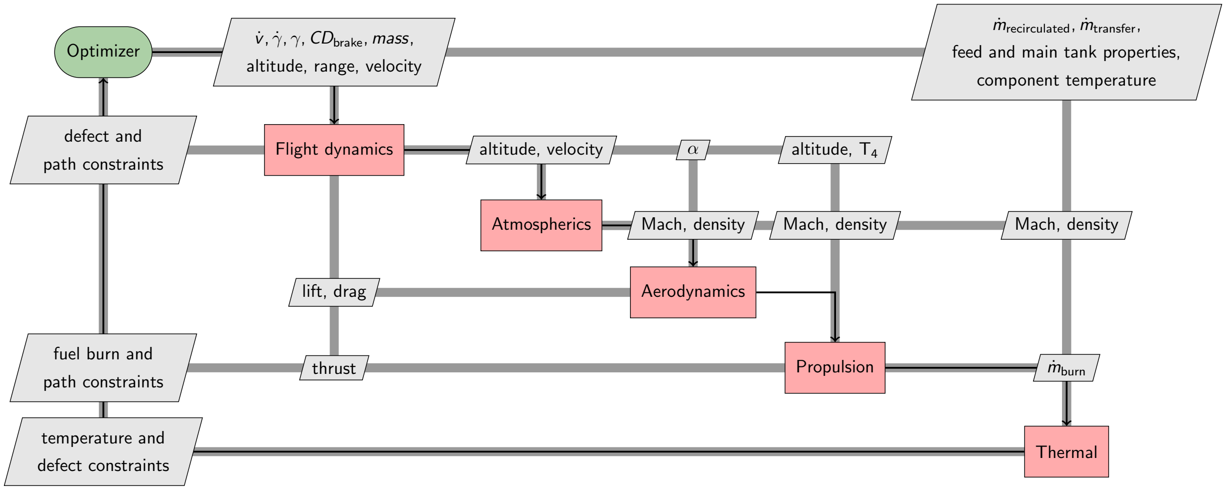


Figure 5.9: XDSM of the fully-coupled problem, including the optimizer, design variables, and constraints, which will be explained in detail in Sec. 5.3.3.

Figure 5.10 shows the trend in avionic component and feed tank temperature as the aircraft flies in steady level flight. The only difference between both cases is the Mach number at which the aircraft flies; the case in orange is at Mach 0.8 whereas the blue case is at Mach 1.5.

After an initial increase in temperature, the component begins to cool down as fuel is recirculated through the thermal system. The recirculated fuel goes through the ducted heat exchanger, which removes the thermal energy introduced by the component. The supersonic case is able to cool the component more effectively due to the increased air mass flow rate through the ducted heat exchanger. This in turn allows more heat to be dissipated in the duct, which lowers the temperature of the fuel, which then enters the feed tank. The lowered temperature of the feed tank, as shown in Fig. 5.10, allows the component temperature to be cooled at a greater rate than in the transonic case.

Based on studies like those presented in Fig. 5.10, we gain a better intuition for the design space. In this context, results from converged optimization problems can be explained in physical terms. This is an extremely complex system, where the interdisciplinary coupling between the

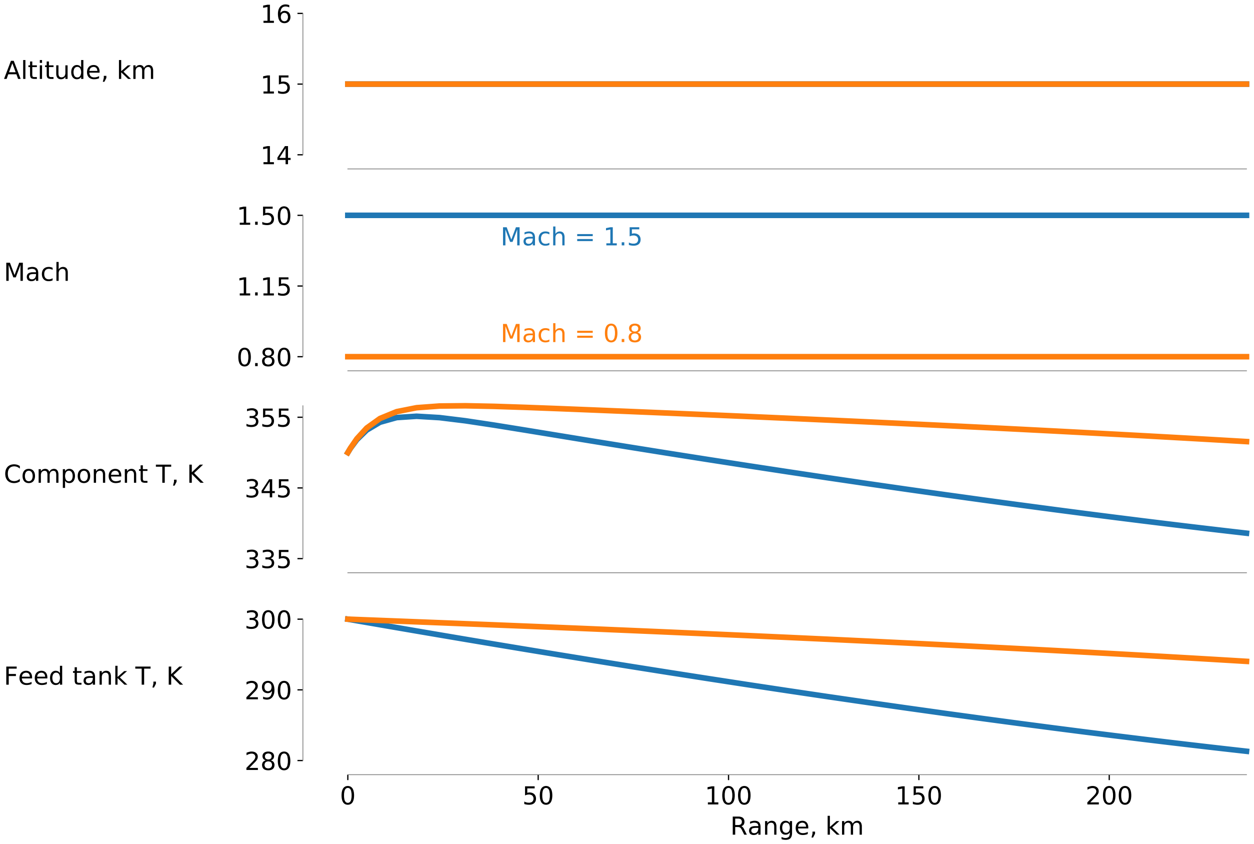


Figure 5.10: Cruising at a fixed Mach number and altitude shows the effects on the thermal system which reaches steady state.

thermal system, air properties, engine performance, and aerodynamics are intrinsically linked.

5.3.3 Optimization problems and results

With a better understanding of the fully coupled model and its design space, we introduce optimization problems and their results in this subsection. Table 5.8 has the complete optimization problem formulation for a reconnaissance mission. Here, we are minimizing the fuel burn for a supersonic mission where the cruise Mach number and altitude are prescribed and the ascent and descent profiles are optimized. Two optimization cases are presented: one without and one with thermal constraints and thermal design variables.

The first case, labeled as "No thermal constraints" in the table, finds the optimal mission without considering the performance of the thermal system, though the thermal states are tracked. The second case, labeled as "Thermal constraints" in the table, includes thermal constraints and thermal design variables. Here, the temperature of the cooled avionics component and the fuel temperature are limited to prevent electronics degradation and fuel overheat, respectively. We would expect the optimizer to increase the fuel flow through the thermal system to better cool the component, or take a different flight path to access air at a lower temperature for the ducted heat exchanger in this problem.

The thermal load coming from the avionics component is set at 20 kW for the ascent and descent phases, and 30 kW for the cruise phase to simulate increased thermal load due to the use of a reconnaissance sensor. The ascent, cruise, and descent time durations are set to 200, 1000, and 200 seconds, respectively. For the case without thermal constraints, the recirculatory flow rate is $\dot{m}_{\text{recirculated}} = 1.0$ kg/s and the main-to-feed tank transfer rate is $\dot{m}_{\text{transfer}} = 1.0$ and both are fixed across the mission. The initial feed tank fuel mass is 1000 kg and the initial temperatures for the main tank, feed tank, and component are all 300 K in both cases.

As highlighted in Table 5.8, these optimization problems are relatively large, as the thermal case has 980 design variables and 1602 functions of interest. These optimization studies are partially enabled by OpenMDAO's sparse coloring algorithms [146]. For the thermal case, coloring reducing the number of independent sensitivity analysis calls from 980 to 258, resulting in a 73.7% improvement in computational cost. This helps to decrease the time between optimization cases and design iterations to increase the rate that the design space can be explored by experts.

Figure 5.11 shows the optimal results of these two cases. Without thermal constraints, the full mission fuel burn is 2323.7 kg, whereas the aircraft burns 2880.9 kg of fuel for the thermally-constrained mission. Examining the case without thermal constraints first, the optimal altitude profile has a straightforward ascent profile where the aircraft accelerates for a short time before climbing the cruise altitude and then accelerating further to the prescribed cruise Mach number. The component temperature increases and reaches steady-state in the ascent phase before increasing when the sensor thermal load is introduced for the cruise phase. Because the descent time duration is fixed at 200 seconds, the aircraft briefly increases altitude while losing speed before continuing its descent.

For the thermally-constrained case, the optimal ascent profile is markedly different than for the unconstrained case. The aircraft focuses on gaining speed instead of altitude to force more air through the ducted heat exchanger, which increases the amount of thermal energy that can be dissipated. This requires the engine to be at full throttle through the ascent at a lower altitude, which is responsible for the increase in fuel burn compared to the thermally-unconstrained case. After reaching Mach 1.5, the aircraft climbs to the cruise altitude and in the process slows to the prescribed cruise Mach number of 1.4. This allows the component and feed tank temperatures to be lower than in the unconstrained case. Although tailoring the flight path helps the thermal system, there are no feasible solutions to this thermally-constrained problem unless the optimizer can also control the recirculated fuel flow rate.

During the ascent phase, the optimizer also increases $\dot{m}_{\text{recirculated}}$ to increase the fuel flow rate through the component and ducted heat exchangers to dissipate more thermal energy. This, coupled with the lower feed tank temperature due to the ascent profile, helps keep the component

Table 5.8: Optimization problem formulations for the fully-coupled supersonic reconnaissance mission

Category	Name	Quantity No thermal constraints	Quantity Thermal constraints	Lower	Upper	Units
Objective	fuel burn	1	1	–	–	kg
Variables	$\dot{\gamma}$	12	12	-0.1	0.1	radians/s
	$\dot{v}$	12	12	-5	5	m/s^2
	CD_{brake}	6	6	0	0.05	
	$\dot{m}_{\text{recirculated}}$	0	18	0	10	kg/s
	$\dot{m}_{\text{transfer}}$	0	18	0	10	kg/s
	mass	83	83	15,000	30,000	kg
	altitude	83	83	0	16	km
	range	83	83	0	–	km
	velocity	83	83	0	1000.	m/s
	γ	82	82	-0.5	0.5	radians
	feed mass	83	83	10	–	kg
	main mass	83	83	10	–	kg
	feed T	83	83	100	1000	K
	main T	83	83	100	1000	K
	component T	83	83	100	1000	K
	m_{pumped}	83	83	–	–	kg
	Total	**944**	**980**			
Constraints	final altitude	1	1	100	100	m
	final Mach	1	1	0.	0.5	
	Mach cruise path constraints	36	36	1.4	1.4	
	altitude cruise path constraints	36	36	13	13	km
	α path constraints	84	84	-15	15	degrees
	T_4 path constraints	84	84	2000	3200	degrees R
	CD_{brake}	24	24	0	–	
	$\dot{m}_{\text{recirculated}}$ path constraints	0	84	0.01	–	kg/s
	feed mass path constraints	0	84	500	–	kg
	$\dot{m}_{\text{transfer}}$ path constraints	0	84	0	–	kg/s
	T_{out} path constraints	0	84	–	100	degrees C
	Component T path constraints	0	84	–	80	degrees C
	main T path constraints	0	84	300	300	K
	range defects	70	70	0	0	km
	altitude defects	70	70	0	0	km
	velocity defects	70	70	0	0	m/s
	γ defects	70	70	0	0	radians
	mass defects	70	70	0	0	kg
	feed mass defects	70	70	0	0	kg
	feed T defects	70	70	0	0	K
	main T defects	70	70	0	0	K
	component T defects	70	70	0	0	K
	m pumped defects	70	70	0	0	kg
	main mass defects	70	70	0	0	kg
	phase continuity constraints	121	121	0	0	
	linkage constraints	24	24	0	0	
	Total	**1181**	**1685**			

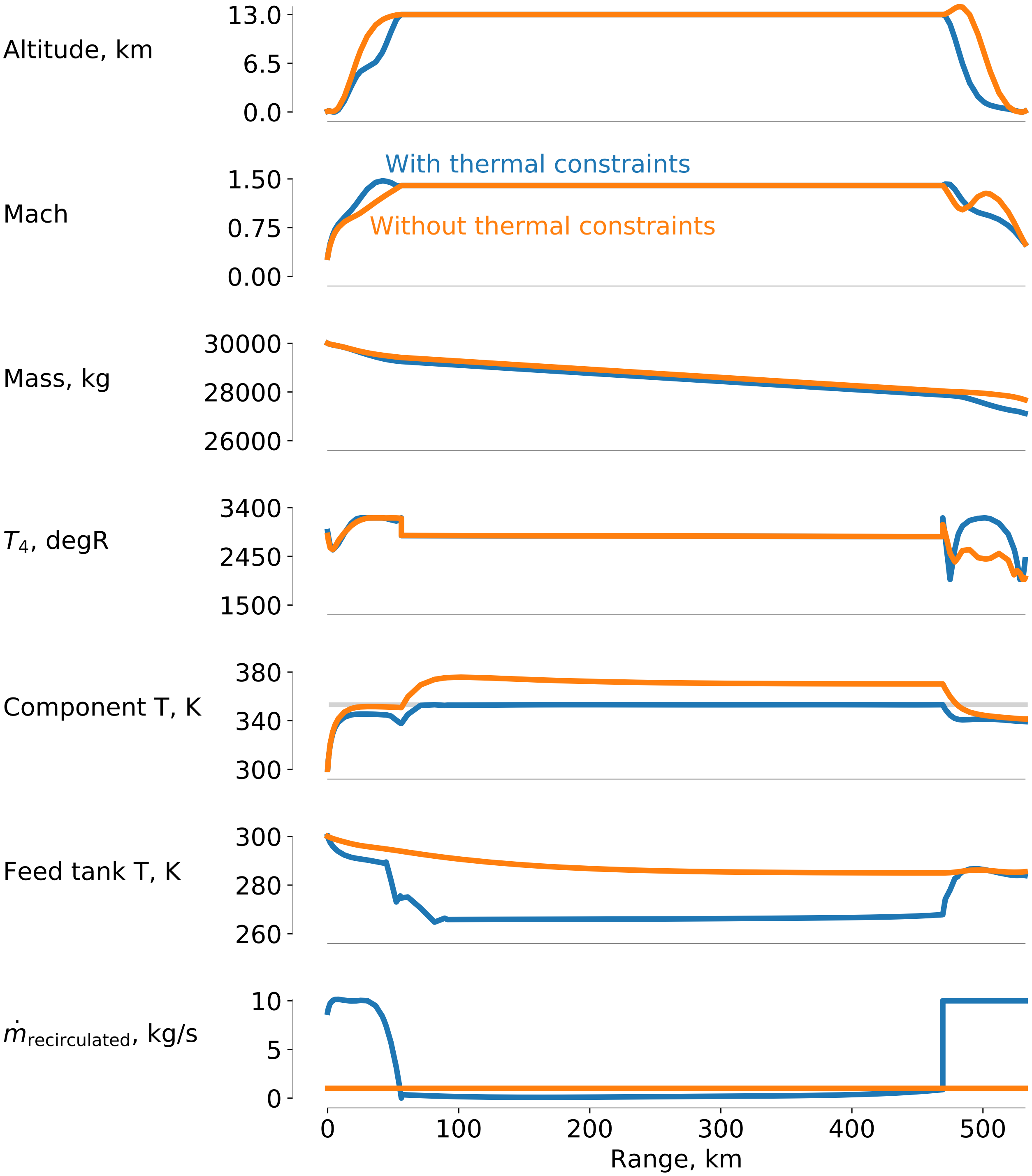


Figure 5.11: Without thermal constraints on the system, the optimal mission takes a more direct ascent profile. In the thermally-constrained case, the optimizer chooses to more quickly increase speed to force more air through the ducted heat exchanger to dissipate more heat in the ascent phase. The thermal constraint for the avionic component temperature is shown in light gray.

temperature below its upper limit. In the problem presented here, there are multiple feasible answers because the thermal constraints can be met through multiple fuel flow histories. Although we are not capturing the added energy cost of pumping fuel as coolant in the thermal system, introducing that coupling would affect the optimal $\dot{m}_{\text{recirculated}}$ profile and reduce the number of feasible solutions. This optimization problem is dependent on having a good initial starting condition due to the collocation scheme.

Lastly, we examine the optimal flight profiles when the optimizer has more freedom to tailor the mission. Here we track the thermal system states but do not constrain the optimization problem based on these states. Instead, we focus on the optimal flight path given the additional freedom in speed and altitude profiles.

Without prescribing the Mach number and altitude profiles for the cruise phase, we would expect a lower fuel burn. Figure 5.12 shows the thermally-unconstrained case from before and the optimal result from a more flexible problem formulation with the cruise requirements removed. The x-axis is time, not range, so the total distance traveled in the two missions is not necessarily equal. The ascent, cruise, and descent phases again have fixed durations of 200, 1000, and 200 seconds, respectively.

With the additional flexibility, the optimizer rounds out the altitude profile and takes a more conservative flight path. Without the prescribed cruise Mach number, the more flexible mission cruises in the transonic regime, which is more fuel efficient. This more efficient profile results in a fuel burn of 1617.6 kg, compared to the 2323.7 kg for the constrained cruise case. The two missions do not fly the same distance, only the same amount of time, so this fuel burn comparison is not entirely valid, though it shows how allowing the optimizer to control the entire mission can result in lower fuel burn. We are not constraining the thermal system here, so the optimized flight profiles do not depend on thermal dissipation performance.

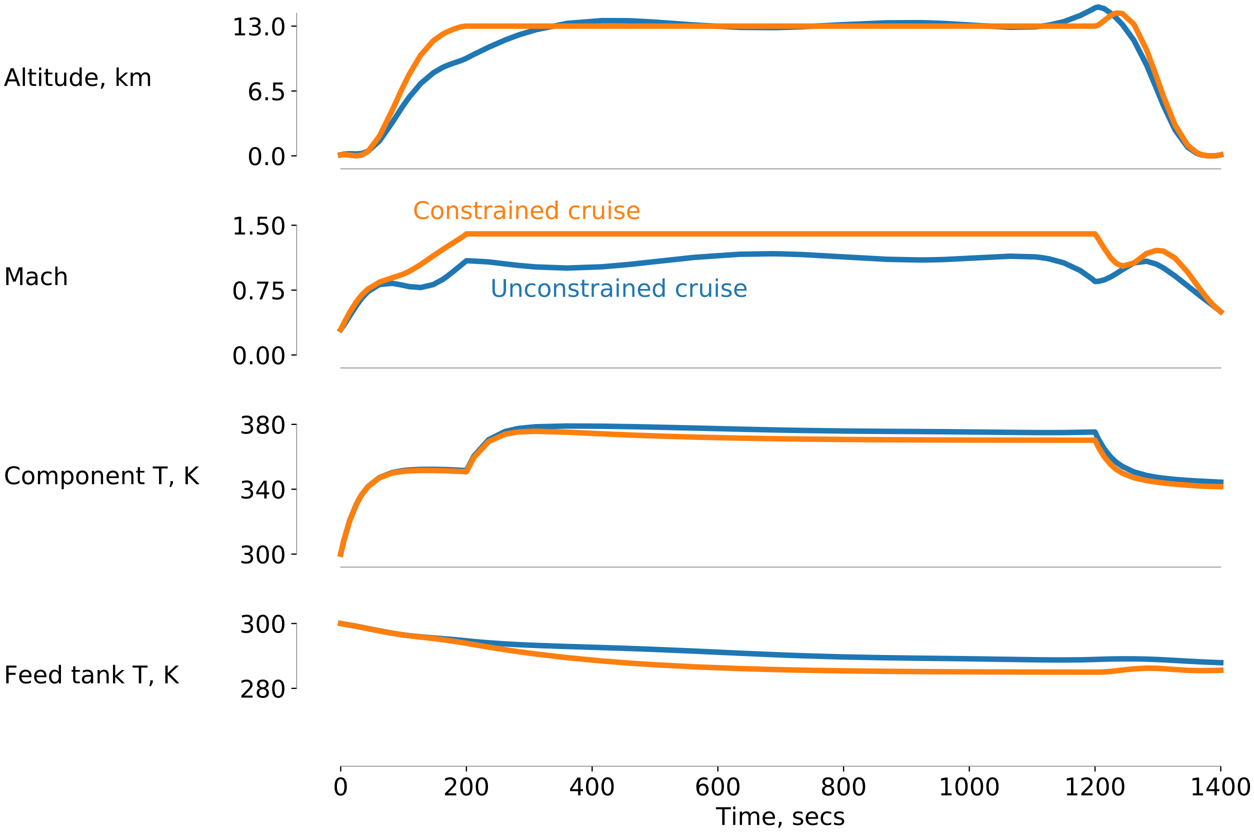


Figure 5.12: Without the Mach and altitude constraints on the cruise phase, the optimizer takes a more conservative ascent profile and cruise at a lower Mach number. The x-axis here is time, so the two aircraft do not fly the same total distance due to their different flight speeds.

5.3.4 Conclusions

In this section, we showed how the aero-thermal-propulsive-mission problem can be formulated and solved to study optimal trajectories of path-dependent systems. We compared the optimal flight path of a supersonic aircraft while tracking its thermal states with and without thermal constraints. The thermal constraints affected both the optimal coolant flow rates as well as the optimal flight path due to the tight coupling between the disciplines. We also showed that given additional design freedom, the optimizer can further improve aircraft effectiveness by tailoring the flight profile. The ability to study these tightly-coupled path-dependent systems helps understand trade-offs in potential aircraft designs.

CHAPTER 6

Concluding Remarks

6.1 Summary of results

Aircraft performance is often limited by path-dependent effects, such as an aircraft's thermal history or control actuator position. Designers must consider these path-dependent effects when creating aircraft systems to optimize performance. Because these systems are multidisciplinary and those disciplines are strongly coupled, single-disciplinary studies alone cannot accurately capture the performance of aircraft. This dissertation focuses on enabling large-scale design optimization of path-dependent aircraft systems.

To design aircraft considering their path-dependent performance, the flight mission must be modeled and optimized. Chapter 2 introduces the ODEs and integration methods used to track the aircraft controls and states. I developed a low-cost open-source aerostructural design tool, *OpenAeroStruct*, which I then used for physics-in-the-loop mission optimization of a morphing aircraft wing. This chapter also introduced the supersonic air vehicle model later used in the dissertation. I optimized multiple full supersonic missions and showed how the collocation discretization affected optimal results.

Thermal systems, which are inherently path-dependent due to the thermal history's effect on future performance, are studied in Chapter 3. I first present a simplified FTMS where we solve to create an ODE that tracks the fuel tank's temperature before developing a modular dual-tank thermal system that includes both component and ducted heat exchangers. This chapter closes with analysis sweeps of the dual-tank thermal system that show how the atmospheric and flight conditions affect the efficacy of the aircraft's thermal dissipation abilities.

Chapter 4 examines engine design and optimization in detail. I present the use of surrogate-enhanced guess tables within a nested solver hierarchy to improve engine model convergence, which is an established problem in the field. I also detail a low-cost way of modeling supersonic inlet installation losses using surrogate models. These techniques are then used to enable multipoint VCE optimization of a mixed-flow turbofan. The VCE capability greatly affects where the engine elements operate on performance maps, which shows how the engine performance can vary across an entire flight mission.

These disciplinary analyses are combined in Chapter 5, where I solve multiple engine-thermal and aero-thermal-propulsive-mission optimization problems. First, the coupling between the propulsion and thermal systems is investigated using an engine model with a bypass duct heat exchanger, where we find that the inclusion of a thermal lifting system improves engine performance by at least 1.4% for our cases. Then, I perform minimum time-to-climb mission optimization with different thermal constraints to see how the optimal flight trajectory differs based on thermal requirements. Lastly, full supersonic missions are optimized considering thermal constraints using the dual-tank thermal system. The aerodynamics and propulsion systems are represented using surrogates obtained from the models presented in previous chapters. These final optimization results show how the optimal flight path is tightly coupled to the path-dependent thermal system performance.

6.2 Contributions

1. *I developed an open-source low-cost aerostructural wing design tool, OpenAeroStruct, useful for efficient physics-in-the-loop mission optimization.*

   OpenAeroStruct [132] was used in this dissertation to perform mission optimization with morphing-wing aerostructural analyses in the loop [90]. OpenAeroStruct has proven useful in both research and education by filling a need that other existing tools did not satisfy. Specifically, its low-cost aerostructural wing design capabilities using gradient-based methods did not exist elsewhere. This tool is currently being used for studies at Michigan, NASA, MIT, SUPAERO, and many other institutions and companies.

2. *I contributed to a modular thermal system modeling framework and demonstrated it using a dual-tank FTMS and bypass duct heat exchanger.*

   This thermal framework provides efficient derivatives for gradient-based optimization and I expanded it by introducing a ducted heat exchanger, thermal lifting system, and transient reservoir [188, 189]. Using this modular setup, different thermal system architectures can be studied easily without needing to re-derive the ODEs of interest.

3. *I advanced engine design capabilities by creating a more robust model using surrogate-enhanced guess tables and presented an inexpensive method of computing supersonic inlet installation drag.*

   Engine design optimization is challenging due to convergence difficulties, but I addressed that by using surrogate-enhanced guess tables to create robust engine models. This allows the optimization method to query the engine model across the entire flight envelope, which was not possible using the prior state-of-the-art techniques [85]. Additionally, I created and implemented a low-cost way of modeling engine installation effects using a surrogate model fit with data from NASA inlet trade studies.

4. *I performed the first fully-coupled gradient-based optimization of an engine with a bypass duct heat exchanger.*

   Previous work on bypass duct heat exchangers only examined analyses or gradient-free optimization of the thermal system alone. I designed the engine and the thermal system simultaneously using gradient-based methods, which had not been done before [188]. I found that there is direct coupling between the optimal engine design and thermal constraints, and that the thermal lifting system allows the optimizer to tailor engine performance to be thermodynamic optimal. This proves the need to model both thermal and propulsion systems simultaneously, especially when using a bypass duct heat exchanger.

5. *I presented the first results from aero-thermal-propulsive-mission optimizations and showed how the coupled subsystems affect optimal aircraft performance.*

   Using a supersonic aircraft as the application case, I showed that the optimal flight profile depends on thermal system constraints. By dividing the mission into multiple phases, I showed that we can examine different physical considerations for each phase. I presented how operational flight constraints limit aircraft performance and how optimizing multiple subsystems of the aircraft simultaneously with its flight path increases overall mission effectiveness.

A subset of these contributions are also presented graphically in Fig. 6.1. This figure highlights where I contributed to both single-disciplinary systems as well as the fully-coupled multidisciplinary problem.

6.3 Recommendations for future work

1. *Use physics-in-the-loop models to enable larger design freedom*

   Using surrogate models for disciplinary analyses restricts how much design freedom of the fully-coupled system. Incorporating CFD, engine modeling, or other physics models directly in the full problem would enable design of the aerodynamic shape or propulsion system, for example. Although this would greatly increase the computational cost of the optimization, high-performance computing clusters can reduce the wall-clock run times of the problems.

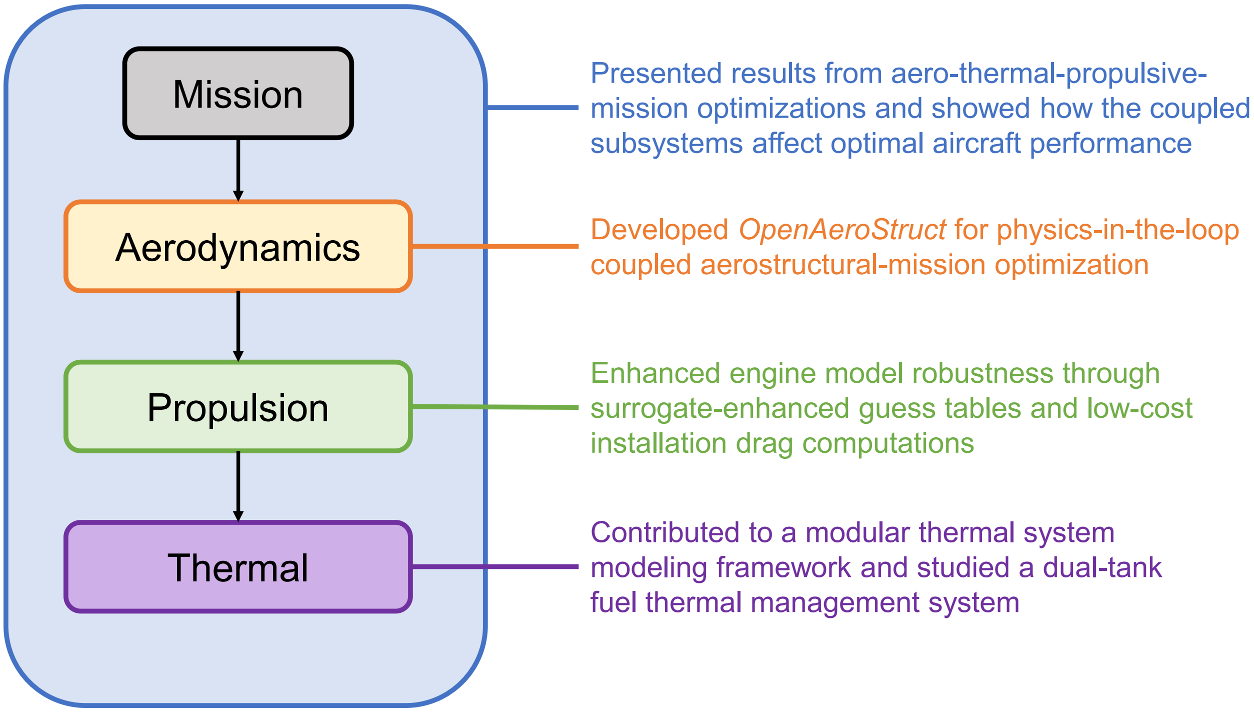


Figure 6.1: Visualization of my contributions both in the single-disciplinary fields and also in the fully-coupled multidisciplinary system context.

Additionally, any analyses introduced within the optimization loop must provide efficient derivatives to enable large-scale gradient-based optimization.

2. *Use more complex disciplinary models in the fully-coupled problem*

   Any one of the disciplinary models presented here could be further developed to include more physical considerations. For example, the dual-tank thermal system could be modified to include more heat exchangers from other thermal sources in the aircraft, or to include an oil-engine coolant loop as well. Additionally, the engine inlet installation effects could be modeled using higher fidelity methods, including using inlet recovery cane curves instead of mil-spec recovery amounts, or even directly modeling the inlet instead of using curve-fit data. What levels of complexity and fidelity is required depends on the specific aircraft, mission type, and design study being conducted. For instance, if morphing control surface actuator performance is being studied across the mission, the propulsion system may not need a high level of fidelity because the coupling between the two disciplines is minimal.

3. *Examine other aircraft and mission types in detail*

   New aircraft configurations can be designed and assessed through the path-dependent optimization process detailed in this dissertation. Other missions of interest might arise in urban air mobility, supersonic combat, radar area coverage, or firefighting missions, for example. These new configurations might have non-intuitive coupling between subsystems that can be efficiently evaluated using the methods presented here.

4. *Incorporate other disciplinary analyses as needed for other aircraft applications*

   Other disciplinary analyses will need to be introduced to this process as new aircraft configurations are studied, and new or existing analyses can be integrated as needed. Control systems and aeroeleastic effects could help optimize aircraft with active aeroelastic controls and high-aspect ratio wings. For electric aircraft, battery and other electrics systems would have to be modeled in detail. Future aircraft applications may require disciplinary analysis capability that does not exist yet.

APPENDIX A

OpenAeroStruct

A.1 Summary

In this appendix, we discuss OpenAeroStruct[1], an open-source Python-based coupled aerostructural analysis and design optimization tool. OpenAeroStruct couples the VLM and FEA methods using six degree-of-freedom (DOF) spatial beam elements with axial, bending, and torsional stiffness.

OpenAeroStruct is developed within the OpenMDAO framework [77], a NASA-developed open-source software framework for MDO. OpenAeroStruct computes derivatives for the aerostructural system using the coupled adjoint method [63, 70, 190]. The aerodynamic forces and structural displacements are transferred between disciplines in a consistent and conservative manner. This process is simplified because the aerodynamic and structural meshes have the same spanwise discretization, so no interpolation is necessary to transfer the loads or displacements. A variety of solvers can be used to converge the coupled aerostructural system, including block Gauss–Seidel, GMRES, or LU decomposition for the linear system, and nonlinear block Gauss–Seidel or Newton for the nonlinear system. The standard aerostructural optimization problem is a fuel-burn minimization using the Breguet range equation, and the design variables consist of twist distribution, spar thickness distribution, and planform variables. The constraints ensure that lift equals weight

---

[1] `https://github.com/mdolab/openaerostruct`

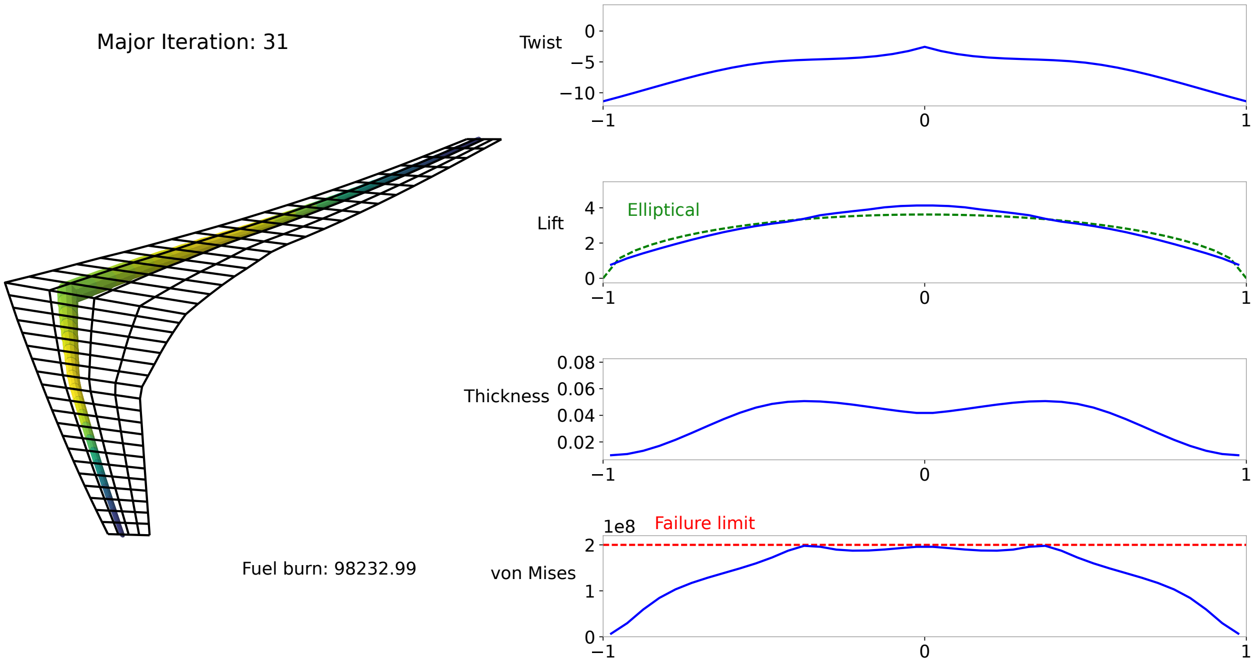


Figure A.1: Screenshot of interactive visualization tool included in OpenAeroStruct. Users can explore the optimization history by stepping through each iteration and examining the wing model, design variables, and output function values.

and the structural spar does not fail. The modular implementation in OpenMDAO makes it easy to reformulate the optimization problem, e.g., to minimize the sum of direct operating cost and acquisition cost, and to subsequently update the derivative computation.

OpenAeroStruct is unique because of its modular implementation and the efficient coupled gradient computation. The model is decomposed into many low-level computations. Each of these computations is implemented as an OpenMDAO *component*, and OpenMDAO computes the overall model-level derivatives given component-level derivatives of the outputs with respect to the inputs of each component. The components use hand-derived analytic partial derivatives computations. We include multiple visualization tools to examine the wing model and design variables during the optimization. Figure A.1 shows a view from the included interactive GUI-based visualization tool.

The main objective of OpenAeroStruct is to provide a physically meaningful multidisciplinary model that can be used to obtain low-order approximations of aircraft performance or to compare solution algorithms. We evaluate on-design aerostructural performance using the Breguet range equation to approximate fuel burn. OpenAeroStruct is useful for a variety of educational areas, including aircraft design, MDO, uncertainty quantification, and numerical solution algorithms.

This appendix caters to three different sets of users:

1. For educators, this appendix presents OpenAeroStruct as a hands-on learning tool for the classroom. The modular implementation within a framework, the coupling within the disciplines, and the simplicity of the physics make it a practical teaching tool, as we have already demonstrated in the classroom setting at two universities. The individual aerodynamic and structural models are well known, but the way that they are coupled to provide analytic coupled derivatives is novel. Educators should read Section A.4 for examples of classroom usage and may wish to read Section A.3 for an understanding of the methodology.

2. For researchers, this appendix presents OpenAeroStruct as a physics-based multidisciplinary model that is computationally inexpensive and uses the adjoint method to compute the coupled derivatives. This yields scalability that makes OpenAeroStruct useful for benchmarking in MDO research [191–194], where it has been used to test solvers, MDO architectures, uncertainty quantification (UQ) methods, etc. Researchers should see Section A.5 for examples of research insights gained from OpenAeroStruct. Researchers may also be interested in Section A.3, which explains the underlying system setup and capabilities.

3. For students, this appendix serves as a reference for the aerodynamic and structural theory of OpenAeroStruct. We document the equations for the VLM and FEA analyses in the model, the equations describing the consistent and conservative load and displacement transfer schemes, and the numerical solvers. Detailed code documentation, examples, and tutorials can be found in the online documentation linked in the GitHub repository. Students will find the background theory in Section A.2 and the details of the coupled system setup

in Section A.3 helpful.

This appendix is organized as follows. First, we detail the existing theory and our implementation for the aerodynamic and structural models (Section A.2). Then we explain the coupled aerostructural solver implementation and the optimization problem formulation in Section A.3. In Section A.4, we present results from research applications of OpenAeroStruct and from its use in graduate courses. We review the research and educational outcomes achieved to date using OpenAeroStruct in Section A.5.

A.2 Background

A.2.1 Motivation

In aircraft wing design, aerodynamics and structures are two of the most important disciplines. Increasing span reduces aerodynamic drag but increases structural weight, so a well-designed wing needs to balance these two trends. Changing the twist distribution to concentrate the lift inboard may hurt aerodynamic efficiency but improves structural efficiency. For wing design, it is important to account for the aerostructural coupling because the aerodynamic loads affect the structural deflections, which in turn affect the aerodynamic loads. The true deflected shape of the wing is thus different from the shape found by applying the aerodynamics loads computed based on the undeflected shape. In addition to finding the true shape of the wing, we must also consider design tradeoffs between wing shape and structural sizing. This is a quintessential multidisciplinary design problem that motivated the development of MDO in both low-fidelity [195] and high-fidelity studies [57, 64].

The interactions between aerodynamics and structures can be captured with inexpensive physics-based models, such as the VLM for aerodynamics and 1-D FEA for structures. A coupled 1-D VLM–FEA model enables optimization of the spanwise distribution of the aerodynamic twist, planform, and structural thicknesses. However, computationally efficient aerodynamic and aerostructural optimization software is not freely available, even for low-fidelity models.

Existing free low-fidelity aerodynamic analysis tools have limited capabilities. XFOIL [196] is restricted to 2-D problems, Tornado [197] does not compute derivatives, and AVL [198] does not natively interface well with optimizers. Similarly, freely available tools for structural analysis, such as Frame3DD [199] and Calculix [200], do not provide derivatives and interface poorly with optimizers. High-fidelity tools for aerostructural design optimization have been developed, but they require the use of parallel high-performance computing platforms [70, 201]. Although software exists for lower fidelity aerodynamic or structural analysis alone, no open-source tool is able to perform coupled aerostructural analysis and design optimization.

In the gradient-based optimization of multidisciplinary systems, computing the coupled derivatives efficiently and accurately is critical and requires significant effort [70]. Gradient-based optimization scales well (often linearly) with the number of design variables because it uses the gradient information to find an efficient path from the initial point to the optimum. The adjoint method computes the gradient in a given optimization iteration at a cost that is comparable to that of the analysis, and that does not increase with the number of design variables. However, implementing the adjoint method is time- and effort-intensive because it requires modification of the source code for a model, in contrast to methods such as finite differences that treat the model as a black box.

We designed OpenAeroStruct as an MDO model that uses gradient-based optimization to solve a physically meaningful problem. The code is entirely open-source, so users can examine exactly what it is doing and expand its capabilities by building on the existing components. Analytic gradients are provided for each analysis component to enable efficient gradient-based optimization through the adjoint method, which has not previously been done for open-source wing design tools. The code is written using the OpenMDAO framework, so users can quickly and easily change the optimization formulation, the optimizer, and the system solvers. The fully functioning pure

Python implementation means that the code can be used without worrying about dependencies or complicated installation procedures; OpenAeroStruct works on macOS, Windows, and Linux.

A.2.2 Aerodynamics model

The aerodynamics model uses a VLM [202] to compute the aerodynamic loads acting on the lifting surfaces. This combines multiple modern numerical lifting-line theory (LLT) models, as described by Phillips and Snyder [203]. The VLM model is more general than the LLT model because it models low aspect ratio wings, swept wings, and delta wings more accurately [202]. The theory of VLM is well established, and we create and use our own implementation so that we can easily obtain the relevant derivatives.

To set up the model, we follow the process outlined by Anderson [202]. A complete derivation of the theory is not provided here, but we summarize the important points below. Given a structured mesh defining a lifting surface, we can compute the aerodynamic properties by examining the circulation distribution. We do this by modeling the lifting surface using horseshoe vortices to represent the vortex system of a wing. Each horseshoe vortex consists of a bound vortex in the spanwise direction and two trailing vortices that extend into the freestream direction. Figure A.2 shows a single horseshoe vortex on a lifting surface. We now examine the mathematical formulation used to calculate the wing circulation.

A vortex filament induces a flow field in the surrounding space. The strength of a vortex filament is its circulation, which produces lift on a surface. The Biot–Savart law relates the velocity of the flow field at an arbitrary point P caused by a segment $d\mathbf{l}$ of a vortex filament with circulation strength Γ via

$$d\mathbf{V} = \frac{\Gamma}{4\pi} \frac{d\mathbf{l} \times \mathbf{r}}{|\mathbf{r}|^3}.$$

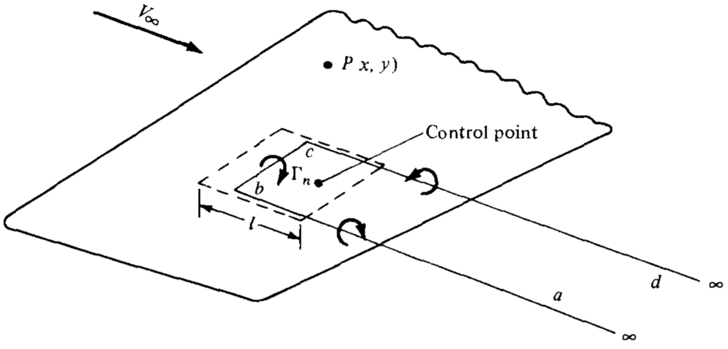


Figure A.2: Single horseshoe on a lifting surface [202].

Integrating over a semi-infinite straight vortex filament, we obtain

$$V = \frac{\Gamma}{4\pi h},$$

where h is the distance from point P to the finite start point of the vortex filament.

Helmholtz's vortex theorems [202] state that:

1. The strength of a vortex filament is constant along its length.
2. A vortex filament cannot end in a fluid; it must extend to the boundaries of the fluid (which can be $\pm\infty$) or form a closed path.

As Fig. A.2 illustrates, a horseshoe vortex is comprised of semi-infinite vortices extending from point b to point a and from c to d, and a bound vortex extending from b to c. The circulations for each of these vortex segments are of equal magnitude and have consistent directions.

To model a lifting surface, we superimpose multiple horseshoe vortices over the span. This corresponds to a Weissinger LLT model, where there is only one horseshoe element in the chordwise direction. VLM is the Weissinger LLT method extended to allow multiple sets of horseshoe vortices in the chordwise direction.

The panel in Fig. A.2 from the mesh of the lifting surface corresponds to the dashed lines, while the horseshoe vortex associated with that panel is drawn with solid lines. For a panel of length l, the bound vortex is at a distance of $\frac{1}{4}l$ from the front of the panel. Additionally, we define a control point at the centerline of the panel $\frac{3}{4}l$ from the front of the panel. We enforce a flow tangency condition at this control point that specifies that the velocity normal to the panel must be zero. Physically, this means that flow cannot go through the lifting surface. By imposing flow tangency conditions at each of the control points for all of the horseshoe vortices on the lifting surface, we obtain the linear system

$$\mathbf{A}\,\boldsymbol{\Gamma} = -\mathbf{V}_\infty \cdot \mathbf{n},$$

where $\mathbf{A}$ is the aerodynamic influence coefficients matrix, $\mathbf{V}_\infty$ is the freestream velocity, and $\mathbf{n}$ is the normal to the panel. We can solve this linear system to obtain the circulation strengths for each of the horseshoe vortices.

Now that we have the circulation strengths of the vortices, we can compute the aerodynamic forces acting on each individual panel using

$$\mathbf{F}_i = \rho \Gamma_i (\mathbf{V}_\infty + \mathbf{v}_i) \times \mathbf{l}_i,$$

where $\mathbf{v}_i$ is the induced velocity at the center of the bound vortex, and $\mathbf{l}_i$ is the bound vortex vector. The length and direction of the bound vortex directly influence the forces and allow for different width horseshoe vortices across the span.

With the sectional panel forces we can compute the lift, drag, and other lifting surface metrics. The lift and drag correspond to the components of the force vector in the upward and freestream directions, respectively. We also compute the skin friction drag using flat-plate-based estimates [204, Sec. 12.5.3]. This skin friction drag estimate is based on the airfoil thickness-to-chord ratio, the Reynolds number, and other aircraft and flow properties. The drag estimate is adjusted using a form factor, which accounts for pressure drag due to flow separation. The semi-empirical models used for drag estimation are considered valid up to the drag-divergence Mach number.

A.2.3 Structural model

For the structural model, we use a finite element method (FEM) approach that uses spatial beam elements, resulting in six DOFs per node. The spatial beam element is a combination of truss, beam, and torsion elements, which means that it simultaneously carries axial, bending, and torsional loads. We implement our own structural model using well-established theory so we can easily obtain the relevant derivatives.

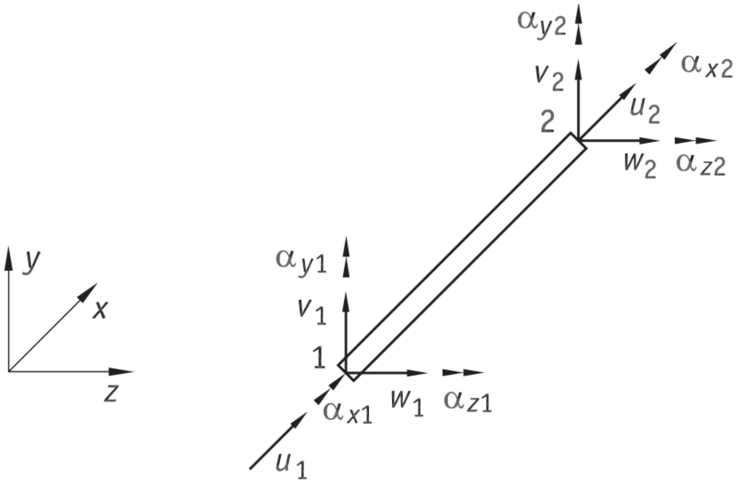


Figure A.3: 6-DOF spatial beam element [205].

Each spatial beam element has 12 DOFs in total, as shown in Fig. A.3. At each end of the beam, there are three translational displacements in the x, y, and z directions, and three rotational DOFs with respect to the x, y, and z axes. The displacements and rotations shown in Fig. A.3 are in the local coordinate frame of the element. The axial DOFs are u_1 and u_2; the z-plane bending DOFs are v_1, v_2, α_{z1}, and α_{z2}; the y-plane bending DOFs are w_1, w_2, α_{y1}, and α_{y2}; and the torsion DOFs are α_{x1} and α_{x2}. During the assembly of the global stiffness matrix, transformation matrices are used to convert the element stiffness matrix from the local frame to the global frame aligned with the x, y, and z axes. The stiffness matrix for a single element is given by

$$[k]_e = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & 0 & -k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12k_2^z & 0 & 0 & 0 & 6k_2^z l & 0 & -12k_2^z & 0 & 0 & 0 & 6k_2^z l \\ 0 & 0 & 12k_2^y & 0 & -6k_2^y l & 0 & 0 & 0 & -12k_2^y & 0 & -6k_2^y l & 0 \\ 0 & 0 & 0 & k_3 & 0 & 0 & 0 & 0 & 0 & -k_3 & 0 & 0 \\ 0 & 0 & -6k_2^y l & 0 & 4k_2^y l^2 & 0 & 0 & 0 & 6k_2^y l & 0 & 2k_2^y l^2 & 0 \\ 0 & 6k_2^z l & 0 & 0 & 0 & 4k_2^z l^2 & 0 & -6k_2^z l & 0 & 0 & 0 & 2k_2^z l^2 \\ -k_1 & 0 & 0 & 0 & 0 & 0 & k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12k_2^z & 0 & 0 & 0 & -6k_2^z l & 0 & 12k_2^z & 0 & 0 & 0 & -6k_2^z l \\ 0 & 0 & -12k_2^y & 0 & 6k_2^y l & 0 & 0 & 0 & 12k_2^y & 0 & 6k_2^y l & 0 \\ 0 & 0 & 0 & -k_3 & 0 & 0 & 0 & 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & -6k_2^y l & 0 & 2k_2^y l^2 & 0 & 0 & 0 & 6k_2^y l & 0 & 4k_2^y l^2 & 0 \\ 0 & 6k_2^z l & 0 & 0 & 0 & 2k_2^z l^2 & 0 & -6k_2^z l & 0 & 0 & 0 & 4k_2^z l^2 \end{bmatrix},$$

where

$$k_1 = \frac{EA}{L} \quad , \quad k_2^z = \frac{EI_z}{L^3} \quad , \quad k_2^y = \frac{EI_y}{L^3} \quad , \quad k_3 = \frac{GJ}{L},$$

and E is the Young's modulus, A is the beam cross-sectional area, L is the beam length, G is the shear modulus, J is the polar moment of inertia, and the Is are the second moments of area about the three local coordinate directions.

In OpenAeroStruct, spatial beam elements are always connected end-to-end in a single sequence, so the resulting global stiffness matrix exhibits a banded structure where stiffness submatrices are added on block diagonals. Once the stiffness matrix has been assembled, OpenAeroStruct solves the linear system $\mathbf{Ku} = \mathbf{f}$, where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u}$ is the vector of displacements and rotations at the nodes, and $\mathbf{f}$ are the forces and moments acting at the nodes.

A.3 Methodology

The aerodynamic and structural models described in the previous section are well known. In this section, we describe our implementation of the coupled solver derivative computation for these two disciplines. The implementation relies on the OpenMDAO, so we include a brief overview of this framework. Then we explain the load and displacement coupling as well as the implementation of the aerostructural system. Finally, we detail the aerostructural optimization formulation.

A.3.1 OpenMDAO framework

We first provide an overview of the OpenMDAO framework since it is the computational modeling software within which OpenAeroStruct is built. OpenMDAO is an open-source NASA-developed framework for multidisciplinary design, analysis, and optimization with a focus on gradient-based approaches [77]. It is unique because it facilitates the efficient computation of the derivatives needed for optimization using methods such as the adjoint method.

As with other software frameworks, its primary function is to enable the modular construction of computational models, where a larger, more complex model is decomposed into smaller units of code, called components, that are simpler and more portable. The components are partitioned into *groups* that can in turn be part of other groups, forming a hierarchy tree. OpenMDAO performs data transfers between components, potentially across processors in parallel settings, and includes nonlinear and linear solvers with support for matrix-free or sparse linear algebra. Any nonlinear or linear solver can be applied in each group or component in the hierarchy tree. This means that OpenMDAO in general uses hierarchical solution approaches: e.g., a Newton solver can be used in a component while the group containing that component can use a nonlinear block Gauss–Seidel solver that calls its children in sequence, including the component with the Newton solver.

A unique and extremely useful feature of OpenMDAO is its ability to compute derivatives using a wide range of methods including the adjoint method. It does this via the MAUD architecture [67], which enables the computation of total (model) derivatives using a unified equation [65]. This matrix equation unifies the chain rule, the direct method, the adjoint method, a coupled form of the chain rule, and hybrid methods. For OpenAeroStruct, the significance of the unification is that our task is limited to computing the partial derivatives of each component's outputs or residuals with respect to its inputs. If we then specify the appropriate linear solvers, OpenMDAO solves the unified equation to compute the needed derivatives. This amounts to the adjoint method if we set up a purely aerodynamic or purely structural OpenAeroStruct model, or the coupled adjoint method if we include both disciplines [70, 190]. If, in a different setting, we set up a model that has no states and residuals, solving the unified equation would be equivalent to applying the chain rule to assemble the partial derivatives to form total derivatives.

A.3.2 Load and displacement transfer

In OpenAeroStruct, the load and displacement transfer is simplified by the assumption that the same spanwise discretization is used for the aerodynamic and structural models. The nodes of the structural mesh and the spanwise sections of the aerodynamic mesh are computed from the same wireframe mesh used for the wing. The structural nodes have the same spanwise discretization as the wireframe mesh and are placed a distance from the leading edge in the chordwise direction based on a specified percentage of the local chord. Likewise, each spanwise section of the aerodynamic mesh is obtained by uniformly splitting the corresponding section of the overall wireframe mesh into the desired number of edges.

The load and displacement transfer scheme used in OpenAeroStruct satisfies the requirements of being *consistent* and *conservative* [190]. Consistency states that the sum of the nodal forces and moments obtained via the load transfer must be equal to the forces and moment resultants computed from the continuous pressure and shear force distributions on the element. Since the resultants are computed by integrating over a region, an infinite number of choices of nodal forces and moments provide consistency. Conservativeness states that the virtual work done by the forces over virtual displacements on the aerodynamic and structural meshes are equal. We now explain the load and displacement transfer schemes and then show that they are consistent and conservative.

For the load transfer, our objective is to transfer the traction $\vec{T}$ from each panel to the structural nodes, as shown in Fig. A.4. The traction is assumed to be uniformly distributed over the panel, where the net force is equal to that computed by the VLM. Since the edges of the panels are aligned with the structural nodes, half of the traction on the panel is applied to each of the two structural nodes. The nodal force and moment vectors on either the left or right structural node are given by

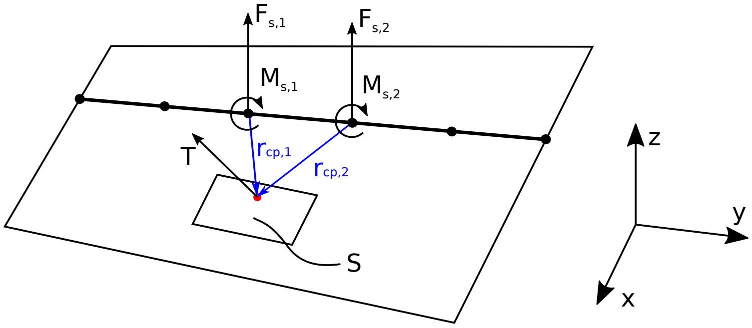


Figure A.4: Load transfer from a panel on the aerodynamic mesh to adjacent structural nodes. Since the force is assumed to be uniformly distributed on the panel, we can compute the resultant moments on the nodes using the vectors pointing to the aerodynamic centers of pressure.

$$\vec{F}_{s,i} = \int_{\text{panel}} \frac{1}{2}\vec{T}dS = \frac{1}{2}\vec{T}S \tag{A.1}$$

$$\vec{M}_{s,i} = \int_{\text{panel}} \vec{r}_i \times \frac{1}{2}\vec{T}dS = \frac{1}{2}\vec{r}_{cp,i} \times \vec{T}S, \tag{A.2}$$

where the subscript s indicates that the quantity is on the structural mesh, $i = 1, 2$ indicates the left or right node, S is the panel area, $\vec{r}_i$ is the vector pointing from the structural node to a point on the panel, and $\vec{r}_{cp,i}$ points from the structural node to the aerodynamic center of pressure for the panel. We consider the center of pressure to be at the middle of the panel in the spanwise direction and a quarter-length of the panel from the front in the chordwise direction. With the $\vec{F}_{s,i}$ and $\vec{M}_{s,i}$ obtained using this load transfer scheme, consistency is satisfied by construction because the nodal forces and moments are defined to be the equivalent resultants from the traction field.

For the displacement transfer, our goal is to transfer the computed displacements on the structural mesh, $\vec{u}_{s,i}$, to deflections on the aerodynamic mesh, $\vec{u}_a$. The structural displacements are partitioned into translational displacements, $\vec{d}_{s,i}$, and rotations, $\boldsymbol{\theta}_{s,i}$. The displacement transfer is given by

$$\vec{u}_a = \frac{1}{2}\sum_{i=1}^{2}\left(\vec{d}_{s,i} + \boldsymbol{\theta}_{s,i} \times \vec{r}_i\right), \tag{A.3}$$

where again, $\vec{r}_i$ is a vector pointing from the left or right structural node to the point on the aerodynamic mesh corresponding to $\vec{u}_a$. A constant factor of one-half is used, since we average the contributions from the left and right structural nodes no matter where we are in the panel. We can do this since we evaluate the aerodynamic mesh only at the midpoint in the spanwise direction.

We now verify that our load and displacement transfer scheme is conservative. The virtual work done on the structural mesh by the nodal forces and moments corresponding to a panel is

$$\delta W_s = \sum_{i=1}^{2}\left(\vec{F}_{s,i} \cdot \delta\vec{d}_{s,i} + \vec{M}_{s,i} \cdot \delta\boldsymbol{\theta}_{s,i}\right). \tag{A.4}$$

Inserting $\vec{F}_{s,i} = \frac{1}{2}\vec{T}S$ and $\vec{M}_{s,i} = \frac{1}{2}\vec{r}_{cp,i} \times \vec{T}S$, we obtain

$$\delta W_s = \frac{1}{2}\sum_{i=1}^{2}\left(\vec{T} \cdot \delta\vec{d}_{s,i} + \vec{r}_{cp,i} \times \vec{T} \cdot \delta\boldsymbol{\theta}_{s,i}\right) S. \tag{A.5}$$

The virtual work done on the aerodynamic mesh by the traction is

$$\delta W_a = \int_{\text{panel}} \vec{T} \cdot \delta\vec{u}_a dS. \tag{A.6}$$

Inserting $\vec{u}_a$ from Eq. (A.3), we obtain

$$\delta W_a = \frac{1}{2}\sum_{i=1}^{2}\int\left(\vec{T} \cdot \delta\vec{d}_{s,i} + \vec{T} \cdot \delta\boldsymbol{\theta}_{s,i} \times \vec{r}_i\right) dS. \tag{A.7}$$

Since $\vec{T}$, $\vec{d}_{s,i}$, and $\boldsymbol{\theta}_{s,i}$ are constant over the panel, the integration yields

$$\delta W_a = \frac{1}{2}\sum_{i=1}^{2}\left(\vec{T}\cdot\delta\vec{d}_{s,i} + \vec{T}\cdot\delta\boldsymbol{\theta}_{s,i}\times\vec{r}_{cp,i}\right)S. \tag{A.8}$$

By vector algebra, we have

$$\begin{aligned}\vec{T}\cdot\delta\boldsymbol{\theta}_{s,i}\times\vec{r}_{cp,i} &= \delta\boldsymbol{\theta}_{s,i}\times\vec{r}_{cp,i}\cdot\vec{T}\\ &= \delta\boldsymbol{\theta}_{s,i}\cdot\vec{r}_{cp,i}\times\vec{T} = \vec{r}_{cp,i}\times\vec{T}\cdot\delta\boldsymbol{\theta}_{s,i}.\end{aligned} \tag{A.9}$$

Therefore, from Eqs. (A.5), (A.8), and (A.9), we conclude that $\delta W_a = \delta W_s$, which proves that our load and displacement transfer scheme is conservative.

A.3.3 Aerostructural system

We now couple the aerodynamic and structural systems described above to solve the aerostructural system. We can think of the aerodynamics and structures as two separate groups that receive inputs and produce outputs. The aerodynamics group receives a mesh and outputs aerodynamic loads, whereas the structural group receives aerodynamic loads and outputs structural displacements.

The default setting within OpenAeroStruct uses Gauss–Seidel fixed-point iterations to converge the multidisciplinary analysis (MDA). This means that each analysis is run using the most recent output from the other analysis until a consistent set of state variables is returned. However, it is possible to use Newton's method to converge the coupled aerostructural system: the same partial derivatives used in implementing the adjoint method are used in computing the Newton step at every iteration.

One important feature of OpenMDAO is the ability to subdivide a problem into components that have a small number of inputs and outputs and contain relatively simple analyses. An advantage of this decomposition is that the component-level partial derivatives are simpler to symbolically differentiate if they are not done with AD. We can reorganize the components within the system, and OpenMDAO automatically computes the correct total coupled derivatives using the MAUD architecture [67]. Additionally, we can simply add a few lines of code to obtain derivatives with respect to a new output.

Now that we have presented the underlying theory for the aerodynamic and structural models, we examine how the internal components within OpenAeroStruct pass data; see Fig. A.5. The *prob_vars* component contains information about the airflow around the lifting surfaces. Just below this, we have the *wing* group, which contains geometric information about the lifting surface and structural spar. There is one such group for each lifting surface, with a user-defined name (the default name is *wing*).

We then have a *coupled* group that solves for the aerodynamic and structural state values. Within this *coupled* group, the *aero_states* group assembles the aerodynamic influence coefficient matrix and computes the aerodynamic loads acting on each panel of the lifting surfaces. There is only one *aero_states* group no matter how many lifting surfaces are defined by the user. The *struct_states* group computes the structural displacements based on the aerodynamic loads.

Next, there is a *wing_perf* group, which calculates the lift and drag performance of the lifting surface. Again, there is one *_perf* group for each lifting surface defined by the user. Lastly, the *total_perf* group calculates aerostructural performance metrics for the entire aircraft, such as fuel burn and coefficient of moment.

We now offer some ideas for modifications to the model. Currently the structural mesh and aerodynamic mesh must have the same spanwise discretization, but it is possible to use interpolation to allow different spanwise discretizations. To do so, a consistent and conservative load and displacement transfer method must be used. Additionally, a different FEA model could be used in place of the tubular spar. A past user of OpenAeroStruct implemented a structural model that computes the cross-sectional properties of a wingbox to obtain more realistic wing deflection

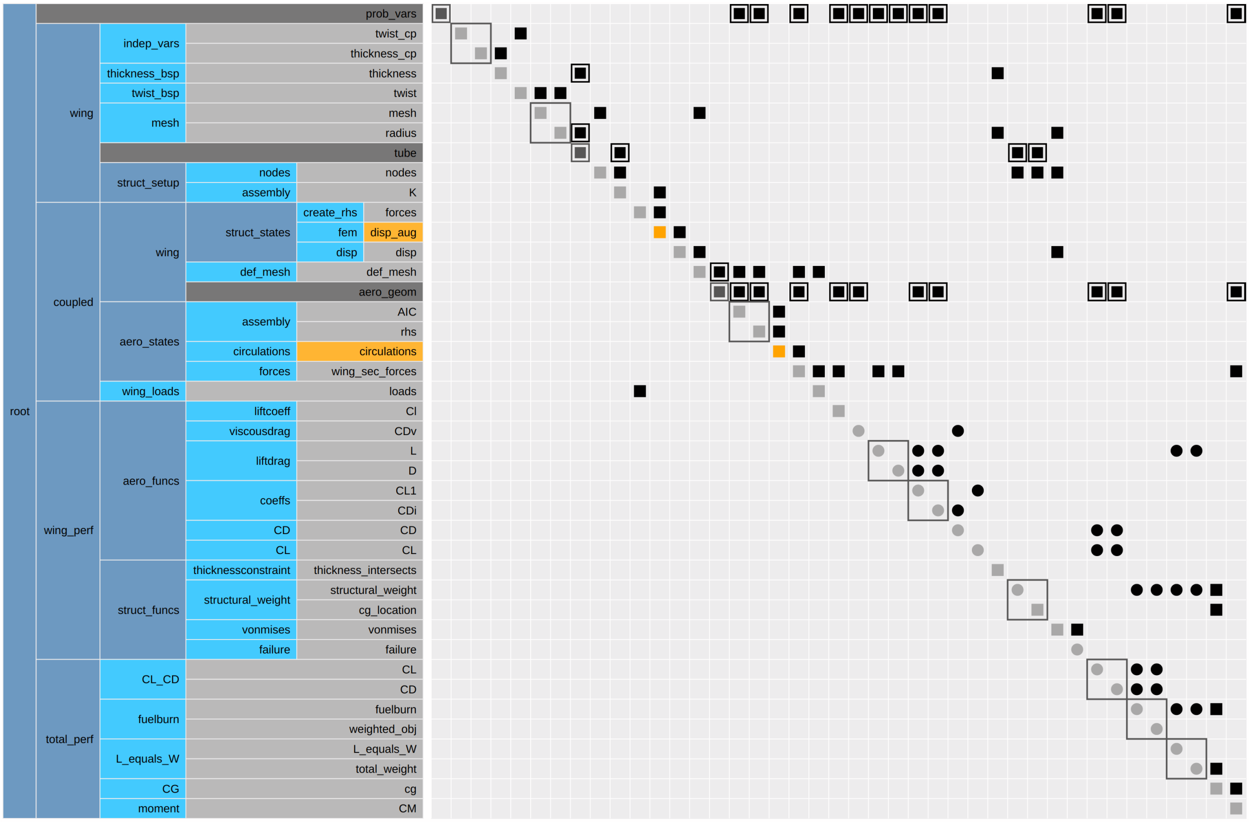


Figure A.5: Component layout for aerostructural analysis and optimization; this hierarchy and data-passing diagram is automatically produced by OpenMDAO for debugging and educational purposes.

properties. Because of the code modularity, the user was able to do this cleanly and quickly by replacing the existing tubular spar with his wingbox model.

A.3.4 Aerostructural optimization

For aerostructural optimization, we can use any combination of the previously discussed design variables. We also have two aerostructural outputs: fuel burn computed via the Breguet range equation, and a constraint ensuring that lift equals weight. The Breguet range equation is given by

$$W_f = (W_0 + W_s)\left[\exp\left(\frac{R \cdot \text{SFC}}{V}\left(\frac{L}{D}\right)^{-1}\right) - 1\right],$$

where W_f is the fuel weight, W_0 is the aircraft empty weight, W_s is the structural weight, R is the range, V is the velocity, and SFC is the specific fuel consumption.

Users can easily add their own objective functions or constraints by creating an OpenMDAO component that computes the desired quantities. Because the total derivatives are computed using the unified chain rule and the adjoint method, the user simply needs to supply partial derivatives for the calculations done in the new component, which makes introducing new outputs a straightforward process. Since the software is open-source, these components can be added to the main repository to expand the capabilities of OpenAeroStruct. Figure A.6 shows the XDSM diagram corresponding to aerostructural optimization. Here, the input and output vectors, x and y, respectively, vary depending on the design variables, objectives, and constraints selected.

A.4 Education

OpenAeroStruct has been used in courses at ISAE-SUPAERO (the University of Toulouse's Institute for Aerospace Engineering) and the University of Michigan. These were graduate-level MDO courses taken by students whose previous exposure to Python ranged from none to using it in

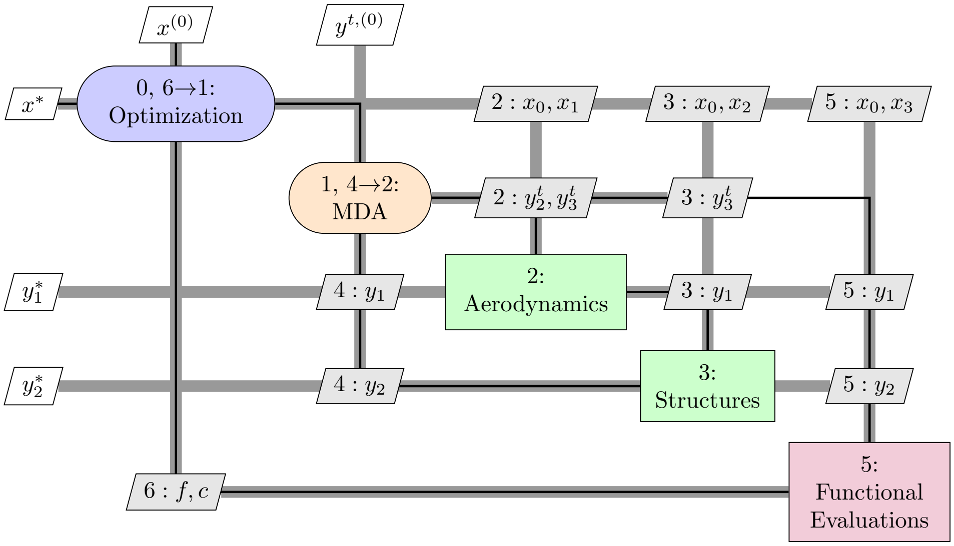


Figure A.6: XDSM diagram [68] for default aerostructural optimization. The x vectors are design variables and the y vectors are states, where * represents the values at the design optimum. The default solver for the MDA is Gauss–Seidel within OpenAeroStruct, shown here.

their daily research. Most of the students had experience only with Matlab or C/C++, but they were able to easily use OpenAeroStruct because of its simple run scripts and detailed documentation. The course prerequisites were not demanding: basic calculus, linear algebra, and programming skills. The students appreciated being able to test theories and hypotheses quickly by running tightly coupled aerostructural optimizations in minutes on their personal computers.

A.4.1 ISAE-SUPAERO

At ISAE-SUPAERO, the students used OpenAeroStruct to learn about structural optimization, MDA, and MDO over the span of a few hours. They worked in small groups and examined the effects of mesh size, algorithmic options, and component grouping on the overall optimization performance. The students were given problem sets instructing them to run the existing OpenAeroStruct model in various configurations and draw conclusions based on the results. To aid in their understanding of the different models and the problem hierarchy, we developed an interactive program to visualize the optimization history and wing model. A sample visualization for an optimized aerostructural model was shown in Fig. A.1. In both years in which OpenAeroStruct was used, the problem sets consisted of a progression from structural optimization to MDA and finally to MDO.

In the structural optimization section, the students interpreted the optimization problem, discussed the optimal thickness distributions, and looked at the effect of increasing the mesh size. The goal here was to have the students work with a simple hands-on numerical optimization problem that they could interpret and evaluate using their knowledge of physics.

In the MDA section, the students ran aerostructural analyses, assigning their own nonlinear and linear solvers to converge the coupled systems. They were instructed to compare fixed-point iteration (nonlinear block Gauss–Seidel), Newton, and a hybrid approach where fixed-point iteration is used as a start-up strategy for the Newton iterations, effectively providing globalization. They were also instructed to look at various linear solvers for Newton's method, including direct methods, linear block Gauss–Seidel, Krylov subspace methods, and Krylov subspace methods with preconditioning.

In the MDO section, the students performed aerostructural optimization. They were instructed to compare the effect of the derivative computation method (finite differences and the adjoint method) on the computation time and the number of optimization iterations. They interpreted the optimized thickness and twist distributions, and they were encouraged to experiment with different solver choices and options to minimize the total optimization time.

A theme throughout the assignments was the emphasis on the benefits of modularity, especially when it comes to gradient computation using the adjoint method. Using OpenMDAO's model structure visualization, shown earlier in Fig. A.5, the students were able to see how the overall OpenAeroStruct model decomposes into geometry, aerodynamics, structures, load and displacement transfer, and performance. In fact, the students were asked to add their own component in the MDO section. The aerostructural optimization was given to them as a fuel-burn minimization problem, but they were instructed to write a new component, one that computes range for a given fuel burn, enabling them to reformulate the fuel-burn minimization for a fixed range into a range maximization for a fixed amount of fuel. They were responsible for deriving and computing the partial derivatives of their new component, but, as they saw first-hand, the modular approach can update the total derivative computation without any additional effort from the user.

A.4.2 University of Michigan

OpenAeroStruct was also used as a basis for the final project in the MDO course at the University of Michigan. The students worked individually to set up an aircraft of their choice within OpenAeroStruct and performed aerodynamic, structural, and aerostructural analysis and optimization. They produced Pareto fronts of the fuel burn and aircraft weight, and they examined how sequential optimization compared to the multidisciplinary design feasible architecture [66, 206]. The students evaluated aircraft ranging from small-scale UAVs to the AN-225 and compared the results from OpenAeroStruct with real-world values. This allowed them to combine their aircraft design knowledge with hands-on experience of optimization methods, leading to a more intuitive understanding of MDO. Previous assignments in this course asked the students to implement optimization methods from scratch. Using OpenAeroStruct allowed them to explore different multidisciplinary solvers and design variables much more rapidly.

Figure A.7 appeared in a student's final report. Because OpenAeroStruct uses OpenMDAO, the students can easily change the nonlinear solver used to converge the coupled aerostructural system to investigate convergence trends and computational costs, as shown in Fig. A.7. Here, the student compares the time to converge the coupled system using different nonlinear solvers for decreasing spar thicknesses, which increases wing flexibility and thus coupling strength. Because of the short wall times, the students can perform many aerostructural optimizations with different formulations to compare relative aircraft performance. We see that NLBGS generally requires less time to solve the coupled system, but it cannot converge the most strongly coupled systems that Newton can

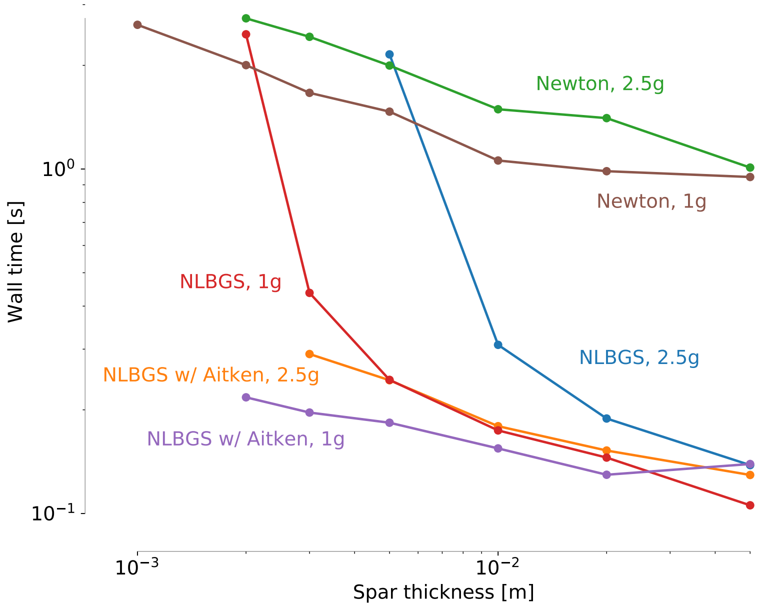


Figure A.7: Comparison of different solvers for the solutions of the coupled aerostructural system for level flight (1 g) and a pull-up maneuver (2.5 g). As spar thickness decreases, the system becomes more strongly coupled, and nonlinear block Gauss–Seidel (NLBGS) without Aitken relaxation cannot converge the coupled system as well as Newton's method [194].

handle. As the spar thickness decreases, the coupling strength decreases and all solvers require less time to converge the system.

For the MDO course project, the first author compared designs produced by sequential optimization and MDO for a small-scale flying-wing unmanned aerial vehicle. Figure A.8 shows that the Pareto front for the MDO optima always dominates the sequential optimization results. For the sequential optimization, we follow the method presented by Chittick and Martins [207]. First, we size the structure based on fixed aerodynamic loads to obtain a structural weight estimate, and then we calculate the optimal twist distribution through an aerodynamic optimization based on the fixed structural weight. We repeat this process until the results converge. We run sequential optimizations at multiple fixed span values to obtain the different points on the Pareto front. The optimizer in the MDO approach can control span, twist, and structural spar thickness.

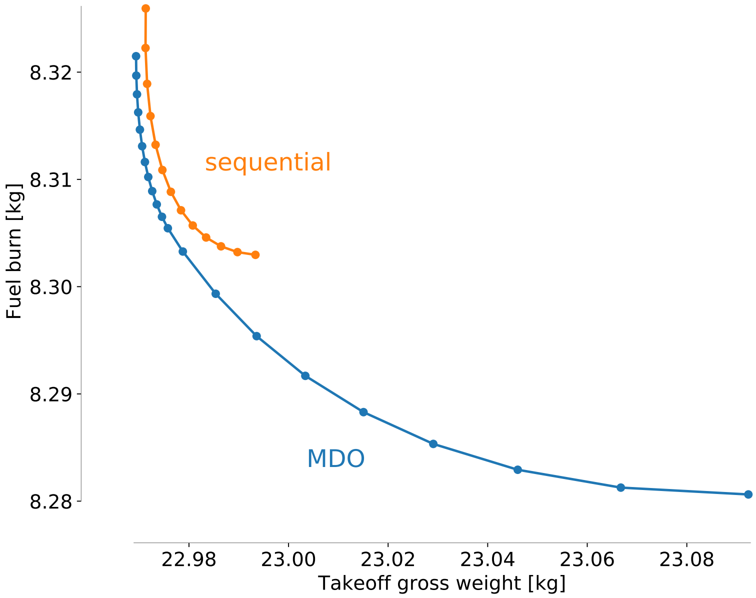


Figure A.8: The MDO-obtained Pareto front always dominates the sequential-optimization Pareto front.

A.5 Research applications

OpenAeroStruct has also proved useful for research and has been used as a realistic testbed application by Bons et al. [193], Cook et al. [191], and Friedman et al. [192]. Because all of the internal analysis and derivative computations are exposed to the user, invasive methods can be implemented, including advanced methods for robust optimization, reliability-based design optimization, and conditional value-at-risk optimization.

Cook et al. [191] used OpenAeroStruct to benchmark a new method for optimization under uncertainty called horsetail matching, which is a flexible approach to optimization under a mix of probabilistic and interval uncertainties. Horsetail matching minimizes the difference between the expected values of a quantity of interest and a desired target. This allows designers to perform robust optimization to try to minimize the chance of failure for the majority of operating cases. Cook et al. used OpenAeroStruct to easily and quickly compare their proposed method with other optimization-under-uncertainty methods using a physical problem representative of larger systems of interest. Compared to a deterministic optimization, their robust optimization resulted in a higher angle of attack and a less aggressive aerodynamic twist for better performance under uncertain conditions.

Friedman et al. [192] used OpenAeroStruct as a test case to quantify the effects of model discrepancy (uncertainty associated with the fact that no model is perfect). Complex multidisciplinary systems often consist of multiple pre-existing physics-based models, which each have their own associated uncertainty. Friedman et al. compared different formulations of model discrepancy in coupled systems and performed a pattern search optimization to minimize the difference between each variable's marginal and conditional distributions. The coupled aerostructural model in OpenAeroStruct was evaluated thousands of times using a Gibbs sampler, which would not be tractable with a more expensive coupled model.

Recently, Chauhan et al. [60, 133] expanded OpenAeroStruct's structural modeling capabilities and used it for studies of electric aircraft trajectories. Hendricks et al. [47] examined tiltwing commuter aircraft mission optimizations using OpenAeroStruct as the aerostructural solver. Jasa et al. [208] optimized multiple aircraft wings of different sizes under a variety of conditions.

As another example of OpenAeroStruct's usefulness in research, we present an aerostructural optimization study that shows some of the fundamental tradeoffs in wing aerostructural design. Table A.1 shows the problem formulation for the optimization problems in this study, where β is a fixed weighting parameter that combines the functions of interest. For each β value, the optimized result is the same regardless of the initial planform.

Figure A.9 shows three optimized wing planform shapes and the corresponding structural thickness, twist, and lift distributions. The leftmost column shows the initial planform for the optimizations and the legend for the plots. The next three columns show the optimization results for different objective functions, where the leftmost is a fuel-burn minimization, the rightmost is a structural spar weight minimization, and the middle one corresponds to an equally weighted optimization. Because fuel burn and spar weight are of the same order and have the same units, we do

Table A.1: Sample aerostructural optimization problem formulation within OpenAeroStruct. FB stands for fuel burn. n_{cp} corresponds to the number of control points for the B-spline interpolation that controls the spanwise distribution of the variables. The numbers of thickness and twist control points do not necessarily need to be the same. The structural failure constraint is aggregated using a Kreisselmeier–Steinhauser (KS) function [139, 140].

	Function/variable	**Description**	**Size**
minimize	$\beta\text{FB} + (1-\beta)W_{\text{struct}}$	objective function	1
w.r.t.	thickness	structural spar thickness	n_{cp}
	twist	aerodynamic twist	n_{cp}
	α	angle of attack	1
	root chord	root chord	1
	taper	taper ratio	1
subject to	$L = W$	lift equals weight	1
	$\text{KS}\left(\sigma_{2.5g}\right) \leq \sigma_{\text{yield}}$	aggregated spar failure	1

not nondimensionalize the weighted objective function.

For the fuel burn minimization, the optimizer increases the span and decreases the root chord to produce a more aerodynamically efficient wing. Additionally, the optimized wing twist is positive for most of the wing, except at the tip. Aerodynamic considerations dominate the design of this specific wing, so the lowest fuel burn results in a lift distribution that is close to elliptical. The optimized thickness is greatest at the root and gradually decreases as we approach the tip, as expected. This optimization results in a lower total aircraft weight than that obtained from the structural spar weight minimization. When minimizing structural spar weight, the optimizer reduces the span and tries to minimize the thickness of the spar throughout. To avoid structural failure, it twists down the outboard section of the wing to unload the tip, as shown in the lift distribution. In this case, the optimizer does not consider the aerodynamic performance, except for its effect on the lift constraint.

We have observed these same trends when using high-fidelity analyses in aerostructural optimization [64], but we can explore the design space much more quickly using OpenAeroStruct. This ability to rapidly explore the wing design space was especially valuable in the work of Bons et al. [193], where OpenAeroStruct enabled the rapid exploration of multimodality in aerodynamic planform optimization, helped explain the physics of this multimodality, and provided promising starting points for much more costly design optimizations based on the solution of the Reynolds-averaged Navier–Stokes equations.

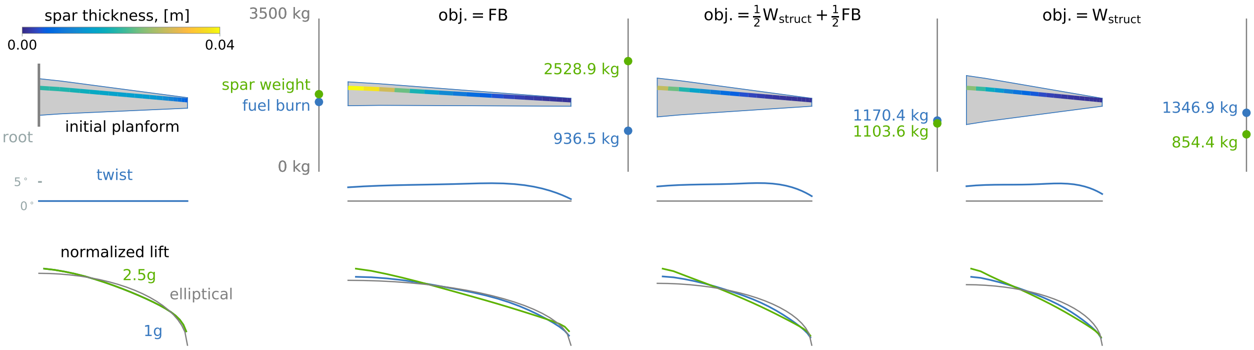


Figure A.9: Optimized planforms, twist, lift, and thickness distributions for three aerostructural optimizations with three different objective functions. The first column shows the initial shape and provides a legend for the plot. The second column is the result from a fuel burn minimization and the rightmost column corresponds to a structural weight minimization. The column between them shows the optimized result for an equally-weighted combined objective function. Each case took under a minute to run on a desktop computer. Multiple different starting points were tested, but each starting point converged to the same optimum for a given objective function, so we only show the results from one starting point here.

A.6 Conclusions

We have presented OpenAeroStruct, an educational open-source low-fidelity aerostructural analysis and optimization tool that uses NASA's OpenMDAO framework. Its modular implementation and efficient derivative computation make it a unique tool for teaching the adjoint method and solution methods for MDA and MDO.

OpenAeroStruct has already proved useful in both educational and research settings. It is straightforward to install and use because it is open-source and well-documented. Students can learn MDO techniques through realistic aircraft design problems while experimenting with optimization formulation and problem size. Additionally, it produces simple MDO problems with variable levels of fidelity that allow state-of-the-art MDA solvers and uncertainty quantification methods to be tested quickly. Because OpenAeroStruct captures some of the same trends as high-fidelity analyses, it can also be used to explore the design space before resorting to more computationally expensive methods for design refinement and better accuracy.

The modular nature of OpenAeroStruct encourages the addition of more features through collaboration. Going forward, we will expand OpenAeroStruct's capabilities for a variety of purposes, including stability-constrained UAV optimization, multifidelity optimization, and trajectory optimization.

Bibliography

[1] Falck, R. D., Chin, J. C., Schnulo, S. L., Burt, J. M., and Gray, J. S., "Trajectory Optimization of Electric Aircraft Subject to Subsystem Thermal Constraints," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, CO, June 2017.

[2] Fleming, P. A., Ning, A., Gebraad, P. M., and Dykes, K., "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, Vol. 19, No. 2, 2016, pp. 329–344. doi:https://doi.org/10.1002/we.1836.

[3] Bessonov, N. and Volpert, V., *Dynamic models of plant growth*, Editions Publibook, 2006.

[4] Kaiser, F., "The climb of jet-propelled aircraft, part I. speed along the path of optimum climb," *Ministry of Supply (Gt. Brit.), RTP/TIB Translation No. Tech. Rep. GDC/15/14ST*, 1944.

[5] Merritt, S. R., Cliff, E. M., and Kelley, H. J., "Energy-modelled climb and climb-dashthe Kaiser technique," *Automatica*, Vol. 21, No. 3, 1985, pp. 319–321.

[6] Bryson, A. E., "Optimal control-1950 to 1985," *IEEE Control Systems Magazine*, Vol. 16, No. 3, 1996, pp. 26–33.

[7] Weston, A. R., Cliff, E. M., and Kelley, H. J., "Altitude transitions in energy climbs," *Automatica*, Vol. 19, No. 2, 1983, pp. 199–202.

[8] Cliff, E. M., "Energy-state models," 1998.

[9] Lush, K., *A Review of the Problem of Choosing a Climb Technique, with Proposals for a New Climb Technique for High Performance Aircraft*, His Maj. Stat. Office, 1951.

[10] Rutowski, E. S., "Energy approach to the general aircraft performance problem," *Journal of the Aeronautical Sciences*, Vol. 21, No. 3, 1954, pp. 187–195.

[11] Bell, D., "A Review of Flight Optimisation Along Synergic Paths in the Period 1945-1960," *The Aeronautical Journal*, Vol. 67, No. 626, 1963, pp. 119–123.

[12] Faulders, C., "Low-thrust rocket steering program for minimum time transfer between planetary orbits," Tech. rep., SAE Technical Paper, 1958.

[13] Jr, A. E. B., “Optimum rocket trajectories with aerodynamic drag,” *Journal of Jet Propulsion*, Vol. 28, No. 7, 1958, pp. 465–469.

[14] Kelley, H. J., “Gradient theory of optimal flight paths,” *Ars Journal*, Vol. 30, No. 10, 1960, pp. 947–954.

[15] Kelley, H. J., “An investigation of optimal zoom climb techniques,” *Journal of the Aerospace Sciences*, Vol. 26, No. 12, 1959, pp. 794–802.

[16] Bryson, A. E. and Denham, W. F., “A steepest-ascent method for solving optimum programming problems,” *Journal of Applied Mechanics*, Vol. 29, No. 2, 1962, pp. 247–257.

[17] Bryson Jr, A. E., Desai, M. N., and Hoffman, W. C., “Energy-state approximation in performance optimization of supersonic aircraft,” *Journal of Aircraft*, Vol. 6, No. 6, 1969, pp. 481–488.

[18] Hale, A. L., Dahl, W., and LISOWSKI, J., “Optimal simultaneous structural and control design of maneuvering flexible spacecraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 1, 1985, pp. 86–93.

[19] Miller, D. F. and Shim, J., “Gradient-based combined structural and control optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 3, 1987, pp. 291–298.

[20] Belvin, W. K. and Park, K., “Structural tailoring and feedback control synthesis-An interdisciplinary approach,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 424–429.

[21] Onoda, J. and Haftka, R. T., “An approach to structure/control simultaneous optimization for large flexible spacecraft,” *AIAA journal*, Vol. 25, No. 8, 1987, pp. 1133–1138.

[22] Rahn, M., Schoettle, U., and Messerschmid, E., “Multidisciplinary design tool for system and mission optimization of launch vehicles,” *6th Symposium on Multidisciplinary Analysis and Optimization*, 1996, p. 4130.

[23] Braun, R., Powell, R., Lepsch, R., Stanley, D., and Kroo, I., “Comparison of two multidisciplinary optimization strategies for launch-vehicle design,” *Journal of Spacecraft and Rockets*, Vol. 32, No. 3, 1995, pp. 404–410.

[24] Padula, S., Korte, J., Dunn, H., and Salas, A., “Multidisciplinary optimization branch experience using iSIGHT software,” 1999.

[25] Bowcutt, K. G., “Multidisciplinary optimization of airbreathing hypersonic vehicles,” *Journal of Propulsion and Power*, Vol. 17, No. 6, 2001, pp. 1184–1190.

[26] Zou, B., Buxi, G. S., and Hansen, M., “Optimal 4-D aircraft trajectories in a contrail-sensitive environment,” *Networks and Spatial Economics*, Vol. 16, No. 1, 2016, pp. 415–446.

[27] Zhao, Y. and Tsiotras, P., "Analysis of energy-optimal aircraft landing operation trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, 2013, pp. 833–845.

[28] Pervier, H., Nalianda, D., Espi, R., Sethi, V., Pilidis, P., Zammit-Mangion, D., Rogero, J.-M., and Entz, R., "Application of genetic algorithm for preliminary trajectory optimization," *SAE International Journal of Aerospace*, Vol. 4, No. 2011-01-2594, 2011, pp. 973–987.

[29] Anderson, R. and Milutinovic, D., "Optimization of taxiway traversal at congested airports," *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010, p. 9212.

[30] Anderson, R. and Milutinović, D., "An approach to optimization of airport taxiway scheduling and traversal under uncertainty," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of aerospace engineering*, Vol. 227, No. 2, 2013, pp. 273–284.

[31] Celis, C., Sethi, V., Zammit-Mangion, D., Singh, R., and Pilidis, P., "Theoretical optimal trajectories for reducing the environmental impact of commercial aircraft operations," *Journal of Aerospace Technology and Management*, Vol. 6, No. 1, 2014, pp. 29–42.

[32] Clive, P. D., Johnson, J. A., Moss, M. J., Zeh, J. M., Birkmire, B. M., and Hodson, D. D., "Advanced Framework for Simulation, Integration and Modeling (AFSIM)(Case Number: 88ABW-2015-2258)," *Proceedings of the International Conference on Scientific Computing (CSC)*, The Steering Committee of The World Congress in Computer Science, Computer , 2015, p. 73.

[33] Reuter, R. A., Iden, S., Snyder, R. D., and Allison, D. L., "An overview of the optimized integrated multidisciplinary systems program," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, p. 0674.

[34] Alyanak, E. J. and Allison, D. L., "Multi-Parameter Performance Evaluation, the Next Step in Conceptual Design Concept Assessment," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 0648.

[35] Allison, D. L. and Kolonay, R. M., "Expanded MDO for Effectiveness Based Design Technologies: EXPEDITE Program Introduction," *2018 Multidisciplinary Analysis and Optimization Conference*, 2018, p. 3419.

[36] Davies, C. C., "Lockheed Martin Overview of the AFRL EXPEDITE Program," *AIAA Scitech 2019 Forum*, 2019, p. 0173.

[37] Clark Jr, D. L., Allison, D. L., Bae, H., and Forster, E. E., "Effectiveness-Based Design of an Aircraft Considering Mission Uncertainties," *Journal of Aircraft*, 2019, pp. 1–12.

[38] Bodie, M., Russell, G., McCarthy, K., Lucas, E., Zumberge, J., and Wolff, M., "Thermal analysis of an integrated aircraft model," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010, p. 288.

[39] Alyanak, E. J. and Allison, D. L., "Fuel Thermal Management System Consideration in Conceptual Design Sizing," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, p. 0670.

[40] Klatt, N. D., "On-board thermal management of waste heat from a high-energy device," Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF , 2008.

[41] DeSimio, M. P., Hencey, B. M., and Parry, A. C., "Online prognostics for fuel thermal management system," *ASME 2015 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2015, pp. V001T08A003–V001T08A003.

[42] Gvozdich, G., Weise, P., and von Spakovsky, M., "INVENT: Study of the Issues Involved in Integrating a Directed Energy Weapons Subsystem into a High Performance Aircraft System," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012, p. 490.

[43] Wolff, M., "INVENT Tip to Tail Energy/Engine/Power/Thermal Modeling, Simulation, & Analysis (MS&A)," *Presentation at 5th Annual Research Consortium for Multidisciplinary System Design Workshop*, 2010.

[44] Hendricks, E. S., Falck, R. D., and Gray, J. S., "Simultaneous Propulsion System and Trajectory Optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, No. AIAA-2017-4435, Denver, CO, June 2017. doi:10.2514/6.2017-4435, AIAA-2017-4435.

[45] Schnulo, S. L., Jeff Chin, R. D. F., Gray, J. S., Papathakis, K. V., Clarke, S. C., Reid, N., and Borer, N. K., "Development of a Multi-Segment Mission Planning Tool for SCEPTOR X-57," *2018 Multidisciplinary Analysis and Optimization Conference*, AIAA, Atlanta, GA, June 2018. doi:10.2514/6.2018-3738.

[46] Falck, R. D. and Gray, J. S., "Optimal control within the context of multidisciplinary design, analysis, and optimization," *AIAA Scitech 2019 Forum*, 2019, p. 0976.

[47] Hendricks, E. S., Falck, R. D., Gray, J. S., Aretskin-Hariton, E. D., Ingraham, D. J., CHapman, J. W., Schnulo, S. L., Chin, J. C., Jasa, J. P., and Bergeson, J. D., "Multidisciplinary Optimization of a Turboelectric Tiltwing Urban Air Mobility Aircraft," *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX, June 2019.

[48] Capristan, F. M. and Welstead, J., "LEAPS: An Initial Assessment Towards a Multi-Order Approach to Air Vehicle Mission Analysis," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4325.

[49] Capristan, F. M. and Welstead, J. R., "An Energy-Based Low-Order Approach for Mission Analysis of Air Vehicles in LEAPS," *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1755.

[50] Welstead, J. R., Caldwell, D., Condotta, R., and Monroe, N., "An Overview of the Layered and Extensible Aircraft Performance System (LEAPS) Development," *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1754.

[51] Liem, R. P., Mader, C. A., Lee, E., and Martins, J. R. R. A., "Aerostructural design optimization of a 100-passenger regional jet with surrogate-based mission analysis," *2013 Aviation Technology, Integration, and Operations Conference*, September 2013. doi:10.2514/6.2013-4372.

[52] Kao, J. Y., Hwang, J. T., Martins, J. R. R. A., Gray, J. S., and Moore, K. T., "A Modular Adjoint Approach to Aircraft Mission Analysis and Optimization," *Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech)*, Kissimmee, FL, January 2015, AIAA 2015-0136.

[53] Kao, J. Y., Hwang, J. T., and Martins, J. R. R. A., "A Modular Approach for Mission Analysis and Optimization," *56th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015.

[54] Liem, R. P., *Multimission Fuel-Burn Minimization in Aircraft Design: A Surrogate-Modeling Approach*, Ph.D. thesis, University of Toronto, 2015.

[55] Hwang, J. T., Jasa, J., and Martins, J. R. R. A., "High-fidelity design-allocation optimization of a commercial aircraft maximizing airline profit," *Journal of Aircraft*, Vol. 56, No. 3, May 2019, pp. 1165–1178. doi:10.2514/1.C035082.

[56] Liem, R. P., Mader, C. A., and Martins, J. R. R. A., "Surrogate Models and Mixtures of Experts in Aerodynamic Performance Prediction for Aircraft Mission Analysis," *Aerospace Science and Technology*, Vol. 43, June 2015, pp. 126–151. doi:10.1016/j.ast.2015.02.019.

[57] Liem, R. P., Kenway, G. K. W., and Martins, J. R. R. A., "Multimission Aircraft Fuel Burn Minimization via Multipoint Aerostructural Optimization," *AIAA Journal*, Vol. 53, No. 1, January 2015, pp. 104–122. doi:10.2514/1.J052940.

[58] Burdette, D. A. and Martins, J. R. R. A., "Impact of Morphing Trailing Edge on Mission Performance for the Common Research Model," *Journal of Aircraft*, Vol. 56, No. 1, January 2019, pp. 369–384. doi:10.2514/1.C034967.

[59] Brelje, B. J. and Martins, J. R. R. A., "Development of a Conceptual Design Model for Aircraft Electric Propulsion with Efficient Gradients," *Proceedings of the AIAA/IEEE Electric Aircraft Technologies Symposium*, Cincinnati, OH, July 2018. doi:10.2514/6.2018-4979.

[60] Chauhan, S. and Martins, J. R. R. A., "Tilt-wing eVTOL takeoff trajectory optimization," *Journal of Aircraft*, 2019. doi:10.2514/1.C035476, (In press).

[61] Marta, A. C., Mader, C. A., Martins, J. R. R. A., van der Weide, E., and Alonso, J. J., "A methodology for the development of discrete adjoint solvers using automatic differentiation tools," *International Journal of Computational Fluid Dynamics*, Vol. 21, No. 9, 2007, pp. 307–327. doi:10.1080/10618560701678647.

[62] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and van der Weide, E., "ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers," *AIAA Journal*, Vol. 46, No. 4, April 2008, pp. 863–873. doi:10.2514/1.29123.

[63] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., "Effective Adjoint Approaches for Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 110, October 2019, pp. 100542. doi:10.1016/j.paerosci.2019.05.002.

[64] Kenway, G. K. W. and Martins, J. R. R. A., "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, Vol. 51, No. 1, January 2014, pp. 144–160. doi:10.2514/1.C032150.

[65] Martins, J. R. R. A. and Hwang, J. T., "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, Vol. 51, No. 11, November 2013, pp. 2582–2599. doi:10.2514/1.J052184.

[66] Martins, J. R. R. A. and Lambe, A. B., "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, Vol. 51, No. 9, September 2013, pp. 2049–2075. doi:10.2514/1.J051895.

[67] Hwang, J. T. and Martins, J. R. R. A., "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives," *ACM Transactions on Mathematical Software*, Vol. 44, No. 4, June 2018, pp. Article 37. doi:10.1145/3182393.

[68] Lambe, A. B. and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, August 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[69] Sobieszczanski-Sobieski, J., "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results," *Journal of Aircraft*, Vol. 27, No. 12, Dec. 1990, pp. 993–1001. doi:10.2514/3.45973.

[70] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, May 2014, pp. 935–951. doi:10.2514/1.J052255.

[71] Hargraves, C. R. and Paris, S. W., "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[72] Betts, J. T. and Cramer, E. J., "Application of direct transcription to commercial aircraft trajectory optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 151–159.

[73] Herman, A. L. and Conway, B. A., "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[74] Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[75] Allison, D. L., Alyanak, E. J., and Shimmin, K., "Aircraft System Effects Including Propulsion and Air Cycle Machine Coupled Interactions," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, p. 0671. doi:10.2514/6.2016-0671.

[76] Perullo, C. A., Trawick, D., Clifton, W., Tai, J., and Mavris, D. N., "Development of a suite of hybrid electric propulsion modeling elements using NPSS," *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, American Society of Mechanical Engineers Digital Collection, 2014.

[77] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, April 2019, pp. 1075–1104. doi:10.1007/s00158-019-02211-z.

[78] Stanley, A. P. J., Ning, A., and Dykes, K., "Coupled Wind Turbine Design and Layout Optimization with Non-Homogeneous Wind Turbines," *Wind Energy Science*, July 2018. doi:10.5194/wes-2018-54.

[79] Thomas, J., Gebraad, P., , and Ning, A., "Improving the FLORIS Wind Plant Model for Compatibility with Gradient-Based Optimization," *Wind Engineering*, Vol. 41, No. 5, August 2017, pp. 313–329. doi:10.1177/0309524X17722000.

[80] Barrett, R. and Ning, A., "Integrated Free-Form Method for Aerostructural Optimization of Wind Turbine Blades," *Wind Energy*, Vol. 21, No. 8, August 2018, pp. 663–675. doi:10.1002/we.2186.

[81] Gray, J. S., Mader, C. A., Kenway, G. K. W., and Martins, J. R. R. A., "Modeling Boundary Layer Ingestion Using a Coupled Aeropropulsive Analysis," *Journal of Aircraft*, Vol. 55, No. 3, May 2018, pp. 1191–1199. doi:10.2514/1.C034601.

[82] Gray, J. S. and Martins, J. R. R. A., "Coupled Aeropropulsive Design Optimization of a Boundary-Layer Ingestion Propulsor," *The Aeronautical Journal*, Vol. 123, No. 1259, January 2019, pp. 121–137. doi:10.1017/aer.2018.120.

[83] Gray, J., *Design Optimization of a Boundary Layer Ingestion Propulsor Using a Coupled Aeropropulsive Model*, Ph.D. thesis, University of Michigan, 2018.

[84] Yildirim, A., Gray, J. S., Mader, C. A., and Martins, J. R. R. A., "Aeropropulsive Design Optimization of a Boundary Layer Ingestion System," *AIAA Aviation Forum*, Dallas, TX, June 2019. doi:10.2514/6.2019-3455.

[85] Jasa, J. P., Gray, J. S., Seidel, J. A., Mader, C. A., and Martins, J. R. R. A., "Multipoint Variable Cycle Engine Design Using Gradient-based Optimization," *57th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, 2019*, San Diego, CA, January 2019. doi:10.2514/6.2019-0172.

[86] Gray, J. S., Chin, J., Hearn, T., Hendricks, E., Lavelle, T., and Martins, J. R. R. A., "Chemical Equilibrium Analysis with Adjoint Derivatives for Propulsion Cycle Analysis," *Journal of Propulsion and Power*, Vol. 33, No. 5, September 2017, pp. 1041–1052. doi:10.2514/1.B36215.

[87] Hendricks, E. S. and Gray, J. S., "pyCycle: A Tool for Efficient Optimization of Gas Turbine Engine Cycles," *Aerospace*, Vol. 6, No. 87, August 2019. doi:10.3390/aerospace6080087.

[88] Hearn, T. A., Hendricks, E., Chin, J., and Gray, J. S., "Optimization of Turbine Engine Cycle Analysis with Analytic Derivatives," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, June 2016. doi:10.2514/6.2016-4297.

[89] Jasa, J. P., Mader, C. A., and Martins, J. R. R. A., "Trajectory Optimization of a Supersonic Air Vehicle with Thermal Fuel Management System," *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, June 2018. doi:10.2514/6.2018-3884.

[90] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A., "Design and Trajectory Optimization of a Morphing Wing Aircraft," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; AIAA SciTech Forum*, Orlando, FL, January 2018. doi:10.2514/6.2018-1382.

[91] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, September 2003, pp. 245–262. doi:10.1145/838250.838251.

[92] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., "pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, January 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.

[93] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:10.1137/S1052623499350013.

[94] Hwang, J. T. and Martins, J. R. R. A., “A fast-prediction surrogate model for large datasets,” *Aerospace Science and Technology*, Vol. 75, April 2018, pp. 74–87. doi:10.1016/j.ast.2017.12.030.

[95] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A., “A Python surrogate modeling framework with derivatives,” *Advances in Engineering Software*, Vol. 135, September 2019, pp. 102662. doi:10.1016/j.advengsoft.2019.03.005.

[96] Brooks, T. R., Kenway, G. K. W., and Martins, J. R. R. A., “Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings,” *AIAA Journal*, Vol. 56, No. 7, July 2018, pp. 2840–2855. doi:10.2514/1.J056603.

[97] Vassberg, J. C., DeHaan, M. A., Rivers, M. S., and Wahls, R. A., “Retrospective on the Common Research Model for Computational Fluid Dynamics Validation Studies,” *Journal of Aircraft*, Vol. 55, No. 4, 2018, pp. 1325–1337. doi:10.2514/1.C034906.

[98] Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D. W., Brodersen, O. P., Eisfeld, B., Crippa, S., Wahls, R. A., Morrison, J. H., Mavriplis, D. J., and Murayama, M., “Summary of the Fourth AIAA Computational Fluid Dynamics Drag Prediction Workshop,” *Journal of Aircraft*, Vol. 51, No. 4, jul 2014, pp. 1070–1089. doi:10.2514/1.c032418.

[99] Allison, D., Morris, C., Schetz, J., Kapania, R., Sultan, C., Deaton, J., and Grandhi, R., “A multidisciplinary design optimization framework for design studies of an efficient supersonic air vehicle,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5492. doi:10.2514/6.2012-5492.

[100] Burton, S., Alyanak, E., and Kolonay, R., “Efficient supersonic air vehicle analysis and optimization implementation using SORCER,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5520. doi:10.2514/6.2012-5520.

[101] Alyanak, E. and Kolonay, R., “Efficient supersonic air vehicle structural modeling for conceptual design,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5519. doi:10.2514/6.2012-5519.

[102] Davies, C., Stelmack, M., Zink, P. S., De La Garza, A., and Flick, P., “High fidelity MDO process development and application to fighter strike conceptual design,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012, p. 5490. doi:10.2514/6.2012-5490.

[103] Hwang, J. T. and Martins, J. R. R. A., “Allocation-mission-design optimization of next-generation aircraft using a parallel computational framework,” *57th AIAA/ASCE/AHS/ASC*

Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, January 2016. doi:10.2514/6.2016-1662.

[104] Molinari, G., Quack, M., Dmitriev, V., Morari, M., Jenny, P., and Ermanni, P., "Aero-Structural Optimization of Morphing Airfoils for Adaptive Wings," *Journal of Intelligent Material Systems and Structures*, Vol. 22, No. 10, 2011, pp. 1075–1089. doi:10.1177/1045389X11414089.

[105] Kota, S., Osborn, R., Ervin, G., Maric, D., Flick, P., and Paul, D., "Mission Adaptive Compliant Wing–Design, Fabrication and Flight Test," *RTO Applied Vehicle Technology Panel (AVT) Symposium*, 2009.

[106] Waschl, H., Kolmanovsky, I., Steinbuch, M., and Del Re, L., *Optimization and optimal control in automotive systems*, Springer, 2014.

[107] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[108] Lee, E. B. and Markus, L., "Foundations of optimal control theory," Tech. rep., Minnesota Univ Minneapolis Center For Control Sciences, 1967.

[109] Conway, B. A., *Spacecraft trajectory optimization*, Vol. 29, Cambridge University Press, 2010.

[110] Hwang, J. T., Lee, D. Y., Cutler, J. W., and Martins, J. R. R. A., "Large-Scale Multidisciplinary Optimization of a Small Satellite's Design and Operation," *Journal of Spacecraft and Rockets*, Vol. 51, No. 5, September 2014, pp. 1648–1663. doi:10.2514/1.A32751.

[111] Falck, R. and Dankanich, J., "Optimization of low-thrust spiral trajectories by collocation," *AIAA/AAS Astrodynamics Specialist Conference*, 2012, p. 4423.

[112] Ross, I., D'Souza, C., Fahroo, F., and Ross, J., "A fast approach to multi-stage launch vehicle trajectory optimization," *aiaa Guidance, Navigation, and control conference and Exhibit*, 2003, p. 5639.

[113] Brelje, B. and Martins, J. R. R. A., "Electric, Hybrid, and Turboelectric Fixed-Wing Aircraft: A Review of Concepts, Models, and Design Approaches," *Progress in Aerospace Sciences*, Vol. 104, January 2019, pp. 1–19. doi:10.1016/j.paerosci.2018.06.004.

[114] Burdette, D. A. and Martins, J. R. R. A., "Design of a Transonic Wing with an Adaptive Morphing Trailing Edge via Aerostructural Optimization," *Aerospace Science and Technology*, Vol. 81, October 2018, pp. 192–203. doi:10.1016/j.ast.2018.08.004.

[115] Molinari, G., Quack, M., Arrieta, A. F., Morari, M., and Ermanni, P., "Design, realization and structural testing of a compliant adaptable wing," *Smart Materials and Structures*, Vol. 24, 2015, pp. 105027. doi:10.1088/0964-1726/24/10/105027.

[116] Kota, S. and Martins, J. R. R. A., "FlexFoil Shape Adaptive Control Surfaces—Flight Test and Numerical Results," *Proceedings of the 31st Congress of the International Council of Aeronautical Sciences*, Belo Horizonte, Brazil, September 2018.

[117] Xu, J. and Kroo, I., "Aircraft Design with Active Load Alleviation and Natural Laminar Flow," *Journal of Aircraft*, Vol. 51, No. 5, September 2014, pp. 1532–1545. doi:10.2514/1.C032402.

[118] Haghighat, S., Liu, H. H. T., and Martins, J. R. R. A., "A Model-Predictive Gust Load Alleviation Controller for a Highly Flexible Aircraft," *Journal of Guidance, Control and Dynamics*, Vol. 35, No. 6, November 2012, pp. 1751–1766. doi:10.2514/1.57013.

[119] Xu, J., Andrew Ning, S., Bower, G., and Kroo, I., "Aircraft route optimization for formation flight," *Journal of Aircraft*, Vol. 51, No. 2, 2014, pp. 490–501.

[120] Ning, S. A., *Aircraft drag reduction through extended formation flight*, Stanford University, 2011.

[121] Prete, J. and Mitchell, J., "Safe routing of multiple aircraft flows in the presence of time-varying weather data," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 4791.

[122] Clarke, J.-P. B., Ho, N. T., Ren, L., Brown, J. A., Elmer, K. R., Zou, K., Hunting, C., McGregor, D. L., Shivashankara, B. N., Tong, K.-O., et al., "Continuous descent approach: Design and flight test for Louisville International Airport," *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1054–1066.

[123] Thipphavong, D. P., Apaza, R., Barmore, B., Battiste, V., Burian, B., Dao, Q., Feary, M., Go, S., Goodrich, K. H., Homola, J., et al., "Urban air mobility airspace integration concepts and considerations," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3676.

[124] Bosson, C. and Lauderdale, T. A., "Simulation evaluations of an autonomous urban air mobility network management and separation service," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3365.

[125] Vascik, P. D., Hansman, R. J., and Dunn, N. S., "Analysis of urban air mobility operational constraints," *Journal of Air Transportation*, Vol. 26, No. 4, 2018, pp. 133–146.

[126] Seywald, H., "Trajectory optimization based on differential inclusion (Revised)," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487. doi:https://arc.aiaa.org/doi/10.2514/3.21224.

[127] Hwang, J. T. and Munster, D. W., "Solution of ordinary differential equations in gradient-based multidisciplinary design optimization," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, AIAA, Kissimmee, FL, January 2018. doi:10.2514/6.2018-1646.

[128] Ross, I. M. and Fahroo, F., "Legendre pseudospectral approximations of optimal control problems," *New trends in nonlinear dynamics and control and their applications*, Springer, 2003, pp. 327–342.

[129] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851.

[130] Falck, R. D., Gray, J. S., and Naylor, B., "Parallel aircraft trajectory optimization with analytic derivatives," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 3207.

[131] Falck, R. D., Ingraham, D., and Aretskin-Hariton, E., *Multidisciplinary Optimization of Urban-Air-Mobility Class Aircraft Trajectories with Acoustic Constraints*. doi:10.2514/6.2018-4985.

[132] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A., "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 4, April 2018, pp. 1815–1827. doi:10.1007/s00158-018-1912-8.

[133] Chauhan, S. S. and Martins, J. R. R. A., "Low-Fidelity Aerostructural Optimization of Aircraft Wings with a Simplified Wingbox Model Using OpenAeroStruct," *Proceedings of the 6th International Conference on Engineering Optimization, EngOpt 2018*, Springer, Lisbon, Portugal, September 2018, pp. 418–431. doi:10.1007/978-3-319-97773-7_38.

[134] Reich, G. and Sanders, B., "Introduction to morphing aircraft research," *Journal of Aircraft*, Vol. 44, No. 4, 2007, pp. 1059–1059.

[135] Barbarino, S., Bilgen, O., Ajaj, R. M., Friswell, M. I., and Inman, D. J., "A review of morphing aircraft," *Journal of intelligent material systems and structures*, Vol. 22, No. 9, 2011, pp. 823–877.

[136] Sofla, A. Y. N., Meguid, S. A., Tan, K. T., and Yeo, W. K., "Shape Morphing of Aircraft Wing: Status and Challenges," *Materials & Design*, Vol. 31, No. 3, 3 2010, pp. 1284–1292. doi:10.1016/j.matdes.2009.09.011.

[137] Vos, R., Gürdal, Z., and Abdalla, M., "Mechanism for warp-controlled twist of a morphing wing," *Journal of Aircraft*, Vol. 47, No. 2, 2010, pp. 450.

[138] Claus, R. W., Evans, A. L., and Follen, G. J., "Multidisciplinary propulsion simulation using NPSS," *AIAA Paper*, Vol. 92, 1992, pp. 4709.

[139] Kreisselmeier, G. and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems, Zurich, Switzerland*, 1979. doi:10.1016/S1474-6670(17)65584-8.

[140] Lambe, A. B., Martins, J. R. R. A., and Kennedy, G. J., "An Evaluation of Constraint Aggregation Strategies for Wing Box Mass Minimization," *Structural and Multidisciplinary Optimization*, Vol. 55, No. 1, January 2017, pp. 257–277. doi:10.1007/s00158-016-1495-1.

[141] van der Weide, E., Kalitzin, G., Schluter, J., and Alonso, J. J., "Unsteady Turbomachinery Computations Using Massively Parallel Platforms," *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2006. doi:10.2514/6.2006-421, AIAA 2006-0421.

[142] Kenway, G. K. W., Secco, N., Martins, J. R. R. A., Mishra, A., and Duraisamy, K., "An Efficient Parallel Overset Method for Aerodynamic Shape Optimization," *Proceedings of the 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, TX, January 2017. doi:10.2514/6.2017-0357.

[143] Secco, N. R., Jasa, J. P., Kenway, G. K. W., and Martins, J. R. R. A., "Component-based Geometry Manipulation for Aerodynamic Shape Optimization with Overset Meshes," *AIAA Journal*, Vol. 56, No. 9, September 2018, pp. 3667–3679. doi:10.2514/1.J056550.

[144] Coalson, M. S., "Development of the F110-GE-100 engine," *ASME 1984 International Gas Turbine Conference and Exhibit*, American Society of Mechanical Engineers, 1984, pp. V002T02A002–V002T02A002.

[145] Simmons, R. J., *Design and control of a variable geometry turbofan with and independently modulated third stream*, Ph.D. thesis, The Ohio State University, 2009.

[146] Gray, J. S., Hearn, T. A., and Naylor, B. A., "Using Graph Coloring To Compute Total Derivatives More Efficiently in OpenMDAO," *AIAA Aviation 2019 Forum*, 2019, p. 3108. doi:https://doi.org/10.2514/6.2019-3108.

[147] Figliola, R., Tipton, R., and Li, H., "Exergy approach to decision-based design of integrated aircraft thermal systems," *Journal of aircraft*, Vol. 40, No. 1, 2003, pp. 49–55.

[148] Gertler, J., "F-35 Joint Strike Fighter (JSF) Program," Library of Congress Washington DC Congressional Research Service, 2012.

[149] "Airborne Laser System (ABL) YAL 1A," 2011.

[150] Gunzinger, M. and Dougherty, C., *Changing the game: The promise of directed-energy weapons*, Center for Strategic and Budgetary Assessments, 2012.

[151] Huang, H., Spadaccini, L. J., and Sobel, D. R., "Fuel-cooled thermal management for advanced aero engines," *ASME Turbo Expo 2002: Power for Land, Sea, and Air*, American Society of Mechanical Engineers, 2002, pp. 367–376.

[152] Mahulikar, S. P., Sonawane, H. R., and Rao, G. A., "Infrared signature studies of aerospace vehicles," *Progress in Aerospace Sciences*, Vol. 43, No. 7-8, 2007, pp. 218–245.

[153] Doman, D. B., “Fuel flow control for extending aircraft thermal endurance part I: Underlying principles,” *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 1621.

[154] Puterbaugh, R. L., Brown, J., and Battelle, R., “Impact of Heat Exchanger Location on Engine Performance,” *SAE Technical Paper*, SAE International, 10 2012. doi:10.4271/2012-01-2168.

[155] Cai, H., Su, L., Liao, Y., and Weng, Z., “Numerical and experimental study on the influence of top bypass flow on the performance of plate fin heat exchanger,” *Applied Thermal Engineering*, Vol. 146, 2019, pp. 356–363.

[156] Kim, S., Min, J. K., Ha, M. Y., and Son, C., “Investigation of high-speed bypass effect on the performance of the surface air–oil heat exchanger for an aero engine,” *International Journal of Heat and Mass Transfer*, Vol. 77, 2014, pp. 321–334.

[157] Clark, S. F., “787 Propulsion System,” 2012.

[158] Williams, M. A., *A framework for the control of electro-thermal aircraft power systems*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2017.

[159] Carlson, L. A. and Horn, W., “New thermal and trajectory model for high-altitude balloons,” *Journal of Aircraft*, Vol. 20, No. 6, 1983, pp. 500–507.

[160] Kays, W. and London, A., *Compact Heat Exchangers, Third Edition*, McGraw-Hill Book Company, 1984.

[161] Manglik, R. M. and Bergles, A. E., “Heat Transfer and Pressure Drop Correlations for the Rectangular Offset Strip Fin Compact Heat Exchanger,” *Experimental Thermal and Fluid Science*, 1995, pp. 171–180. doi:10.1016/0894-1777(94)00096-Q.

[162] Bruno, T. J., “Thermodynamic, transport and chemical properties of reference JP-8,” Tech. rep., NATIONAL INST OF STANDARDS AND TECHNOLOGY BOULDER CO PHYSICAL AND CHEMICAL , 2006.

[163] Curlett, B. P. and Felder, J. L., “Object-oriented approach for gas turbine engine simulation,” 1995.

[164] Janeway, R., “Quantitative analysis of heat transfer in engines,” *SAE Transactions*, 1938, pp. 371–380.

[165] English, R. E. and Hauser, C. H., “A Method of Cycle Analysis for Aircraft Gas-turbine Power Plants Driving Propellers,” 1948.

[166] Hall, E. J., Lynn, S. R., Heidegger, N. J., and Delaney, R. A., “Energy Efficient Engine Low Pressure Subsystem Flow Analysis,” *NASA Technical Report*, 1998.

[167] Kumar, V., Panda, P., Mongia, H., and Naik, S., "Innovative Approaches for Reducing CO2 Emissions of Aviation Engines Part 2: NPSS calibration with existing gas turbine engines," *10th International Energy Conversion Engineering Conference*, 2012, p. 4228.

[168] CLAUS, R., EVANS, A., and FOLLEN, G., "Multidisciplinary propulsion simulation using NPSS," *4th Symposium on Multidisciplinary Analysis and Optimization*, 1992, p. 4709.

[169] Gronstedt, T. and Wallin, M., "A comparative study of genetic algorithms and gradient methods for RM12 turbofan engine diagnostics and performance estimation," *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, American Society of Mechanical Engineers, 2004, pp. 615–624.

[170] Heath, C., Seidel, J., and Rallabhandi, S. K., "Viscous Aerodynamic Shape Optimization with Installed Propulsion Effects," *35th AIAA Applied Aerodynamics Conference*, 2017, p. 3046. doi:10.2514/6.2017-3046.

[171] Hendricks, E. S., *A multi-level multi-design point approach for gas turbine cycle and turbine conceptual design*, Ph.D. thesis, Georgia Institute of Technology, 2017.

[172] Gray, J. S., Kenway, G. K. W., Mader, C. A., and Martins, J. R. R. A., "Aero-propulsive Design Optimization of a Turboelectric Boundary Layer Ingestion Propulsion System," *2018 AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, June 2018. doi:10.2514/6.2018-3976, AIAA 2018-3976.

[173] Hearn, T., Hendricks, E., Chin, J., Gray, J. S., and Moore, K. T., "Optimization of Turbine Engine Cycle Analysis with Analytic Derivatives," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, part of AIAA Aviation 2016 (Washington, DC)*, 2016. doi:10.2514/6.2016-4297.

[174] DAngelo, M. M., Gallman, J., Johnson, V., Garcia, E., Tai, J., and Young, R., "N+ 3 small commercial efficient and quiet transportation for year 2030-2035," 2010.

[175] Jones, S. M., Haller, W. J., and Tong, M. T.-H., "An N+ 3 Technology Level Reference Propulsion System," 2017.

[176] Rallabhandi, S. K. and Mavris, D. N., "Simultaneous airframe and propulsion cycle optimization for supersonic aircraft design," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 38–55. doi:10.2514/1.33183.

[177] Muller, G. L. and Gasko, W. F., "Subsonic-transonic-drag of supersonic inlets." *Journal of Aircraft*, Vol. 4, No. 3, 1967, pp. 231–236.

[178] Slater, J., "Design and analysis tool for external-compression supersonic inlets," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2012, p. 16.

[179] Kowalski, E. J., "Computer code for estimating installed performance of aircraft gas turbine engines. Volume 1," 1979.

[180] Berton, J. J., Haller, W. J., Senick, P. F., Jones, S. M., and Seidel, J. A., "A comparative propulsion system analysis for the high-speed civil transport," 2005.

[181] Johnson, J., "Variable cycle engines-The next step in propulsion evolution," *12th Propulsion Conference*, 1976, p. 758.

[182] Willis, E., "Variable-cycle engines for supersonic cruise aircraft," 1976.

[183] Johnson, J., "Variable cycle engine developments at General Electric 1955-1995," *Developments In High-Speed Vehicle Propulsion Systems*, 1995, pp. 105–158.

[184] Kurzke, J., "The mission defines the cycle: turbojet, turbofan and variable cycle engines for high speed propulsion," Tech. rep., DACHAU (GERMANY), 2010.

[185] Mattingly, J. D., *Elements of propulsion: gas turbines and rockets*, American Institute of Aeronautics and Astronautics, 2006.

[186] Cumpsty, N. and Heyes, A., *Jet propulsion*, Cambridge University Press, 2015.

[187] Lyu, Z., Xu, Z., and Martins, J. R. R. A., "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization," *Proceedings of the 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, July 2014, ICCFD8-2014-0203.

[188] Jasa, J. P., Brelje, B. J., Mader, C. A., and Martins, J. R. R. A., "Coupled Design of a Supersonic Engine and Thermal System," *World Congress of Structural and Multidisciplinary Optimization*, Beijing, China, May 2019.

[189] Brelje, B. J., Jasa, J. P., Gray, J. S., and Martins, J. R. R. A., "Development of a Conceptual-Level Thermal Management System Design Capability in OpenConcept," *NATO Applied Vehicle Technology Panel*, Trondheim, Norway, October 2019.

[190] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design," *Optimization and Engineering*, Vol. 6, No. 1, March 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62.

[191] Cook, L. W., Jarrett, J. P., and Willcox, K. E., "Horsetail Matching for Optimization Under Probabilistic, Interval and Mixed Uncertainties," *19th AIAA Non-Deterministic Approaches Conference*, 2017, p. 0590.

[192] Friedman, S., Ghoreishi, S. F., and Allaire, D. L., "Quantifying the Impact of Different Model Discrepancy Formulations in Coupled Multidisciplinary Systems," *19th AIAA Non-Deterministic Approaches Conference*, 2017, p. 1950.

[193] Bons, N. P., He, X., Mader, C. A., and Martins, J. R. R. A., "Multimodality in Aerodynamic Wing Design Optimization," *AIAA Journal*, Vol. 57, No. 3, March 2019, pp. 1004–1018. doi:10.2514/1.J057294.

[194] Chauhan, S. S., Hwang, J. T., and Martins, J. R. R. A., "Benchmarking approaches for the multidisciplinary analysis of complex systems using a Taylor series-based scalable problem," *12th World Congress on Structural and Multidisciplinary Optimization*, Braunschweig, Germany, June 2017. doi:10.1007/978-3-319-67988-4_7.

[195] Haftka, R. T., "Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints," *AIAA Journal*, Vol. 15, No. 8, 1977, pp. 1101–1106. doi:10.2514/3.7400.

[196] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds Number Aerodynamics*, Springer, 1989, pp. 1–12.

[197] Melin, T., "A vortex lattice MATLAB implementation for linear aerodynamic wing applications," *Master's Thesis, Department of Aeronautics, Royal Institute of Technology (KTH), Stockholm, Sweden*, 2000.

[198] Drela, M. and Youngren, H., "Athena vortex lattice," *Software Package, Ver*, Vol. 3, 2004.

[199] Gavin, H. P., "Frame3DD Structural Analysis Code, Duke University," .

[200] Dhondt, G. and Wittig, K., "Calculix: A free software three-dimensional structural finite Element Program," *MTU Aero Engines GmbH, Munich*, 1998.

[201] Kennedy, G. J. and Martins, J. R. R. A., "A parallel aerostructural optimization framework for aircraft design studies," *Structural and Multidisciplinary Optimization*, Vol. 50, No. 6, December 2014, pp. 1079–1101. doi:10.1007/s00158-014-1108-9.

[202] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw–Hill, 1991.

[203] Phillips, W. and Snyder, D., "Modern adaptation of Prandtl's classic lifting-line theory," *Journal of Aircraft*, Vol. 37, No. 4, 2000, pp. 662–670.

[204] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA, 5th ed., 2012.

[205] Pesare, G., *Modeling of Unsteady Vortex Lattice Method for wing flutter control in OpenMDAO*, Master's thesis, Sapienza - Università di Roma, 2016.

[206] Cramer, E. J., Dennis, J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[207] Chittick, I. R. and Martins, J. R. R. A., "Aero-Structural Optimization Using Adjoint Coupled Post-Optimality Sensitivities," *Structures and Multidisciplinary Optimization*, Vol. 36, 2008, pp. 59–70. doi:10.1007/s00158-007-0200-9.

[208] Jasa, J. P., Chauhan, S. S., Gray, J. S., and Martins, J. R. R. A., "How Certain Physical Considerations Impact Aerostructural Wing Optimization," *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Dallas, TX, June 2019. doi:10.2514/6.2019-3242.