MISSION-SPECIFIC DESIGN OF AIRCRAFT ENERGY SYSTEMS

BY

CHRISTOPHER THOMAS AKSLAND

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2023

Urbana, Illinois

Doctoral Committee:

Professor Prashant Mehta, Chair
Professor Andrew Alleyne, Director of Research
Professor Carolyn Beck
Professor Kai James
Dr. Daniel Clark Jr., Air Force Research Laboratory

Abstract

The promise of electrified aircraft aims to revolutionize the transportation industry as a more sustainable and capable mode of modern travel. However, the current power and energy density of these systems is lacking, resulting in aircraft with significantly limited range in comparison to their traditional fuel-powered counterparts. This limitation is partially an artifact of the complex aircraft energy systems that are comprised of multi-domain and multi-timescale dynamics with strict operational limitations. Additionally, accounting for such constraints and complex dynamic interactions can be challenging when also considering the wide range of potential applications for electrified aircraft.

To address these challenges and improve power and energy density, one approach is more intelligent system operation; make optimized control decisions that more effectively manage and utilize onboard energy resources. This dissertation proposes a multi-timescale hierarchical model predictive control approach for the operation of the multi-domain aircraft energy systems that inherently accounts for system coupling and operational constraints in the control decision-making process. A key element of the control design is the creation and integration of an optimization-oriented graph-based modeling framework, designed specifically for conservation systems. This graph modeling framework facilitates computationally efficient control design by decomposing the optimal control problem into a hierarchy of coordinated sub-problems. The hierarchical control approach is experimentally validated on an experimental testbed that represents an aircraft integrated power, propulsion, and thermal management system. In comparison to a conventional control approach, the hierarchical controller demonstrated advancements in performance and efficiency metrics with significant improvements in operational reliability. While this demonstration established that control alone can improve a system's performance, it is also necessary to consider the additional flexibility and effectiveness gained from plant design.

Combined plant and control design (co-design) offers an approach to optimal system-level aircraft design by considering architecture, sizing, and control design disciplines. Together, these design methodologies comprise the components/technologies integrated into the aircraft design, the connections between those components to build the system, and their operation. This dissertation focuses on integrating feedback control design within architecture and sizing optimization processes. The design methods leverage the graph-based modeling approach to efficiently encode the plant dynamics into the optimization routine. Modular graph models play a key role in the architecture optimization process because they enable the evaluation of a variety of system topologies. A scalable combined plant sizing and feedback control optimization is applied to a hybrid UAV design, demonstrating significant improvements in closed-loop system performance in contrast to other conventional design strategies. Additionally, a novel and efficient, relaxed approach to the architecture optimization problem was used to create mission-specific plant and feedback control designs for a thermal management system. The result identified a system topology that is optimized for a range of operating requirements. Notably, the proposed approach could identify near optimal system designs an order of magnitude faster than traditional exhaustive search approaches.

To my friends, family, and teachers.

Acknowledgments

I would like to thank my adviser, Professor Andrew Alleyne, for his guidance throughout my graduate studies. I have been incredibly fortunate for the numerous opportunities you have provided and your continued commitment to my professional and personal development despite being over 500 miles away. In my future ventures, I hope that I can make a similarly positive impact on others. Dr. Alleyne has been a great mentor, and thank you for fostering such an open and friendly work environment.

I would like to thank and acknowledge my doctoral committee, Professors Prashant Mehta, Carolyn Beck, Kai James, and Dr. Daniel Clark Jr. for their feedback during my studies.

I have been lucky to work with many outstanding individuals and organizations throughout my graduate studies. Thank you to all past and present members of the Alleyne Research Group (ARG). Special thanks to Justin Koeln for introducing me to research as a undergraduate. Your mentorship has had a clearly significant impact on my life and career. I'm grateful to Matt Williams, Herschel Pangborn, Pamela Tannous, Ashley Armstrong, Oyuna Angatkina, Malia Kawamura, Nate Weir, Bryan Keating, Spencer Kieffer, Sarah Garrow, Sunny Sharma, and Spencer Igram (S^3) for welcoming me into the lab. Thank you to Donald Docimo, Cary Laird Butler, Mindy Wagenmaker, Phil Renkert, Reid Smith, Kayla Russel, Frank Andujar Lugo, Chris Urbanski, Dylan Charter, Kurt Kuipers, Zhara Marvi, Bo Wang, Enrico Sisti, and Erik Salazar for being great lab mates and friends throughout my graduate studies.

I would like to thank POETS students, faculty, and staff. Special thanks to Jodi Gritten who has made mine and so many other graduate students lives much easier. Thank you to Jack Jordan and Cyril Rybicki for your help building and instrumenting the testbeds I used in my dissertation. Thank you to Dr. Karen Bender, Sonya Chambers, Dr. Jessica Perez,

Joe Muskin, John Wierschem, Kiruba Haran, Owen Doyle, Nenad Miljkovic, and Sherry Yu for providing a multitude of opportunities for POETS Students.

Thank you to all my collaborators at and with the Air Force Research Laboratory and CU Aerospace. Your insights have been very helpful in shaping and providing value to my work. I have appreciated your mentorship.

Thank you to my family for your continued support and interest in my work. Thank you to my friends for always being available to get me away from work. Although our in-person interactions were limited by a pandemic over the past few years, I am thankful that we have still been able to stay in touch virtually a few times a week. Since the start of the pandemic, we have play over 400 games of euchre, of which I have won 53.8%. Is it luck or is it skill?

Lastly, I would like to acknowledge the support of the National Science Foundation Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) under cooperative agreement EEC-1449548. This support has provided me the flexibility to pursue transformative research that will benefit both my field and society. I would also like to thank the American Society of Mechanical Engineers (ASME) for the Graduate Teaching Fellowship. Teaching Signal Processing was one of the highlights of my graduate career.

Contents

List of Tables

List of Figures

List of Abbreviations

AVC	Air Vehicle Controller
ARE	Algebraic Riccati Equation
AE	Algebraic Equations
Arch.	Architecture
AALQR	Architecture Adaptive Linear Quadratic Regulator
CDKF	Central Difference Kalman Filter
CLCD	Closed-Loop Co-Design
CP	Cold Plate
DAE	Differential Algebraic Equations
DP	Dynamic Programming
Dyno.	Dynamometer Motor
EPS	Electrical Power System
eVTOL	Electric Vertical Take-Off and Landing
ECMS	Equivalent Consumption Minimization Strategy
EKF	Extended Kalman Filter
FoM	Figure of Merit
Gen.	Generator Motor
HX	Heat Exchanger
IPPTS	Integrated Power, Propulsion, and Thermal Management System
LQR	Linear Quadratic Regulator

LV	Low Voltage
L1, L2, L3	Fluid Loops 1, 2, and 3
MPS	Mechanical Power System
MIQP	Mixed-Integer Quadratic Program
MPC	Model Predictive Control
MEA	More Electric Aircraft
MDO	Multi-disciplinary Design Optimization
OLCD	Open-Loop Co-Design
ODE	Ordinary Differential Equations
PT	Powertrain
PTC	Powertrain Controller
P1S1	1 Primary and 1 Secondary Flow Paths
P2S1	2 Primary and 1 Secondary Flow Paths
PID	Proportional-Integral-Derivative
Prop.	Propulsion Motor
PWM	Pulse Width Modulation
QP	Quadratic Program
TMS	Thermal Management Systems
TMSC	Thermal Management System Controller
UAV	Unmanned Aerial Vehicle
UAM	Urban Air Mobility

List of Symbols

All variables and notation is defined in text when introduced. Common variables and notation are defined below. All model variable definitions are provided in the Appendix in Tables B.1-B.12.

Common Variables

a	Algebraic state/variable
J	Cost/objective function
d	Disturbance
x	Dynamic state/variable
P^e	External power flow
x^e	External vertex state
C	Graph capacitance
D	Graph external incidence matrix
M	Graph internal incidence matrix
P	Graph power flow
u	Input
N_i	Number of ...
x_{ref}	Reference for variable x
A, B, C, D, E, F, G, H	State space model matrices
σ	Switching variable
ρ	Topology design variable

λ	Topology selector, plant, or control design parameter

Notation and Operations

	$\|x\|_\Lambda^2 = x^T \Lambda x$
	$[1:N] = \{1, 2, \ldots, N\}$
	$x[k] = x_k$
$\otimes$	Kronecker product
$\bar{\cdot}$	Upper bound or equilibrium condition
$\underline{\cdot}$	Lower bound
$N(\cdot, \cdot)$	Normal distribution

Chapter 1

Introduction

1.1 Electrified Mobility

In response to a zero-emission regulation for production vehicles in California, General Motors released the EV1 as one of the first commercially available all electric cars in 1996. Equipped with an all electric powertrain, the EV1 had zero emissions and a fast, at the time, 8.4s 0-60mph time that was touted by celebrities on live TV [1]. However, battery technology was nascent and early models had a limited 50 miles of range per charge. In the early 2000s, the emissions regulations were repealed and the EV1 production was eventually stopped. General Motors decision to end the program was motivated by a lack of public interest and profitability, citing that modern battery technology could not meet consumers' demands for more range [2].

About 5 years after the introduction of the original EV1, the Toyota Prius was released with an estimated fuel economy that approximately doubled the fuel economy of many similar compact cars released that year, thus garnering attention for those looking to save money. The high fuel economy is attributed to the Prius' hybrid powertrain design that leveraged both a battery and internal combustion engine, allowing the vehicle to operate efficiently over a range of operating speeds. Then in 2008, the Tesla Roadster was released, a production all-electric sports vehicle with a recorded 3.7s 0-60mph time that outpaced the Ferrari F430's recorded 3.9s time. The Roadster showed significant improvements over the EV1, demonstrating that electrified vehicles have both a positive environmental impact and high-end performance. Again in 2012, Tesla was revitalized through the production of its luxury sedan, the Model S; a daily driver with a 4.6s 0-60mph time and over 200 estimated miles of range. Since 2012, the all electric vehicle market has continued to grow (Figure

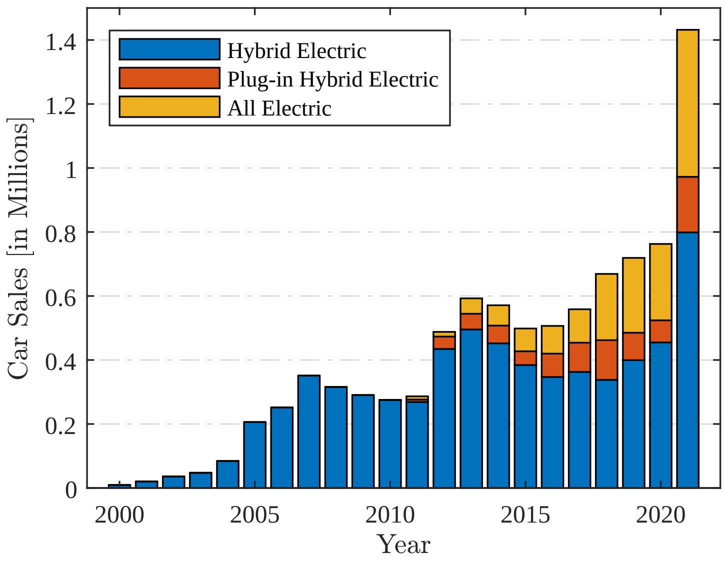


Figure 1.1: The increasing sales trend of hybrid electric, plug-in hybrid electric, and all electric cars sales in the United States [3].

1.1) as a result from advancements in power system technology such as more energy dense batteries yielding over 400 miles of range. In 2002, electric vehicles were not profitable, but now, in 2023, an electric car company is one of the highest valued companies in the United States. Although this anecdote neglects to mention years of foundational research and other landmark developments in all electric and hybrid powertrains, it highlights how persistent advancements resulted in widespread adoption of a once impractical technology.

While engineers at GM were designing the EV1, their counterparts in the aviation sector were focused on developing the concept of the More Electric Aircraft (MEA) [4]. The concepts were similar; leverage evolving advancements in electrical power systems to replace traditionally power intensive and inefficient pneumatic and hydraulic systems with their efficient and power dense electronic analogs. Note that these aircraft are only *more* electric, meaning that they still utilize traditional fuel-powered engines for propulsion. The key design advancement is the replacement of the traditionally non-electric subsystems with electrical ones. Similar to the EV1, the MEA concept was motivated by a beneficial environmental impact, but there was also additional expectation of increased performance and reliability and decreased operating and maintenance costs. The hybridization of aircraft was going to be a necessary step in the evolution towards the all electric aircraft.

But it was not until the early 2010s where the MEA transitioned from concept to reality with the release of the Boeing 787 and Airbus A350. As proposed in the 1990s, these aircraft replaced traditional power demanding pneumatic systems with more power dense 1MW electrical systems. Although the propulsion system remained entirely fuel powered, hybridizing the power system resulted in significant improvements in fuel economy, decreased maintenance costs, and greater cruise speeds. All of these aspects demonstrated increased profitability with positive environmental impact.

In the early 1990s, into the 2000s, and even into the 2020s, batteries are often cited as one of the key technological bottlenecks for electric vehicles because a lack of energy density severely limits range. With the current state of battery technology, it is estimated that all electric aircraft may have at most a few hundred miles of range [5]. Although limited, the range offers a promising business case for the aviation sector.

Short-haul flights (<1000mi) currently comprise an estimated 30% of the CO_2 emissions in the commercial aviation space (Figure 1.2). Therefore, government and industries leaders have identified short-haul and regional flights to be an environmentally beneficial and profitable market in which to develop and mature electric aviation technologies. This is precisely the business case for Heart Aerospace, who plans to deliver 200 all electric aircraft with an estimated 250 mile (400 km) range for short regional flights [6].

In addition to short range commercial flights, there is also interest in the design of *urban air mobility* (UAM) vehicles. Often referred to as air taxis, UAM vehicles are designed as autonomous or piloted all electric vertical take-off and landing (eVTOL) aircraft with the purpose to transport people and/or cargo through urban environments with a range spanning from 10s to 100s of miles [8]. A key capability enabled by electric power systems is the ability for quiet flight at low altitudes. This contrasts commonly noisy helicopters that are should cruise at altitudes over 600-1200 m to not perturb civilians [9].

Unmanned aerial vehicles (UAVs) are also being electrified. Advancements in electronic power systems and the removal of the pilot and life support systems enable a diverse new frontier of applications that include surveying, disaster relief, film and imaging, agriculture, package delivery, and more [10]. Some of these applications utilize novel high power electronic

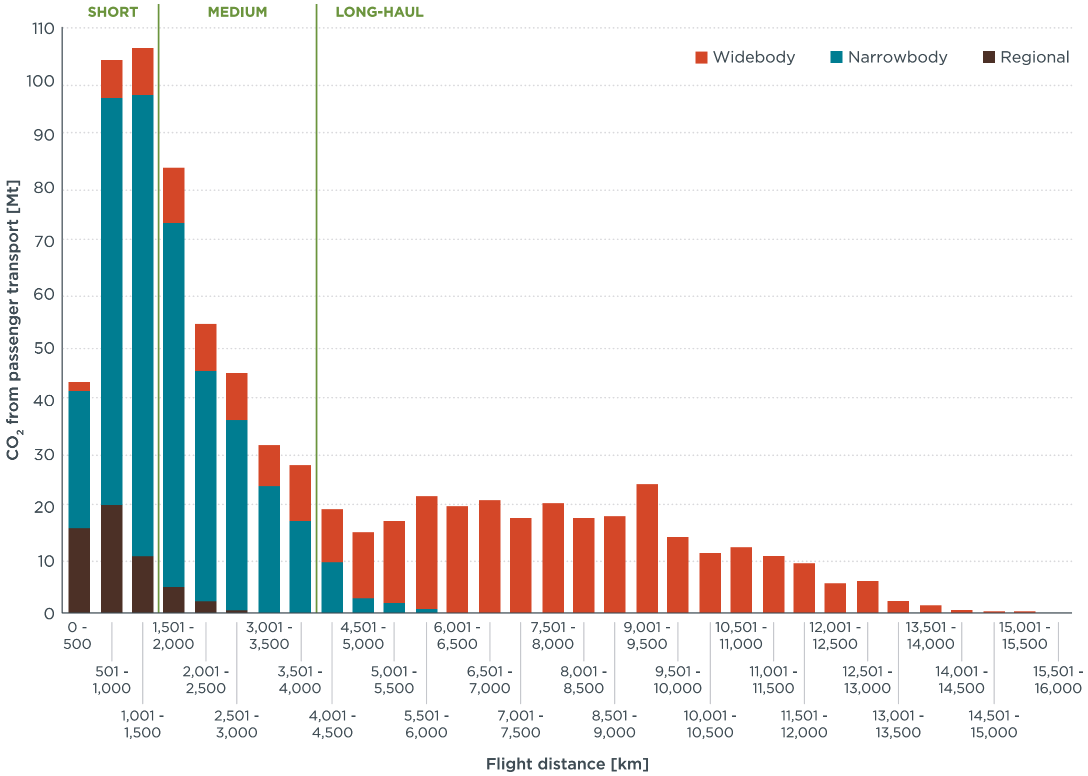


Figure 1.2: Trend of CO_2 emissions for short, medium, and long-haul commercial flights. Approximately 30% of total emissions are generated by short and medium distance flights each. This figure is from [7].

sensors or payloads (e.g. professional camera equipment) that require onboard electric power systems, thus demanding some degree of electrification in the aircraft design.

The design and manufacture of all electric and hybrid aircraft has been motivated by increased efficiency, novel capabilities, improved maintainability, and decreased operating costs. However, vehicle range limits widespread adoption of electric aircraft. One approach to address this challenge is to adopt a mission-specific design framework, wherein the vehicle design is tailored to its application. For example, a close inspection of Figure 1.3 highlights a diverse set of aircraft designs from different companies with unique body shapes, propulsion systems, etc. Although each vehicle serves the same general purpose to transport people short distances through urban environments, each vehicle looks dramatically different. Naturally, one may ask:

Which design is best?

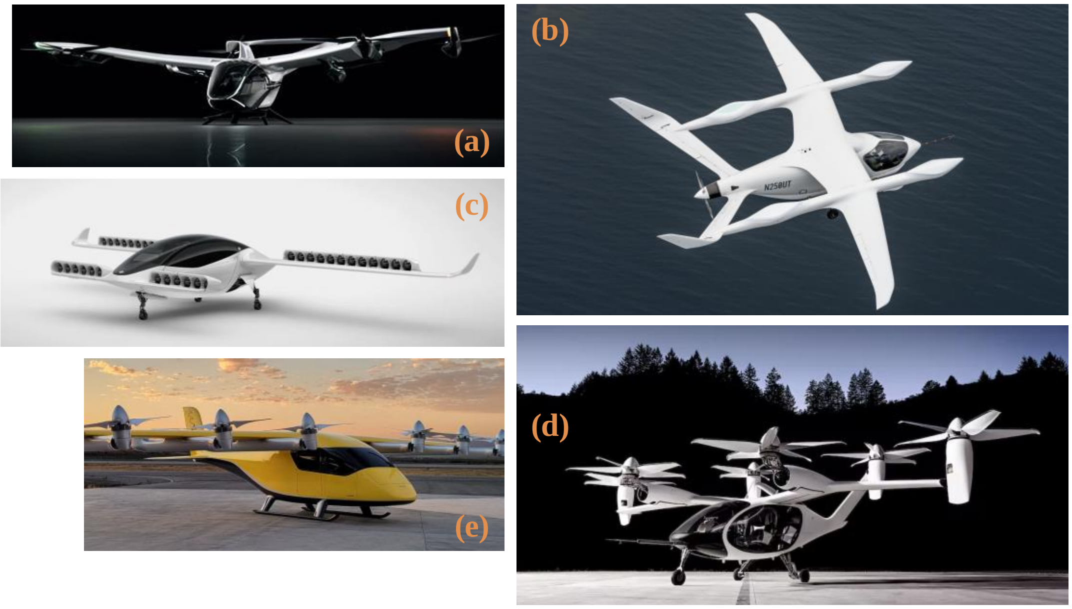


Figure 1.3: Urban air mobility vehicles from different manufacturers: (a) Airbus, (b) Beta, (c) Lilium, (d) Joby Aviation, and (e) Wisk.

or

Could the design be improved if the vehicle transported cargo instead of people?

In summary, aircraft electrification has various societal benefits. A few key advantages are:

- *Sustainability*: By leveraging efficient power systems and renewable energy sources, electrified aircraft offer a promising approach to reduce fuel dependence.
- *Capabilities*: Electrified aircraft offer the advantage of quiet and autonomous operation, with a range of advanced high-power electrical sensors, which makes them well-suited for a diverse set of applications.
- *Power and energy density*: Enhanced energy storage and utilization results in increased aircraft range and greater support for high-power subsystem capabilities.

In particular, this dissertation will focus on developing technologies to improve the power and energy density of electrified systems with a mission mission-specific design methodology tailored for aircraft. However, the scale and complexity of aircraft and their design process pose a variety of unique challenges.

1.1.1 Challenges

A mission-specific design framework seeks to simplify and standardize the complex electrified aircraft design decision-making process. One challenge to consider is *mission variability* wherein single multirole aircraft are designed to complete multiple missions. While versatile, integrating multiple operational capabilities into a single system design increases the vehicle complexity and may result in decreased performance, efficiency, and reliability.

This dissertation chooses to focus on the complex interactions in the energy systems. Notably, the power systems onboard electrified aircraft have inherent efficiencies that manifest themselves as internal heat generation. In response to the heat generation, the system temperature changes, which in turn affects performance and efficiency. A common example of this *power and thermal system coupling* is present in battery systems that operate more efficiently at higher temperatures [11] but fail if they get too hot.

The thermal runaway failure mode highlights the strict *operating constraints* for electrified systems. System reliability suffers if components get too hot, too cold, or operate outside their intended operating regions. These types of failure modes are not acceptable, especially for aircraft that must have less than 1 failure per million flights. Notably, in 2013, a fleet of Boeing 787 Dreamliners were grounded as a result of battery fires caused by internal short circuits [12].

In addition to being coupled and constrained, the energy domain dynamics are also *multi-timescale* where some system states change quickly while others change slowly. To advance the state of the art, it is necessary to account for the temporal disparity within the energy systems [13]. However, it is difficult to systematically account for the timescale variation within the design process.

1.2 State of the Art

Designing aircraft is a complex process, requiring engineers to balance a multitude of factors in order to produce a vehicle that is both efficient and high-performing. From the airframe design to the selection of materials and components, and the development of the control systems, every decision made during the design phase has a significant impact on the final product. To aid in this complex process, optimization methods are utilized to systematically evaluate the impact of key design decisions on operational objectives and constraints. Due to the multifaceted nature of this complex design problem, multi-disciplinary design optimization (MDO) methods are commonly employed, enabling multiple design disciplines to be considered within the same design problem. In particular, *control co-design* (or just *co-design*) is a subset of MDO problems for dynamic system design wherein both the plant and control design disciplines are investigated at the same time. This class of problems has become increasingly relevant because it has been demonstrated that plant and control design decisions are strongly coupled [14].

Figure 1.4 illustrates key steps in the system-level vehicle co-design process: topology generation, topology optimization, sizing optimization, and control optimization [15]. The topology problem involves generating and assessing potential system architectures, with the aim of determining the configuration that best meets the design objective. Subsequently, the sizing problem involves selecting the optimal set of components and parameters that will be utilized to build the desired system topology. Together, architecture and sizing decisions comprise the plant design process. To enable effective plant operation, a controller is designed during the control optimization phase. In the co-design process, consideration is given to

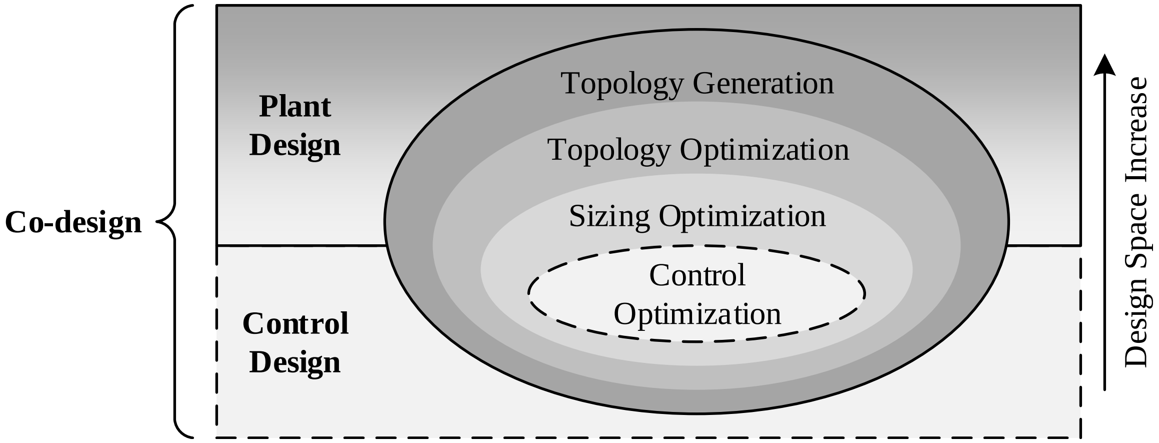


Figure 1.4: Illustration of the plant and control design optimization decisions that comprise the co-design process. The different design disciplines are sized according to the relative size of the design space. Modified from [15].

Table 1.1: System-level energy system co-design studies.

References	Topology	Sizing	Closed-Loop Control	Experimental Validation
[16, 17]	✓			
[18, 19]	✓	✓		
[20–22]	✓		✓	
[23–32]		✓		
[33]		✓		✓
[34–40]		✓	✓	
[41–56]			✓	
[13, 57]			✓	✓

either open-loop co-design (OLCD), which identifies an optimal open-loop control action, or closed-loop co-design (CLCD), which involves the development of an optimal closed-loop controller. Notably, OLCD studies are primarily focused on the plant design process because the application of open-loop control to physical systems is impractical when uncertainty is present.

Traditionally, these design decisions are made sequentially, beginning with topology selection, followed by component selection, and concluding with controller design [15]. However, this sequential approach often leads to suboptimal designs that would not meet the requirements of next-generation aircraft. Instead, it has been shown that an integrated approach, involving simultaneous optimization of both plant and control design, leads to improved system performance [14]. This approach represents the current trend in aircraft co-design. Figure 1.4 is presented with Table 1.1, which highlights various design studies in the co-design literature. A brief review of various topology, sizing, and control design studies follows.

The topology generation process involves constructing the overall system design space by determining the types of systems that meet practical considerations. This can be achieved through generative approaches [58, 59], wherein realizable designs are constructed based on a set of rules, or exhaustive approaches [60–62], where all possible designs are built and then rules are imposed to reduce the design space. While effective, both approaches require a thorough understanding of how the design rules impact the resulting design space.

After feasible topologies are generated, the topology, sizing, and control optimization processes are applied. In [16], various thermal management system topologies were generated and dynamically evaluated with open-loop control against a variety of thermal load profiles to identify design configurations that maximized thermal endurance. Similarly, a tool for systematic generation and evaluation of diverse thermal management system architectures was introduced in [17]. Unlike the approach in [16], the tool does not generate optimal designs. Instead, it utilizes dynamic simulation data to evaluate operational constraints, effectively reducing the feasible topology design space. The result is a reduced set of dynamically feasible designs, which provides a more focused design space for the application of sizing and control design techniques. An alternate graph-based approach for simultaneous topology and sizing was demonstrated with an electric car in [18] wherein the design problem considered the sizing and architecture decisions of the vehicle's power and thermal system. While [16–18] focused on generating and selecting novel system architecture designs, [19–22] instead tailor their approach to designing a controller that can switch between topologies. In [20, 21], they generate driving modes for a hybrid electric vehicle powertrain and use an iterative optimization approach to select the best two driving modes that a controller can switch between in real-time. Notably, the hybrid system is controlled by an equivalent consumption minimization strategy controller. A similar investigation was conducted in [19] where the authors also exhaustively considered motor size in the design problem. A controllable and flexible thermal management system topology was proposed in [22], where two control variables are selected to change between, and evaluate the performance of, different system configurations. Another notable contribution, although not considering energy system design, was in [63] where the authors introduced a tri-level approach to topology generation and optimization, sizing, and open-loop control design. While many of these studies consider system architecture, they neglect either system size or closed-loop control design.

In the context of energy system co-design studies, trajectory optimization is a widely employed tool for developing an optimal plant size and open-loop control action that meets power and thermal constraints. This need was illustrated in [23] wherein a candidate aircraft could complete a minimum-time-to-climb problem 15% faster when considering adequate thermal management system design. In a subsequent study by Hendricks et al. [24], the co-design of a turboelectric tiltwing aircraft was further investigated to include concurrent optimization of thermal and propulsion system sizing, wing shape, flight dynamics, and trajectory. A similar investigation was conducted in [25] where the authors investigated performance and design trade-offs between simultaneous and sequential design methodologies. Renkert et al. designed an electric quadcopter and optimized its trajectory to navigate an obstacle course in minimum time using validated component models [33]. Similar studies have investigated the combined plant sizing and open-loop trajectory optimization, while considering battery and/or motor design, for high performance electric race cars [26, 27]. In [28], the authors utilized a sizing optimization approach to design a hybrid electric aircraft that incorporated a novel thermal management system. The results demonstrated that while the system offered increased robustness to flight and environmental conditions, it came at the cost of increased fuel consumption. To design robust vehicles, chance constrained co-design methods have been developed [64] and applied to both aircraft [29] and truck [30] design. Some other notable contributions for combined plant sizing and open-loop control design include [31, 32]. While previous studies have highlighted the advantage of generating design insights through integrated plant and control design, none of them are physically realizable as they neglect the closed-loop control design.

Physical systems are designed with closed-loop controllers to account for inherent uncertainty resulting from environmental disturbances or system operation. Some approaches integrate traditional proportional-integral-derivative (PID) control into the co-design process. In [34], the authors present a graph-based approach for combined plant sizing and feedback control optimization. They applied their method on a hybrid energy storage system and two PI controllers, yielding improved reference tracking performance and decreased system weight. In another study [35], a continuously variable transmission and a feedback linearization PI controller were jointly designed, resulting in a more efficient and lightweight system. In [36], a DC motor and its PID controller were designed, which was later extended in [37] to develop a robust design using a min-max co-design optimization approach. In these approaches [34–37], the optimization process directly selects the feedback control gains. However, some alternative methods have instead leveraged efficient control synthesis techniques such as LQR [65] or H_∞ [38] within nested co-design processes. Other notable contributions, not relating to energy system co-design, are included [66–71].

However, for complex energy systems, relatively simple tracking and regulatory controllers are typically insufficient for mission level-coordination. Instead, optimization-based dynamic programming (DP) or model predictive control (MPC) are commonly used. The key difference between the two approaches is that the DP optimization process is solved offline and implemented online through lookup tables or heuristics [41] while the MPC optimization is directly solved and implemented online [72]. Centralized MPC approaches for energy system management have been designed [42–45], but suffer when applied to energy systems with multiple timescales [13]. This limitation has been addressed through distributed [46–49] or hierarchical [50–55] MPC strategies that redistribute the centralized optimization's computational load amongst multiple controllers [73]. In one study [56], the authors proposed a multi-horizon MPC that could coordinate multi-timescale dynamics within a centralized MPC formulation. In [13,57], optimal controllers were experimentally validated on testbeds representing vehicle energy management systems. Some studies have utilized DP [39] or MPC [40,74] based controllers within a co-design process, but with the drawback of lengthy optimization times.

1.3 Research Objectives

The literature review in Table 1.1 highlights that while several studies have focused on specific aspects of topology, sizing, and control design, there remains a research gap considering the entire co-design problem. This dissertation aims to address this gap by proposing a mission-specific design framework for aircraft energy systems that simultaneously optimizes topology, sizing, and control to achieve better performance, efficiency, and reliability (Table 1.2).

Table 1.2: A summary of this dissertation's contributions.

References	Topology	Sizing	Closed-Loop Control	Experimental Validation
This Dissertation	✓	✓	✓	✓
Chapter 3			✓	✓
Chapter 4		✓	✓	
Chapter 5	✓		✓	

1.3.1 Desired Capabilities

Electrified aircraft offer increased efficiency, novel capabilities, require less maintenance, and have decreased operating costs. Conventional design approaches do not adequately account for mission variability and the complex nature of multi-domain, constrained, and multi-timescale behavior of aircraft energy systems. To advance the state of the art, it is necessary to shift away from sequential design strategies and towards more advanced integrated co-design approaches.

Therefore, this dissertation proposes a mission-specific approach to energy system design that considers the following:

- *Physics-based models*: Manufacturing physical candidate system designs can be a time-consuming and expensive process. Therefore, this approach should leverage experimentally validated physics-based models that represent the behavior of physical systems. This will allow new designs to be developed from subsystem models without prior data on the overall system.

- *Modularity and scalability*: A mission-specific approach may consider the design of multiple vehicle systems. Therefore, the proposed methods should be able to design different systems of varying scale and complexity for a variety of applications.

- *Simultaneous plant and control design*: Sequentially designing the plant then controller will lead to suboptimal system performance. Leveraging co-design methods that account for coupling between the design disciplines will yield more effective system designs.

1.3.2 Dissertation Scope

The contributions of this dissertation are as follows:

1. A modular, scalable, optimization-oriented, and energy conservation-based modeling framework that captures relevant dynamic interactions of aircraft energy systems.

2. The development and experimental validation of a hierarchical model predictive controller for an aircraft integrated power, propulsion, and thermal management system (IPPTS) that accounts for dynamic multi-domain and multi-timescale coupling amongst subsystems to demonstrate improvements in closed-loop system performance, efficiency, and reliability.

3. The development and demonstration of a generic and scalable sizing and closed-loop control co-design approach. Application of the approach to a candidate hybrid UAV design highlights improvements in design performance over alternative design strategies.

4. A topology and closed-loop control co-design framework that relaxes the discrete topology decisions to continuous decisions, which can be more efficiently evaluated using gradient-based optimization techniques. The design methods are applied to develop mission-specific thermal management system designs.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces the graph-based modeling approach. Derived from conservation laws, the graph-model modeling framework captures multi-domain dynamic interactions present in energy systems. A reduced-order modeling technique, linearization, and discretization methods are introduced. The chapter concludes with an example showing how to interpret the graph-based models introduced in the subsequent chapters.

Chapter 3 demonstrates the utility of hierarchical MPC for a UAV integrated power, propulsion, and thermal management system. A hierarchical MPC is designed and experimentally validated on a hardware-in-the-loop experimental testbed that represents an unmanned air vehicle IPPTS. When compared to a baseline PI and logic-based control approach, the hierarchical control demonstrated superior closed-loop system performance, efficiency, and reliability.

Chapter 4 proposes a scalable approach for sizing and closed-loop control co-design. A convergence study is introduced to evaluate the number of discretization nodes required for accurate optimization results. The proposed co-design approach is compared against alternative strategies by designing a hybrid UAV energy storage system, propulsion system, and their respective feedback controllers. The proposed approach demonstrates improved system performance while generating useful design insights.

Chapter 5 proposes a continuous formulation of the traditionally discrete topology and closed-loop control co-design problem. The discrete design choices are relaxed so that the topology space is instead represented by a convex set. To dynamically model the continuum of system topologies, a superset system model is introduced that captures the dynamic interactions of every feasible topology design. An architecture adaptive linear-quadratic regulator is introduced that recomputes an optimal feedback gain for the continuum of system topologies. Application of the proposed methods to a mission-specific aircraft thermal management system design demonstrate optimal changes in system configurations for variable energy and power mission requirements.

Chapter 6 summarizes the contributions of this dissertation and suggests directions for future work.

Chapter 2

Modeling

2.1 Background

Mathematical models are commonplace in various engineering disciplines because they allow engineers to understand and evaluate the performance of their design in a low-cost computational environment instead of through expensive physical prototypes. As such, these models serve as foundational elements that guide the engineering design process. Selecting a *correct* model is a challenge itself given the wide breadth of models introduced in the engineering literature. As an engineer, it is necessary to focus on the specific problem and select a model that best suits the application. Modeling characteristics that are relevant to consider for the co-design optimization research in this dissertation include:

- *Energy domain agnostic* - The ability for the model to capture complex dynamic interactions across various energy domains and timescales.
- *Optimization-oriented* - The measure of a model's computationally efficiency/cost and compatibility with conventional optimization tools.
- *Modularity* - A model's ability to be reused or repurposed for alternative applications.
- *Scalability* - A model's ability to represent systems of variable size and complexity.
- *Variable fidelity* - The option to change the model's accuracy to fit the computational limits of the end application.

It is necessary that the models are energy domain agnostic because this dissertation investigates the design and control of coupled electrical, thermal, and mechanical dynamics that comprise vehicle energy management systems. Optimization methods, which systematically evaluate many design points before converging to an optimal solution, are commonplace in modern decision-making processes, so developing models that are both computationally efficient and compatible with optimization tools is required. Such optimization-oriented models facilitate a more thorough search of the design space and help return a design solution in less time. The latter three characteristics are desirable to generalize a given modeling method beyond a single design or use case. In particular, a modular modeling approach is required to systematically change system architectures for the topology design process introduced in Chapter 5. What follows is a brief review of common modeling tools and frameworks published in the literature, and an evaluation of the modeling requirements.

Toolboxes/libraries are amongst the most common approaches to model energy systems. They provide a user interface in which developers can visually connect components/blocks in a variety of configurations to represent the dynamics of a wide class of systems. Modelica and Simulink-based toolboxes are amongst the most common. Relevant dynamic system oriented modeling toolboxes include Dymola (Dynamic Modeling Laboratory) and MapleSim, which are built in Modelica, and Simscape and AttmoSphere [75], which are built in Simulink. Similarly, GT-Suite and AMESIM are other notable general-purpose tools built outside the Modelica and Simulink frameworks. Other notable Simulink toolbox contributions tailored to specific applications include Thermosys [76], PowerFlow [77], T-MATS (Toolbox for the Modeling and Analysis of Thermodynamic Systems), EMTAT (Electrical Modeling and Thermal Analysis Toolbox), and Power Systems Analysis Toolbox (PSAT) [78]. Powerflow is developed for analysis and control design of aircraft, Thermosys is tailored towards vapor cycle system design and control, and PSAT focuses on the design and analysis of power systems. Together, the NASA toolboxes EMTAT and T-MATS facilitate the analysis and control design of dynamic electrical and thermal systems. Toolbox models are modular and well parameterized for scalability, which is a key advantage of the toolbox approach. The tip-to-tail system representation enables rapid model prototyping that provides a visual means to comprehend and convey the flow of information within the models. However, model dynamics and parameter sensitivities are often obscured from the user, making it difficult to integrate such models into optimization processes. Although manually reconfigurable, it is also challenging to systematically evaluate the performance of various model topologies, which is a core focus of this dissertation. In summary, Simulink toolboxes are a powerful prototyping and simulation tool, but have less utility in more advanced optimization algorithms.

Another viable approach are bond graphs, which model the dynamics of physical systems using arrows of effort and flows. Bond graph modeling focuses on the development of a causal model [79], and are often modular and scalable, but become increasingly complex when modeling large scale systems [80]. Additionally, bond graphs lack conventional analysis tools (e.g. model order reduction [81]) that facilitate design or control optimization. Finite element tools (e.g. Ansys) may also be useful, particularly for design optimization applications. However, these tools sacrifice computational efficiency and modularity for accuracy, yielding these models impractical for control design.

More recently, there has been some interest in the development of graph-based models for conservation-based systems. In the context of energy system modeling, graph-based models reduce the modeling challenge to a conservation of energy problem, where the modeler seeks to understand how energy is both stored and transferred within a system. Whilst not a panacea, graph models have various advantages. They are inherently modular and scalable and have been used to represent a wide array of physical energy systems [34,53,54,57,82–84] at varying degrees of fidelity [85,86]. Most importantly, they are computationally efficient [87], which facilitates their utilization in model-based optimization algorithms [13, 33, 34, 57]. However, there are some key limitations. First, the system must have conservation-based dynamics, a restriction that limits the research application area. Even so, there are some conservation-based systems that graph models have difficulty representing, such as compressible flow systems. Also, some systems have very complex interactions or strong coupling that may be challenging to represent as a graph model. An example of this challenge was shown in [57] where a genset model was developed outside the graph-based modeling framework.

Graph-based models meet the modeling requirements for this dissertation research because a conservation-based modeling strategy is well suited to represent the dynamics of aircraft energy systems. In some studies, graph-based models have been experimentally validated for this application [53, 57]. From previous plant and control design studies [13, 33, 34, 57], it is evident that graph-based models are optimization oriented as well. In particular, this dissertation intends to leverage graph-model modularity to systematically evaluate different energy system topologies (Chapter 5); a capability that some other modeling techniques lack.

The remainder of this chapter is organized as follows. Section 2.2 introduces the graph-based modeling framework fundamentals and its extension to dynamic and algebraic systems. An approach to model order reduction is introduced in Section 2.3. Section 2.4 describes the model linearization and discretization process. This chapter will not introduce any specific graph-based models used in the dissertation research, and instead those models will be described in the chapter they are used. Therefore, Section 2.5 will provide an example that describes how to interpret the graph-based models introduced in the following chapters. Last, Section 2.6 will conclude this chapter and summarize its contributions.

2.2 Graph-Based Modeling Framework Fundamentals

A graph-based model represents conservation-based dynamics through an oriented graph of *vertices* and *oriented edges*, as illustrated in Figure 2.1. In the context of energy systems, graph-based models obey conservation of energy laws, where the graph vertices represent energy storage devices and the edges represent energy transfer mechanisms. These energy transfer mechanisms are often referred to as *power flows*. Note that power flow is bidirectional and that the edge orientation defines the direction of positive power flow. In Figure 2.1, vertices and edges have solid or dashed lines. The linestyle distinguishes between internal system interactions (solid lines) and external system interactions/ disturbances (dashed lines). The following notation is used to describe a graph model.

An oriented graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of N_v vertices $v_i \in \mathcal{V} : i \in [1 : N_v]$ and N_e oriented edges $e_j \in \mathcal{E} : j \in [1 : N_e]$. The set of all vertices can be decomposed into subsets of *internal vertices* $\mathcal{V}^i$ and *external vertices* $\mathcal{V}^e$ such that $\mathcal{V}^i \cup \mathcal{V}^e = \mathcal{V}$. Similarly, the set of all edges can be decomposed into *internal edges* $\mathcal{E}^i$ and *external edges* $\mathcal{E}^e$ such that $\mathcal{E}^i \cup \mathcal{E}^e = \mathcal{E}$. The number of internal vertices, external vertices, internal edges, and external edges are $N_{v,i}$,

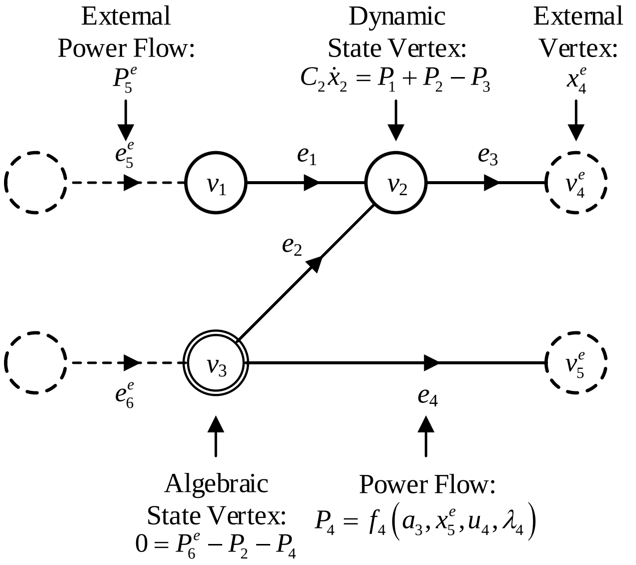


Figure 2.1: Notional graph model from [81].

$N_{v,e}$, $N_{e,i}$, and $N_{e,e}$, respectively, such that $N_{v,i} + N_{v,e} = N_v$ and $N_{e,i} + N_{e,e} = N_e$. Each oriented edge e_j connects a *tail vertex* $v_j^{tail} \in \mathcal{V}_j^{tail}$ to a *head vertex* $v_j^{head} \in \mathcal{V}_j^{head}$. Similarly, the set of edges entering and leaving the vertex v_i is denoted by $\mathcal{E}_i^{head} := \left\{e_j : v_j^{head} = v_i\right\}$ and $\mathcal{E}_i^{tail} := \left\{e_j : v_j^{tail} = v_i\right\}$, respectively. In graph models, states x are associated with internal vertices, and power flows P are associated with internal edges. External disturbances x^e and P^e are associated with external vertices and edges, respectively.

The state dynamics of each vertex obey conservation of energy laws,

$$C_i \dot{x}_i = \sum_{\{j:e_j \in \mathcal{E}_i^{head}\}} P_j - \sum_{\{j:e_j \in \mathcal{E}_i^{tail}\}} P_j, \tag{2.1}$$

where $C_i \geq 0$ is the *capacitance* of the vertex v_i. An individual vertex capacitance C_i may be a function f_i of the vertex state x_i and model parameters λ_i

$$C_i = f_i\left(x_i, \lambda_i\right). \tag{2.2}$$

Note that the state x_i is a scalar, while the parameters λ_i may be a vector. An individual power flow P_j is computed with a function f_j of the adjacent vertex states (x_j^{tail} and x_j^{head}), inputs u_j, and model parameters λ_j

$$P_j = f_j\left(x_j^{tail}, x_j^{head}, u_j, \lambda_j\right). \tag{2.3}$$

Note that here x_j^{tail} and x_j^{head} are scalars because each edge can only have one tail and one head vertex, while u_j or λ_j may be vectors if multiple inputs or parameters affect the power flow along that edge.

Combining (2.1)-(2.3), the complete graph model dynamics can be written in a state-space-like form

$$C(x,\lambda)\dot{x} = -\bar{M}P(x, x^e, u, \lambda) + DP^e \tag{2.4}$$

where $x \in \mathbb{R}^{N_{v,i}}$ is the state vector, $C \in \mathbb{R}^{N_{v,i} \times N_{v,i}}$ is a diagonal capacitance matrix of the vertex capacitances, $P \in \mathbb{R}^{N_{e,i}}$ is a vector of all the edge power flows, $\bar{M}$ is the upper internal incidence matrix, and D is the external incidence matrix. *Incidence matrices* describe the connectivity and structure of a graph model.

The internal incidence matrix $M = [m_{ij}] \in \mathbb{R}^{N_v \times N_{e,i}}$ describes the internal structure of the graph, where

$$m_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is the tail of } e_j, \\ -1 & \text{if } v_i \text{ is the head of } e_j, \\ 0 & \text{else.} \end{cases} \tag{2.5}$$

The internal incidence matrix can be partitioned as

$$M = \begin{bmatrix} \bar{M} \\ \underline{M} \end{bmatrix} \text{ with } \bar{M} \in \mathbb{R}^{N_{v,i} \times N_{e,i}}, \tag{2.6}$$

such that $\bar{M}$ maps the power flows to affected states.

The external incidence matrix $D = [d_{ij}] \in \mathbb{R}^{N_{v,i} \times N_{e,e}}$ describes how the external edges connect to internal vertices

$$d_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is the head of } e_j^s, \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

While this formulation was presented in the context of energy systems, these methods can be applied to other conservation-based systems. An example is mass conservation systems where the vertices represent mass storage (e.g. a tank or vessel) and the edges represent mass transfer (e.g. tubes or pipes) [53].

2.2.1 Dynamic and Algebraic Vertices

Recall that any vertex has a capacitance value $C_i \geq 0$. A vertex is a *dynamic vertex* when $C_i > 0$. When $C_i = 0$, the vertex is called an *algebraic vertex* because (2.1) simplifies to an algebraic equation relating the power inflow and outflow,

$$0 = \sum_{\{j:e_j \in \mathcal{E}_i^{head}\}} P_j - \sum_{\{j:e_j \in \mathcal{E}_i^{tail}\}} P_j. \tag{2.8}$$

There is a change in notation at this point. Dynamics states are represented by the variable x and algebraic states are represented by the variable a. They are illustrated by single and double-edged vertices, as shown in Figure 2.1, respectively. A graph model may have N_d and N_a dynamic and algebraic vertices, respectively. Without loss of generality, (2.4) can be rewritten as

$$\begin{bmatrix} C_d \dot{x} \\ 0 \end{bmatrix} = - \begin{bmatrix} \bar{M}_d \\ \bar{M}_a \end{bmatrix} P(x, a, x^e, u, \lambda) + \begin{bmatrix} D_d \\ D_a \end{bmatrix} P^e \tag{2.9}$$

where C_d is a diagonal capacitance matrix of the dynamic vertices' capacitance values and the upper internal and external incidence matrices have been appropriately partitioned into $\bar{M}_d$, $\bar{M}_a$, D_d, and D_a to represent the internal and external structure of dynamic and algebraic vertices.

Depending on the number of dynamic and algebraic vertices, the graph model (2.9) may take on one of three different forms: differential algebraic equations (DAE), ordinary differential equations (ODE), and system of algebraic equations (AE). The graph model is a DAE when $N_d > 0$ and $N_a > 0$, is an ODE when $N_d > 0$ and $N_a = 0$, and is an AE when $N_d = 0$ and $N_a > 0$. Because the DAE form is the most general, it will be utilized in the following sections.

2.3 Graph-Based Model Order Reduction

Adhering to a variable fidelity modeling framework is one approach to generating computationally efficient design and control-oriented models. Although high order models (e.g. finite element) may be accurate, they are impractical for design and control optimization problems because they have significant computational requirements. To address this issue, it is common to create reduced order models that approximate the high-order system dynamics with fewer states. Several model-order reduction techniques that preserve model structure have been proposed in the literature [86,88]. An aggregation-based approach was introduced where states are combined into superstates to generate a reduced-order model with multiple timescales [85]. For optimization-oriented applications, an approach similar to the singular perturbation method [89,90] is well suited and easily handled by the graph-based modeling framework.

The singular perturbation approach is most commonly applied to *stiff* systems comprised of a combination of fast and slow dynamics, thus applicable to multi-timescale energy systems. Stiff systems are difficult to integrate into model-based optimization studies because they require both small discretization intervals and large optimization time horizons to capture the fast and slow dynamic behavior. The resulting optimization would have many design variables and significant computational requirements. The singular perturbation approach addresses both the stiffness and model order issues by approximating the fast dynamic behavior with a static relationship. However, a key component in this approach is identifying the set of fast states to approximate as static.

For linear or linearized nonlinear systems, the fast states are generally associated with the model's largest magnitude eigenvalues. However, it can be challenging to directly correlate eigenvalues to states. An advantage of graph models is that the vertex capacitance values often describes the state's timescale [13, 85]. Therefore, the graph-based framework makes it possible to identify correlation between eigenvalues and states of a graph-based model. Note that this is not a rigorous approach to timescale classification, but it may be useful for some applications like those introduced in Section 3.4.

Applying an approach similar to the singular perturbation technique to graph-based models requires identifying the fastest state dynamics and setting the capacitance of the associated states to zero. The result of the model order reduction is a DAE graph model, which was introduced in Section 2.2.1. Without loss of generality, suppose a graph-based model is provided in DAE form (2.9) where N_d and N_a are the number of dynamic and algebraic vertices, respectively. Using an eigenvalue analysis, the system modeler categorizes the first N_s dynamic states as slow and the remaining N_f dynamic states as fast, such that $N_s + N_f = N_d$. To apply the model order reduction technique, partition the DAE model (2.9) according to slow, fast, and algebraic states x_s, x_f, and a, respectively (2.10a). Note that the subscripts refer to the fast s, slow f, and algebraic a parameters. After partitioning the DAE graph, set the capacitance value for fast vertices to zero $C_f = 0$ (2.10b).

$$\begin{bmatrix} C_f \dot{x}_s \\ C_s \dot{x}_f \\ 0 \end{bmatrix} = - \begin{bmatrix} \bar{M}_s \\ \bar{M}_f \\ \bar{M}_a \end{bmatrix} P(x_s, x_f, a, x^e, u, \lambda) + \begin{bmatrix} D_s \\ D_f \\ D_a \end{bmatrix} P^e, \tag{2.10a}$$

$$\begin{bmatrix} C_s \dot{x}_s \\ 0 \\ 0 \end{bmatrix} = - \begin{bmatrix} \bar{M}_s \\ \bar{M}_f \\ \bar{M}_a \end{bmatrix} P(x_s, x_f, a, x^e, u, \lambda) + \begin{bmatrix} D_s \\ D_f \\ D_a \end{bmatrix} P^e, \tag{2.10b}$$

Then combine the fast and algebraic state partitions to yield a reduced order DAE graph-based model with fewer dynamic states and more algebraic states,

$$\begin{bmatrix} \hat{C}_d \dot{\hat{x}} \\ 0 \end{bmatrix} = - \begin{bmatrix} \hat{\bar{M}}_d \\ \hat{\bar{M}}_a \end{bmatrix} P(\hat{x}, \hat{a}, x^e, u, \lambda) + \begin{bmatrix} \hat{D}_d \\ \hat{D}_a \end{bmatrix} P^e, \tag{2.11}$$

where,

$$\hat{x} = x_s, \qquad \hat{\bar{M}}_d = \bar{M}_s, \qquad \hat{\bar{D}}_d = \bar{D}_s, \qquad \hat{C}_d = C_s,$$

$$\hat{a} = \begin{bmatrix} x_f \\ a \end{bmatrix}, \quad \hat{\bar{M}}_a = \begin{bmatrix} \bar{M}_f \\ \bar{M}_a \end{bmatrix}, \quad \hat{\bar{D}}_a = \begin{bmatrix} \bar{D}_f \\ \bar{D}_a \end{bmatrix},$$

where the ˆ indicates the reduced order analog of the standard graph model variables from (2.9). Note that this reduction technique is structure preserving because the reduced-order model (2.11) is in the same form as the full-order model (2.9). As mentioned in [89], the reduced order algebraic equations may have multiple roots that may result in unrealistic behavior for physical systems, so special care should be taken when applying this technique.

2.4 Linearization and Discretization

Complex optimizations often leverage linear models to more efficiently solve the optimization problem. Linear models approximate nonlinear behavior through a first order Taylor expansion,

$$f(x) \approx f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} (x - \bar{x}) \tag{2.12}$$

where $f(x)$ is a function and $\bar{x}$ is the linearization point. The following details the linearization of a graph-based model.

The general nonlinear power flow vector function can be linearized about an operating point $\bar{z} = (\bar{x}, \bar{a}, \bar{x}^e, \bar{u})$

$$P(z) \approx P(\bar{z}) + \underbrace{\left.\frac{\partial P}{\partial x}\right|_{\bar{z}}}_{P_x} (x - \bar{x}) + \underbrace{\left.\frac{\partial P}{\partial a}\right|_{\bar{z}}}_{P_a} (a - \bar{a}) + \underbrace{\left.\frac{\partial P}{\partial x^e}\right|_{\bar{z}}}_{P_e} (x^e - \bar{x}^e) + \underbrace{\left.\frac{\partial P}{\partial u}\right|_{\bar{z}}}_{P_u} (u - \bar{u}) \,. \tag{2.13}$$

Note that the parameter argument λ is omitted because parameters are commonly constant.

If the model is parameter varying or state dependent, these same techniques can be applied as presented. Plugging the linearized power flow equation into (2.9) yields

$$\begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \approx \begin{bmatrix} A_{dd} & A_{da} \\ A_{ad} & A_{aa} \end{bmatrix} \begin{bmatrix} x \\ a \end{bmatrix} + \begin{bmatrix} B_d \\ B_a \end{bmatrix} u + \begin{bmatrix} E_d \\ E_a \end{bmatrix} d + \begin{bmatrix} G_d \\ G_a \end{bmatrix}, \tag{2.14}$$

where,

$$\begin{aligned}
A_{dd} &= -C_d^{-1}\bar{M}_d P_x, & A_{ad} &= -\bar{M}_a P_x, \\
A_{da} &= -C_d^{-1}\bar{M}_d P_a & A_{aa} &= -\bar{M}_a P_a, \\
B_d &= -C_d^{-1}\bar{M}_d P_u, & B_a &= -\bar{M}_a P_u, \\
E_d &= \begin{bmatrix} -C_d^{-1}\bar{M}_d P_e & C^{-1} D_d \end{bmatrix}, & E_a &= \begin{bmatrix} -\bar{M}_a P_e & D_a \end{bmatrix}, \\
G_d &= -A_{dd}\bar{x}_d - A_{da}\bar{x}_a - B_d\bar{u} - C_d^{-1}\bar{M}_d P_e \bar{x}^e - C_d^{-1}\bar{M}_d P(\bar{z}), \\
G_a &= -A_{ad}\bar{x}_d - A_{aa}\bar{x}_a - B_a\bar{u} - \bar{M}_a P_e \bar{x}^e - \bar{M}_a P(\bar{z}),
\end{aligned}$$

and $d = \begin{bmatrix} x^e \\ P^e \end{bmatrix}$. The presented model is a linear DAE, and the same general approach can be applied to an ODE or AE form graph model. For dynamic systems, it is often preferred to work with a linear state-space model:

$$\dot{x} = Ax + Bu + Ed + G, \tag{2.15a}$$

$$a = Cx + Du + Fd + H, \tag{2.15b}$$

where,

$$\begin{aligned}
A &= A_{dd} - A_{da}A_{aa}^{-1}A_{ad}, & B &= B_d - A_{da}A_{aa}^{-1}B_a, \\
E &= E_d - A_{da}A_{aa}^{-1}E_a, & G &= G_d - A_{da}A_{aa}^{-1}G_a, \\
C &= -A_{aa}^{-1}A_{ad}, & D &= -A_{aa}^{-1}B_a, \\
F &= -A_{aa}^{-1}E_a, & H &= -A_{aa}^{-1}G_a.
\end{aligned}$$

The model can be discretized using conventional approaches such as Forward Euler or Runge-Kutta 4 approximations. For linear systems, the author's preferred approach is an exact discretization that assumes zero-order hold on the model inputs and disturbances. The linear system (2.15) is discretized at the discretization interval T

$$x\left[k+1\right] = A_z x\left[k\right] + B_z u\left[k\right] + E_z d\left[k\right] + G_z, \tag{2.16a}$$

$$a\left[k\right] = C_z x\left[k\right] + D_z u\left[k\right] + F_z d\left[k\right] + H_z, \tag{2.16b}$$

where,

$$\begin{aligned} A_z &= e^{AT}, & B_z &= \left(\int_0^T e^{A\tau} d\tau\right) B, \\ E_z &= \left(\int_0^T e^{A\tau} d\tau\right) E, & G_z &= \left(\int_0^T e^{A\tau} d\tau\right) G, \\ C_z &= C, & D_z &= D, \\ F_z &= F, & H_z &= H, \end{aligned}$$

where k is the discrete-time index and e^{AT} is a matrix exponential. In future sections, a more simple notation is adopted $x\left[k\right] \coloneqq x_k$.

2.4.1 Switched Linear Model Representation

In preview of the control results introduced in Chapter 3, a single linear model may not represent the nonlinear system dynamics over a sufficiently large state space. While one could opt to use a full nonlinear model, it comes at the expense of increased computational cost and local minima when integrated into an optimal controller. An alternative approach would be to use multiple linear models, each of which represent the nonlinear system dynamics in a different part of the state space. This modeling approach is called a switched linear model. In a switched linear model representation, the nonlinear dynamics are linearized about an integer number of linearization points N_m to generate N_m unique linear models. Each of the linear models is saved, and a switching signal $\sigma(t)$ is used to determine which linear model is active

$$\sigma(t) : [1 : \infty) \rightarrow [1 : N_m] = \{1, 2, \ldots, N_m\} \subset \mathbb{Z}^+. \tag{2.17}$$

Using the linearization and discretization process introduced in Section 2.4, the switched linear representation of the nonlinear power flow vector is,

$$P^\sigma(z) \approx P(\bar{z}^\sigma) + \underbrace{\left.\frac{\partial P}{\partial x}\right|_{\bar{z}}}_{P_x^\sigma}(x - \bar{x}^\sigma) + \underbrace{\left.\frac{\partial P}{\partial a}\right|_{\bar{z}}}_{P_a^\sigma}(a - \bar{a}^\sigma) + \underbrace{\left.\frac{\partial P}{\partial x^e}\right|_{\bar{z}}}_{P_e^\sigma}(x^e - \bar{x}^{e,\sigma}) + \underbrace{\left.\frac{\partial P}{\partial u}\right|_{\bar{z}}}_{P_u^\sigma}(u - \bar{u}^\sigma), \quad (2.18)$$

where the superscript σ describes the linearization point associated with a given operating mode. Plugging the switched linear power flows $P^\sigma(z)$ into the DAE graph model (2.9) yields,

$$\begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \approx \begin{bmatrix} A_{dd}^\sigma & A_{da}^\sigma \\ A_{ad}^\sigma & A_{aa}^\sigma \end{bmatrix} \begin{bmatrix} x \\ a \end{bmatrix} + \begin{bmatrix} B_d^\sigma \\ B_a^\sigma \end{bmatrix} u + \begin{bmatrix} E_d^\sigma \\ E_a^\sigma \end{bmatrix} d + \begin{bmatrix} G_d^\sigma \\ G_a^\sigma \end{bmatrix}, \quad (2.19)$$

where the switched matrix sub-partitions are computed according to the approach introduced in (2.14). Using the same technique from Section 2.4, the switched linear DAE model can be converted to a switched linear ODE form,

$$\dot{x} = A^\sigma x + B^\sigma u + E^\sigma d + G^\sigma, \quad (2.20a)$$

$$a = C^\sigma x + D^\sigma u + F^\sigma d + H^\sigma, \quad (2.20b)$$

where the switched model matrices are computed using (2.15). The ODE model can be discretized assuming zero-order hold on the inputs and disturbances to generate a switched linear discrete-time model,

$$x_{k+1} = A_z^\sigma x_k + B_z^\sigma u_k + E_z^\sigma d_k + G_z^\sigma, \quad (2.21a)$$

$$a_k = C_z^\sigma x_k + D_z^\sigma u_k + F_z^\sigma d_k + H_z^\sigma, \quad (2.21b)$$

where the switched discrete time state-space matrices are computed using (2.16).

2.5 Working With and Interpreting Graph-Based Models

Graph models are inherently modular, allowing model developers to start with a library of components that can be connected in various ways to develop system graph models. Many component graph models for energy systems have been introduced in the literature [34, 53, 54, 57]. Additionally, [33] introduces an approach and toolbox designed to assemble component graph models into system graph models. Because these topics have been introduced previously in the literature, they will not be discussed in this dissertation, and instead an interested reader is directed to [33, 91] for more information. The following section describes the derivation of a graph model's equations from its graph and key model information.

In the following three chapters, graph-based models will be fully described by 5 pieces of information: the model structure, and vertex, edge, input, and parameter information. The model structure is introduced graphically while the vertex, edge, input, and parameter information is presented in tables. This notation was introduced in [92].

An example graph-based model is illustrated in Figure 2.2 and vertex, edge, input, and parameter information is provided in Tables 2.1 to 2.4. In future chapters, the graph vertices may be colored to represent energy domains, and additional information, such as variable units, may be provided in the tables. Using (2.5), and (2.7), the graph model internal and external incidence matrices are derived from Figure 2.2,

$$M = \begin{bmatrix} \bar{M}_d \\ \hline \bar{M}_a \\ \hline \underline{M} \end{bmatrix} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{array}\right], \qquad D = \begin{bmatrix} \bar{D}_d \\ \hline \bar{D}_a \end{bmatrix} = \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \\ \hline 0 & 1 \end{array}\right]. \tag{2.22}$$

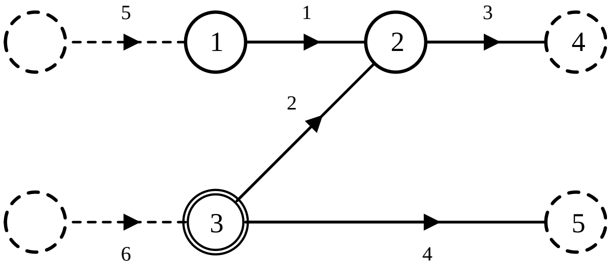


Figure 2.2: Example graph to show how to construct a graph model from its Tables 2.1 to 2.4.

Table 2.1: Vertex information for the example graph model.

Vertex	State Variable	Description	Capacitance
1	x_1	Example Vertex 1	$C_1 = \lambda_a x_1$
2	x_2	Example Vertex 2	$C_2 = \lambda_b$
3	a_3	Example Vertex 3	$C_3 = 0$
4	x_4^e	Example Vertex 4	-
5	x_5^e	Example Vertex 5	-

Table 2.2: Edge information for the example graph model.

Edge	Description	Power flow
1	Example Edge 1	$P_1 = x_1 x_2$
2	Example Edge 2	$P_2 = \lambda_1 (a_3 - x_2)$
3	Example Edge 3	$P_3 = u_1 \lambda_2^2 x_2^2$
4	Example Edge 4	$P_4 = u_2 \lambda_1 \lambda_2 a_3 x_5^e$
5	Example Edge 5	P_5^e
6	Example Edge 6	P_6^e

Table 2.3: Input information for the example graph model.

Input	Description
u_1	Example Input 1
u_2	Example Input 2

Table 2.4: Parameter information for the example graph model.

Parameter	Description	Value
λ_a	Example Capacitance Parameter a	100
λ_b	Example Capacitance Parameter b	1200
λ_1	Example Edge Parameter 1	18
λ_2	Example Edge Parameter 2	16

Using Tables 2.1 and 2.2, the capacitance matrix and power flow vector are developed. Note that the capacitance and power flow equations are arbitrarily selected for illustrative purposes and do not represent physical energy storage and transfer mechanisms.

$$C = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} = \begin{bmatrix} \lambda_a x_1 & 0 & 0 \\ 0 & \lambda_b & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} x_1 x_2 \\ \lambda_1 (a_3 - x_2) \\ u_1 \lambda_2^2 x_2^2 \\ u_2 \lambda_1 \lambda_2 a_3 x_5^e \end{bmatrix} \tag{2.23}$$

The incidence, capacitance, and power flow matrices are plugged into (2.9) to define the graph model equations.

$$\begin{bmatrix} \lambda_a x_1 & 0 & 0 \\ 0 & \lambda_b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ 0 \end{bmatrix} = - \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 x_2 \\ \lambda_1 (a_3 - x_2) \\ u_1 \lambda_2^2 x_2^2 \\ u_2 \lambda_1 \lambda_2 a_3 x_5^e \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_5^e \\ P_6^e \end{bmatrix} \tag{2.24}$$

Once in graph model form, the model can be used as introduced or converted to various other forms such as a nonlinear state space, linear state space, or discrete time forms depending on the use case. For example, the equivalent nonlinear state-space DAE form system is provided below.

$$\begin{aligned} \dot{x}_1 &= \left(-x_1 x_2 + P_5^e\right) / \lambda_a x_1 \\ \dot{x}_2 &= \left(x_1 x_2 + \lambda_1 (a_3 - x_2) - u_1 \lambda_2^2 x_2^2\right) / \lambda_b \\ 0 &= -\lambda_1 (a_3 - x_2) - u_2 \lambda_1 \lambda_2 a_3 x_5^e + P_6^e \end{aligned} \tag{2.25}$$

2.6 Conclusion

This chapter introduces the graph-based modeling framework and its extension to dynamic and algebraic systems. Although described in the context of energy systems, the presented graph-based modeling formulation is generic and can be extended to the general class of conservation-based systems. Model order reduction, linearization, and discretization processes were described to create a model better suited for integration in optimization routines. These processes will become relevant to the optimization-based tools introduced in the remainder of this dissertation. This chapter concludes with an example that demonstrates deriving a graph-based model's equations from its graphical representation and table information.

As mentioned at the start of the chapter, mathematical system models are foundational in the development of model-based analysis, design, and control tools. In order to support the system and control design objectives of the following three chapters, the models should be energy domain-agnostic, optimization-oriented, modular, scalable, and have variable fidelity. Because graph-based models meet all of these criteria, they are leveraged within the plant and control co-design studies presented in the following chapters.

Chapter 3

Hierarchical Predictive Control for Energy Systems

3.1 Background

Electrification in the aviation industry is driven by the combined need for more sustainable and capable aircraft. However, an electrified aircraft's novel power system is inherently inefficient, which results in heat generation and changes in operating temperatures. As temperatures change, so does the vehicle's operating performance and efficiency. Accounting for this coupling between power and thermal systems is required in control design to improve performance and efficiency, but also to maintain safe and reliable operation. Various studies have investigated the development of centralized MPC to address this challenge [42–45], however, the multi-timescale nature of power and thermal systems makes it difficult to design an effective real-time capable MPC [13]. Other studies have investigated hierarchical MPC to address the multi-timescale limitation of the centralized approach [50–55]. In hierarchical MPC, the overall control task is completed with multiple coordinated predictive controllers with different update rates, each tuned to operate a partition of the entire system. The decomposition of the multi-timescale control problem allows hierarchical MPC to retain the performance benefits of centralized MPC with the additional advantage of real-time capability.

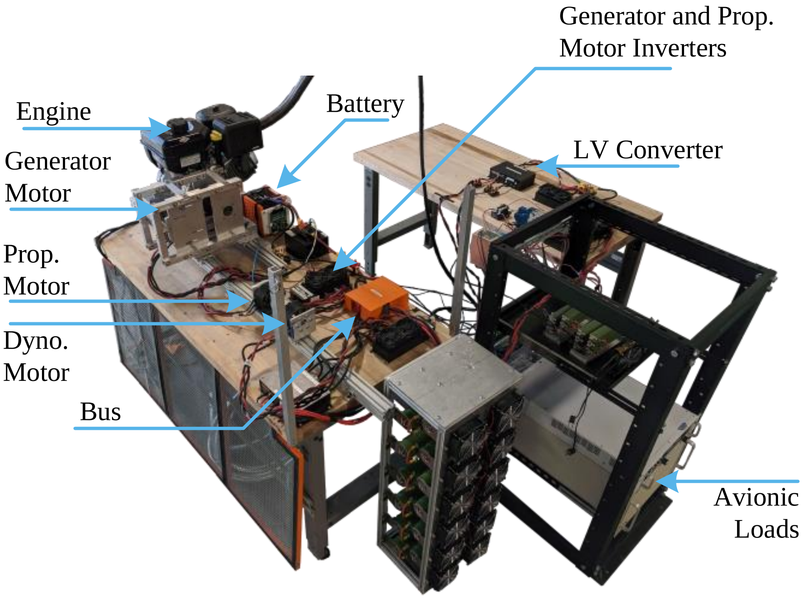
While various MPCs have been developed, few have been experimentally validated on aircraft energy systems. In [13], MPC was experimentally validated on aircraft fuel thermal management system. However, that study used a simple and static power system model that did not consider the fast timescale electrical and mechanical system dynamics. Conversely, a hierarchical MPC was experimentally validated on a UAV powertrain testbed in [57] to highlight superior multi-timescale coordination, but did not consider the system's thermal response.

Therefore, the key contribution of this chapter is the design and experimental demonstration of a hierarchical MPC for an entire aircraft integrated power, propulsion, and thermal management system; highlighting the practicality of the proposed control strategy.

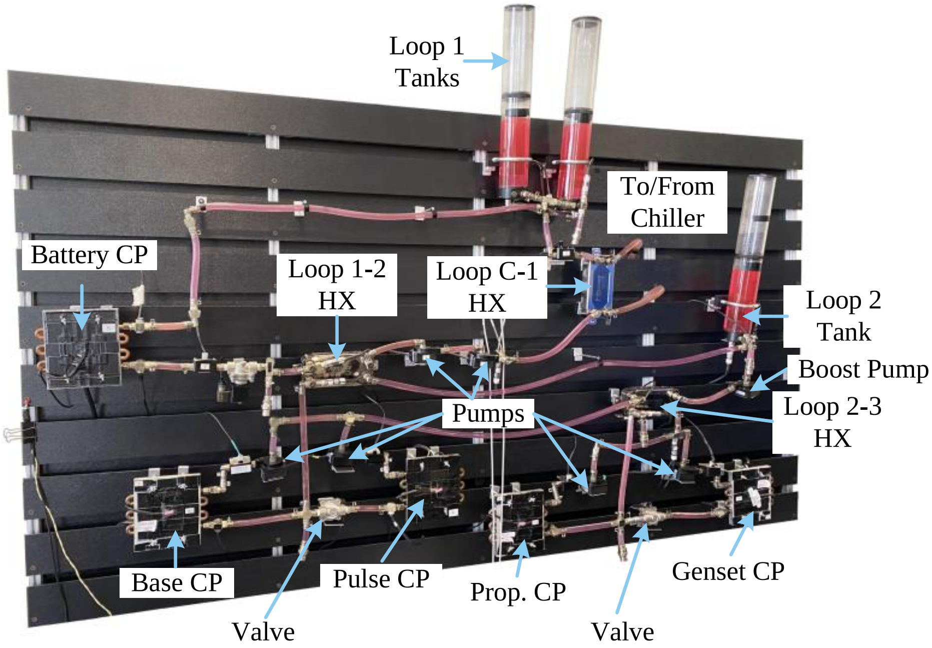
3.2 System Description

The aircraft IPPTS is represented by the experimental testbeds illustrated in Figure 3.1, and comprises an electrical power system (EPS), mechanical power system (MPS), and thermal management system (TMS). The combined EPS and MPS will be referred to as the aircraft's powertrain (PT). The testbed's key energy system components are illustrated schematically in Figure 3.2. While this system architecture does not represent any platform in production, the architecture and component selection was designed to capture the complex multi-domain dynamic coupling that is characteristic of electrified aircraft. The PT was designed based on a standard series hybrid powertrain topology [93] while the TMS was designed based on single-phase air vehicle cooling architectures in the literature [94,95]. It is necessary to emphasize that the PT and TMS testbeds illustrated in Figure 3.1 are physically disconnected, so they could be built, instrumented, and debugged separately without the risk that a cascade of failures affects the entire system. Although physically decoupled, the systems are virtually coupled via a hardware-in-the-loop configuration that allows the emulation of key dynamic interactions between PT and TMS.

The EPS comprises a battery, electronic power conversion devices, and electrical loads. The battery provides dedicated electrical energy storage and can charge and discharge depending on the system's electrical power demand and generation. As illustrated in Figure 3.2a, the battery is directly connected to the aircraft's electrical bus that connects to the other subsystems. In the avionics subsystem, the bus connects to the low voltage (LV) DC/DC converter that regulates the voltage supplied to the base and pulse load converters. The base and pulse loads represent various electronic power requirements for onboard computers, sensors, payload, etc. Specifically, the pulse load is selected to represent the pulsed electrical power requirements of next-generation electrified aircraft [96–98]. Notably, the base and pulse load electrical efficiency is functionally dependent on their operating temperature, which is a key characteristic of electrical power systems. In hardware, the base and pulse loads and their converters are emulated in software with a programmable

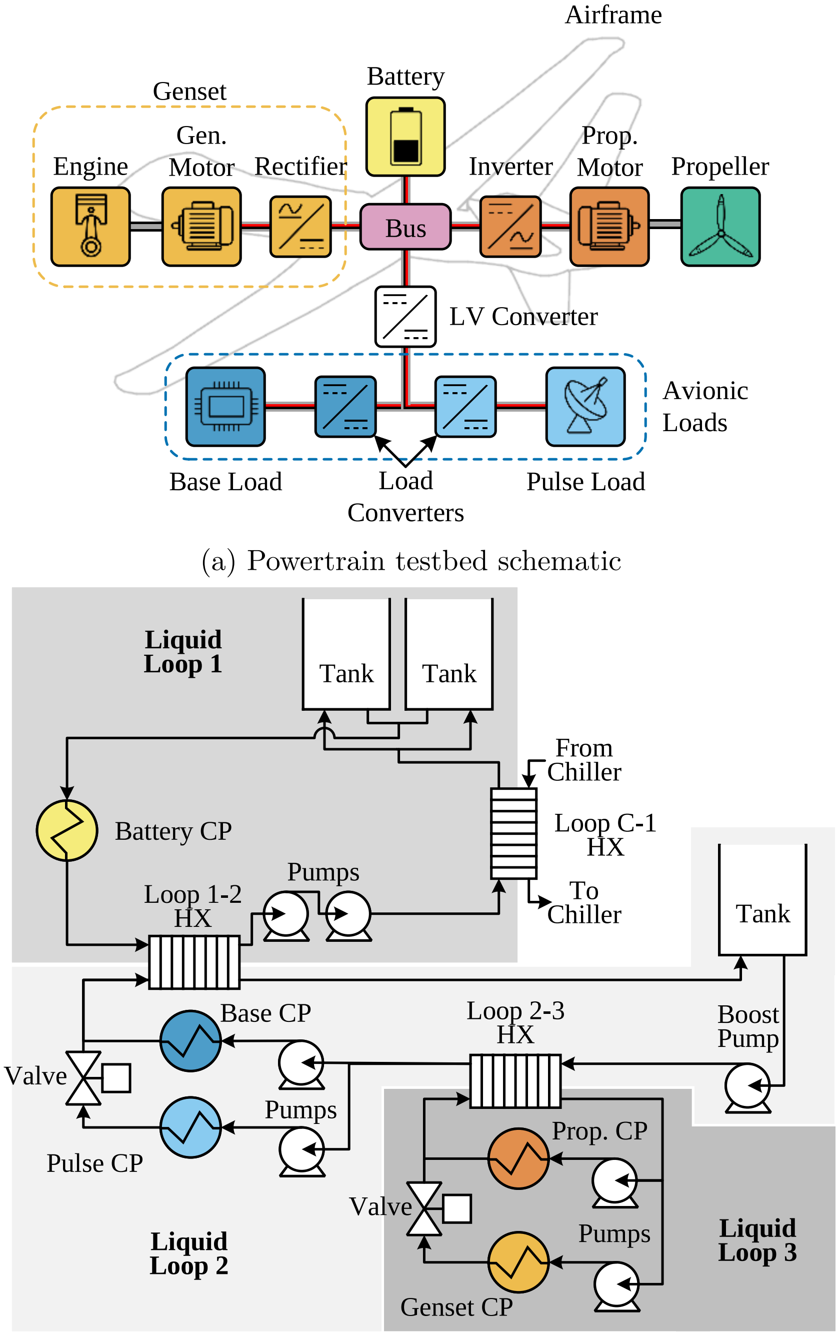


(a) Powertrain testbed



(b) Thermal management system testbed

Figure 3.1: Labeled images of the (a) powertrain and (b) thermal management system testbeds.



(a) Powertrain testbed schematic

(b) Thermal management system testbed schematic

Figure 3.2: Labeled schematics of the (a) powertrain and (b) thermal management system testbeds.

3kW electronic load labeled avionic loads in Figure 3.1a. Also, directly connected to the electrical bus are AC/DC and DC/AC electrical power inverters that control the amount of power generated by and provided to the generator (gen.) and propulsion (prop.) motors, respectively. The generator and propulsion motors couple the EPS to the MPS.

The MPS comprises an internal combustion engine, generator motor, propulsion motor, and propeller. The engine is coupled to the generator via a belt and pulley system and the generator's terminals connect to the AC/DC inverter, and together, the combination of these components (engine, generator, inverter) is called the aircraft's *genset*. The purpose of the genset subsystem is to convert energy stored as fuel to electrical power that can be used to recharge the battery or power the other electrical loads. Note that the generator is also used as a starter to remote start the engine. The aircraft's propulsion subsystem comprises a DC/AC electrical power inverter, propulsion motor, and propeller. The subsystem converts electrical power from the EPS into rotational mechanical power to generate thrust and propel the aircraft. On the experimental testbed, the propulsion loads are simulated in software and imposed on the hardware via a dynamometer (dyno.) motor (Figure 3.1a).

The EPS and MPS generate heat through inherent inefficiencies such as conduction and friction losses. To dissipate this heat, the EPS and MPS are coupled to the TMS that absorbs the heat and rejects it to the chiller (heat sink) onboard the aircraft. The candidate TMS comprises three serially connected fluid loops in Figure 3.2b where the arrows illustrate the fluid flow direction. Heat generated by the battery is rejected into the fluid loop 1 cold plate (CP). Fluid moving through the cold plate absorbs heat and passes through the loop 1-2 heat exchanger (HX) where additional heat is exchanged with fluid loop 2. After passing through the loop 1-2 heat exchanger, heat is rejected to the aircraft's chiller via the loop C-1 heat exchanger. Two large fluid tanks at the top of loop 1 store cool fluid and increase the loop's thermal capacity.

Heat generated by the base and pulse loads and their converters is rejected into the base and pulse cold plates in fluid loop 2. Fluid passing through those cold plates absorbs heat, which is then exchanged with loop 1 via the loop 1-2 heat exchanger. After passing through the loop 1-2 heat exchanger, fluid passes through the fluid tank, and then through the loop 2-3 heat exchanger. As in loop 1, the tank in fluid loop 2 increases the loop's thermal capacity. Because the pulse load has intermittent operation, a binary on/off valve is added inline on that flow path and can be actuated to bypass the pulse cold plate. The bypass

Table 3.1: Summary of the coupling mechanisms present in the aircraft energy systems.

System 1	System 2	Type	Mechanism
EPS	MPS	Bidirectional	Electro-magnetics
EPS	MPS	Bidirectional	Electro-magnetics
MPS	TMS	Unidirectional	Heat load
MPS	TMS	Unidirectional	Heat load
EPS	TMS	Unidirectional	Heat load
EPS	TMS	Bidirectional	Heat load
EPS	TMS	Bidirectional	Heat load

path allows more cooling capacity to be allocated to the base cold plate. Because loop 2 is large, it is equipped with a single high-speed boost pump to increase the flow rates.

Heat generated by the propulsion and genset subsystem is rejected into the propulsion and genset cold plates in loop 3. Loop 3 has similar structure to loop 2, except it does not have a fluid tank or a third pump. The valve in loop 3 is included to divert coolant to the propulsion cold plate when the genset is not being used.

Given the complexity of the IPPTS, it is necessary to summarize the various types of coupling between systems. Both unidirectional (one-way) and bidirectional (two-way) coupling is present. Unidirectional coupling is present when one system's behavior is dependent on a second system, but the second system's behavior is independent of the first system. For bidirectionally coupling, both systems' behavior is dependent on the other. As previously described, the aircraft's energy system comprises coupled EPS, MPS, and TMS. The EPS exhibits bidirectional coupling with the MPS through the generator and prop. motors that transform electrical to mechanical energy and vice versa. The MPS has unidirectional coupling with the TMS, where the heat generated by the prop. motor, generator motor, and their power electronics is directly imposed on the TMS. The EPS battery exhibits unidirectional coupling, where the heat generated is imposed on the TMS. The avionic loads demonstrate bidirectional coupling where heat generated by the avionic loads is imparted on the TMS and the temperature of the TMS affects the efficiency of the avionic loads. All cross-system coupling information is summarized in Table 3.1.

3.2.1 Testbed Capabilities

A description of the testbed components is provided in Appendix A. The PT battery and genset can supply up to 6kW and 3kW of electrical power, respectively, to the propulsion and avionic load subsystems. Both the motor and generator have a 14kW power rating, however, forced air cooling is required for power levels exceeding 4kW to prevent thermal degradation to the motor windings. Note that the 14kW generator is well oversized for the 3kW engine to prevent overheating. The testbed's low voltage subsystems can support up to 3.0kW continuous electrical power, but requires forced air cooling for power levels exceeding 2.25kW. The avionic loads are emulated by a 3kW programmable DC electronic load. In summary, the power system can supply and sink up to a 9kW and 17kW of electrical power, respectively. The motors, battery, and power electronics are equipped with constant speed fans to prevent overheating and potential damages.

Each cold plate on the TMS testbed is equipped with resistive heaters that can emulate heat loads up to 2kW heat loads. The aircraft heat sink is emulated by a chiller with a 5kW cooling capacity. The heat exchangers are rated to transfer up to 4.0kW of heat at 5LPM and a 20°C inlet temperature. Centrifugal pumps are powered by external power supplies and can provide up to 17.5LPM flow rates. In total, the TMS can absorb and reject up to 10kW and 5kW of thermal energy, respectively.

3.2.2 Virtual Testbed Coupling

The PT and TMS are not physically connected, so they are instead coupled virtually via a hardware-in-the-loop configuration with a real-time simulation environment (Figure 3.3). This virtual coupling eases the testbed design process and improves system safety. As illustrated in Figure 3.3, both testbeds communicate with a LabView application running on two separate secondary computers. In turn, the LabView application communicates over User Datagram Protocol (UDP) to a primary control computer running a real-time simulation environment, which is Simulink in this application. This primary control computer simulates the virtual system dynamics, handles various signal processing tasks, and calls the control systems that will be introduced in Section 3.4 and 3.5. The key virtualized system dynamics

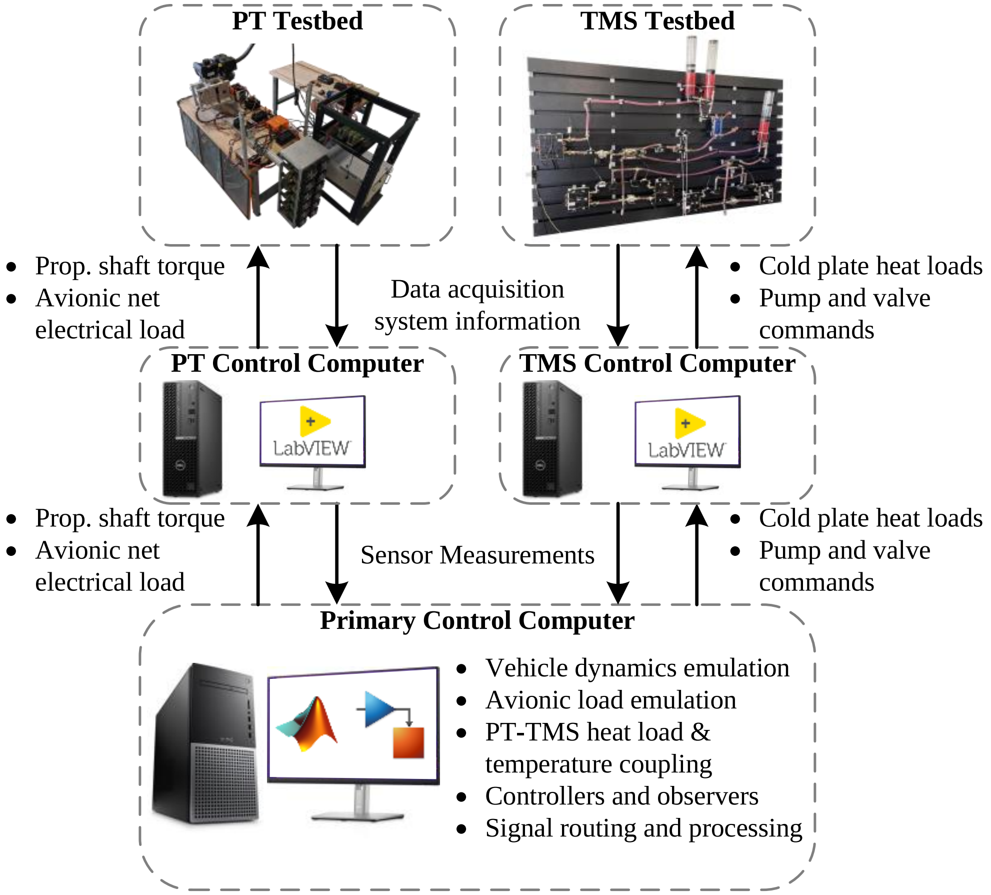


Figure 3.3: Hardware-in-the-loop communication system, illustrating the information passing between the testbeds and control computers.

include the vehicle body dynamics, avionic loads, and heat load coupling.

As mentioned in Section 3.2, the vehicle body dynamics are emulated through a dynamometer motor. The propulsion motor speed is reported to the control computer through the LabView program on the secondary computer. The instantaneous motor speed is passed into a vehicle body simulator that simulates the vehicle speed and computes a propulsion shaft torque, which is communicated back to the LabView program and commanded to the testbed. A similar process is used to emulate the avionic loads, where the key difference is the avionic loads' temperature dependent efficiencies. To emulate the avionic loads, the voltage regulator's voltage and avionic load cold plate temperatures are reported to the control computer via their respective LabView applications. The subsystem voltage and temperatures are used to simulate the avionic loads, from which a net electrical load output is computed. The net electrical load is then communicated back to the LabView application and is commanded to the programmable electronic load on the PT testbed.

In a production system, the powertrain components would be directly connected to a thermal management system to prevent system degradation or thermal runaway. In the IPPTS testbed, the PT and TMS are not connected, so the coupling is emulated through a similar process as the vehicle body and avionic loads. Various current, voltage, speed, and temperature measurements are collected from the testbeds and communicated to the primary control computer through LabView. This data is passed into an experimentally validated powertrain model (Section 3.3.3) to simulate the powertrain losses. It is assumed that the losses are manifested as heat loads that will be imposed on the TMS. Because the PT and TMS testbeds operate at different power levels (Section 3.2.1), each of the losses are scaled by a constant factor before being communicated and imposed as heat loads through the resistive heaters on the TMS testbed. To scale these two systems, the PT heat loads are scaled by constant factors such that the resulting nominal heat load applied to each TMS cold plate is ∼1kW.

3.3 Model Description

3.3.1 Graph-Based Model of an IPPTS

The aircraft IPPTS is mathematically modeled using the graph-based modeling framework outlined in Chapter 2. This modeled was derived by connecting the component graph-based models introduced in [53, 57] according to the system schematic (Figure 3.2). The IPPTS graph model, vertex, edge, input, and parameter information is described by Figure 3.4 and Tables B.1-B.4. This graph model is illustrated in color to highlight the different energy domains and coupling between subsystems. Note that key state and input constraints required for control design are included in Tables B.1 and B.3. The battery's open-circuit voltage curve $V_{ocv}(x_1)$ is provided in [57].

The thermal management system is actuated by pumps that control the quantity of fluid moving through the system. Although each pump is controlled independently with pulse-

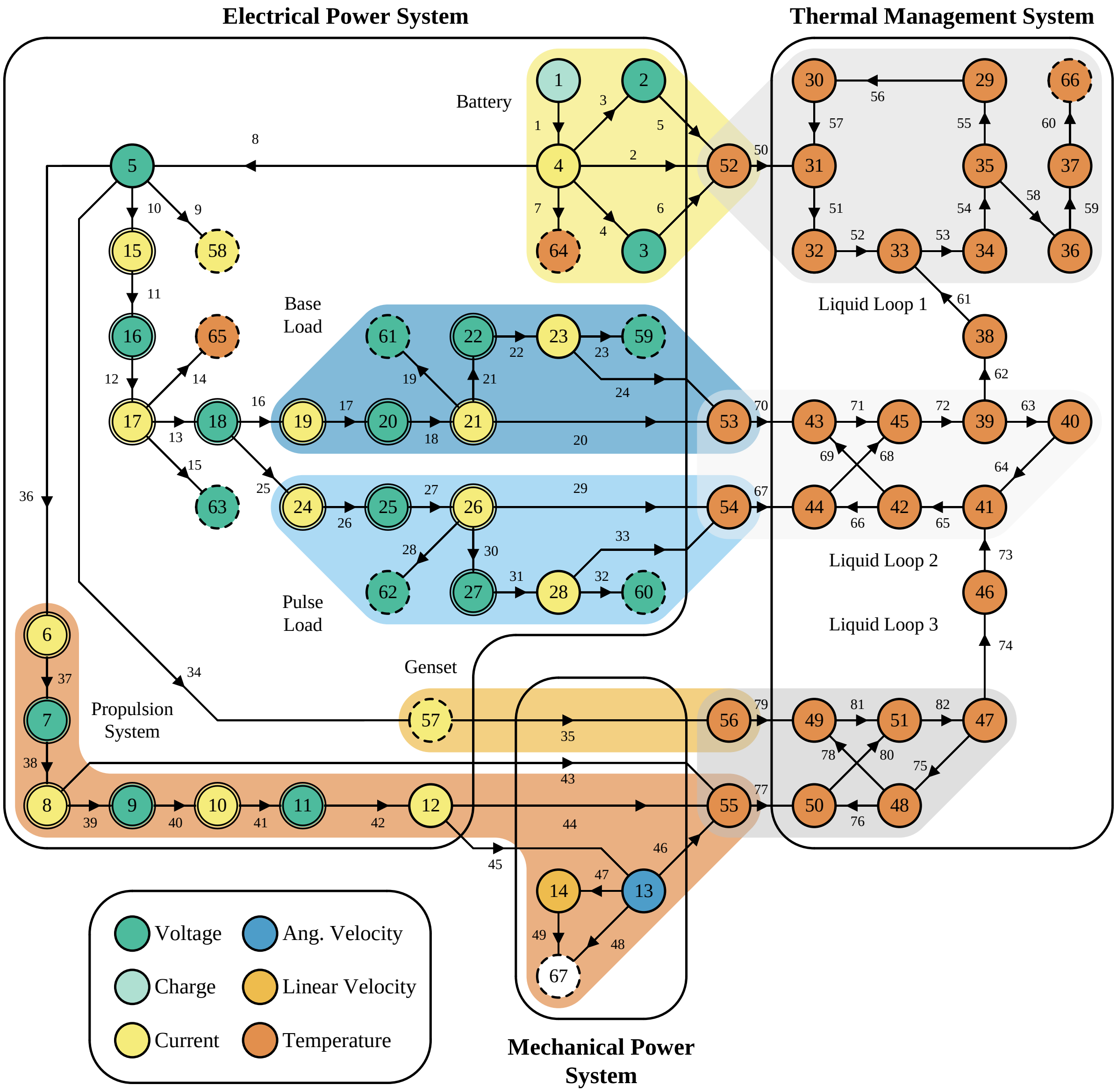


Figure 3.4: Graph-model of the aircraft IPPTS illustrated in Figure 3.1. Component graph models are labeled and colored according to the coloring used in Figure 3.2. Graph vertices are colored to describe the represented energy state.

width modulation (PWM), their outputs are coupled. For example, increasing a pump PWM to increase the flow through the base load flow path will simultaneously decrease the amount of fluid flowing through the pulse load flow path. Because of this coupling, there exists a limited range of flow conditions called a flow envelope. The system's *flow envelope* is the range of achievable flow conditions for a fluid loop. For the IPPTS system, the flow envelope is developed empirically by sweeping across the set of all unique pump

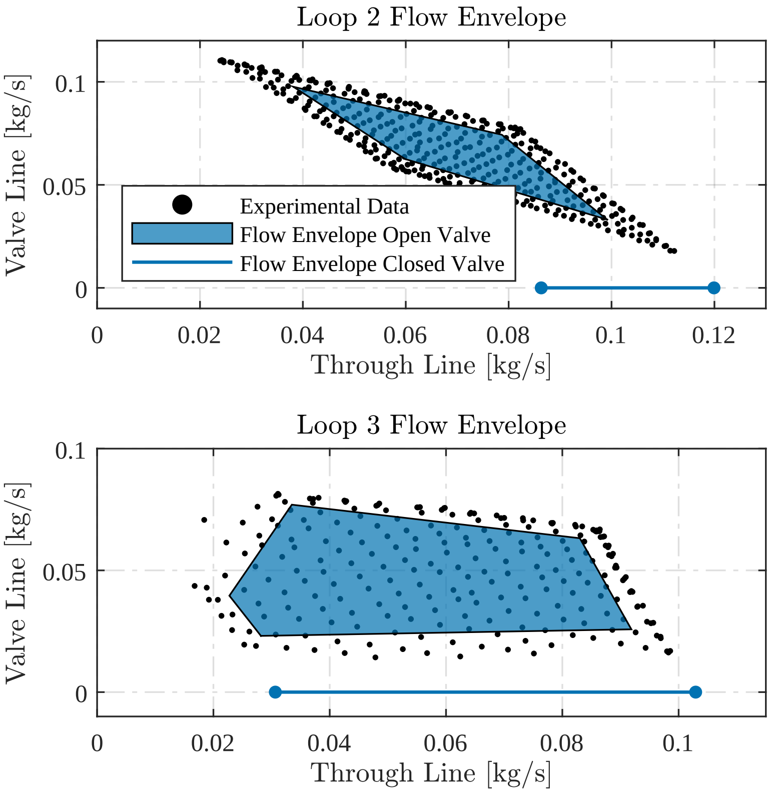


Figure 3.5: Loop 2 (top) and 3 (bottom) flow envelopes for the TMS. The experimental data used to generate identify the flow paths are illustrated by black dots. The *valve line* is the flow rate along the flow path with the valve. The *through line* is the other flow path.

PWM command combinations and recording the resulting flow rates. Figure 3.5 illustrates the flow envelope's for fluid loops 2 and 3 when the valves are both open and closed. When the valve is closed, the flow envelope is a line because fluid will not flow through the valve. The flow envelope for loop 1 is provided in Table B.3.

The avionic loads have a temperature-dependent efficiency (edges 24 and 44 in Table B.2), which is modeled by a temperature-dependent resistance term

$$R_{load}\left(T_{load}\right) = r_{1,load}T_{load} + r_{0,load}, \tag{3.1}$$

where T_{load} is the temperature of either the base or pulse load and $\{r_{1,load}, r_{0,load}\}$ are constant coefficients (Table B.4). The pulse load's temperature-dependent efficiency is illustrated in Figure 3.6 where the load becomes less efficient at high current and temperature. Because

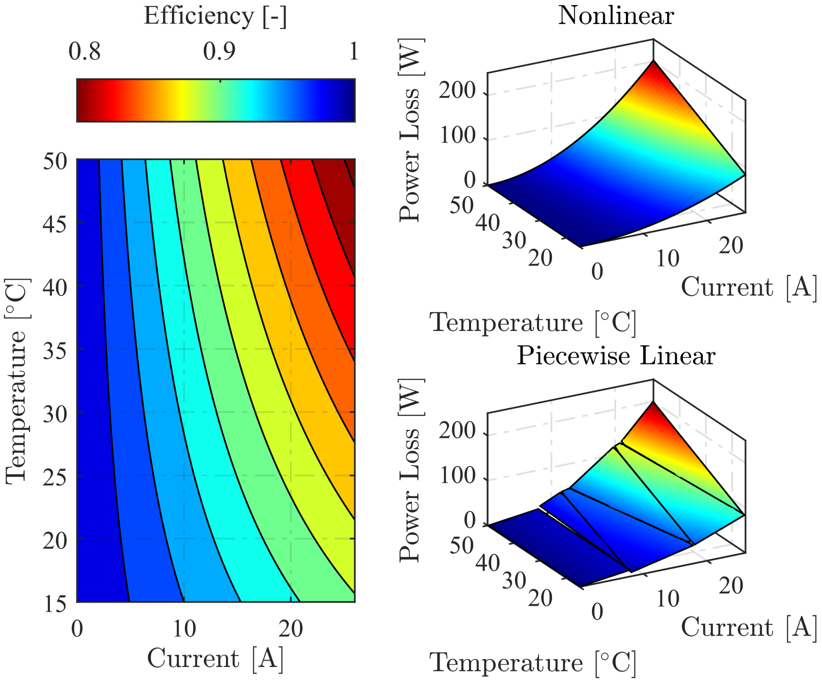


Figure 3.6: An illustration of the pulse load's temperature dependent efficiency and its piecewise linear approximation. The load efficiency varies by up to 20% within the load's safe operating regime.

the nonlinear power flow relationship may degrade the performance of linear predictive controllers (Section 3.4), it is desirable to approximate the power flow using a piecewise linear model, as illustrated in Figure 3.6. The piecewise linear approximation was generated programmatically by solving an optimization problem that minimizes the difference between the nonlinear and piecewise linear surfaces given a prescribed number of operating modes.

3.3.2 Genset Model

As introduced in [57], the genset for this system is modeled outside the graph framework because, due to the testbed's design, it comes with its own embedded controller, which is a black box. This makes it difficult to analyze the underlying system dynamics necessary to formulate a graph-based model. Instead, a system identification approach it utilized to model the DC genset current x^e_{57} as a first-order dynamical system

$$\tau \dot{x}^e_{57} = -x^e_{57} + K u_{13}, \tag{3.2}$$

where K is the input gain and u_{13} is the input command. Because the genset has a 3kW maximum power limit, the maximum current varies as a function of operating voltage x_5. Therefore, the maximum input command must also vary as a function of voltage. An estimated linear bound is found using experimental data

$$\bar{u}_{13} = b_1 x_5 + b_0, \tag{3.3}$$

where $\{b_1, b_0\}$ are constants. The genset's experimentally identified operating regime and current bound is illustrated in Figure 3.7. Note that the genset gain is the slope of approximating surface in Figure 3.7.

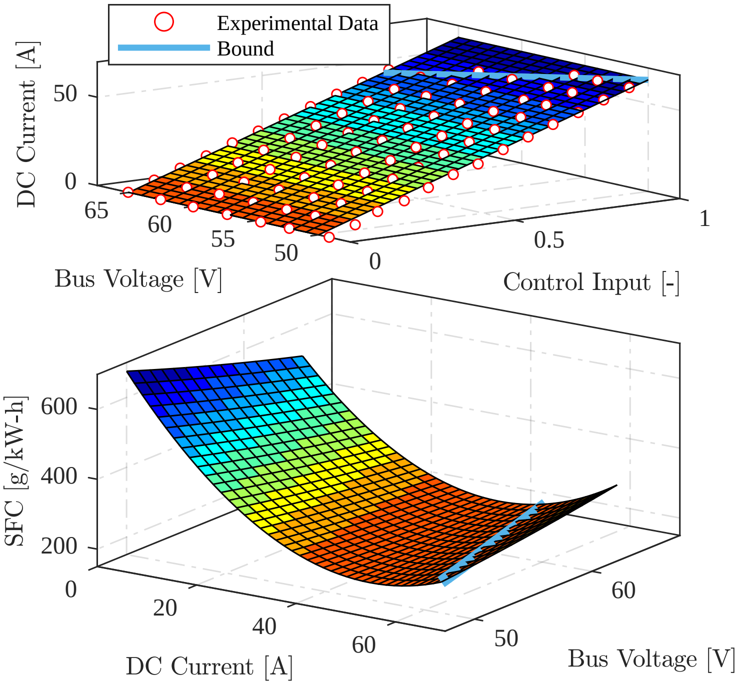


Figure 3.7: Identified genset current (top) and artificial genset SFC curves (bottom) from [57].

A common motivation for electrified aircraft is decreased fuel usage, and a common metric to investigate is a genset's specific fuel consumption (sfc). Adopting the procedure from [57], the genset specific fuel consumption can be estimated as a quadratic function of the genset voltage and current

$$sfc = \begin{cases} a_5 {x_{57}^e}^2 + a_4 x_5^2 + a_3 x_5 x_{57}^e + a_2 x_{57}^e + a_1 x_5 + a_0 & \text{if } x_{57}^e > 0, \\ 0 & \text{if } x_{57}^e = 0, \end{cases} \tag{3.4}$$

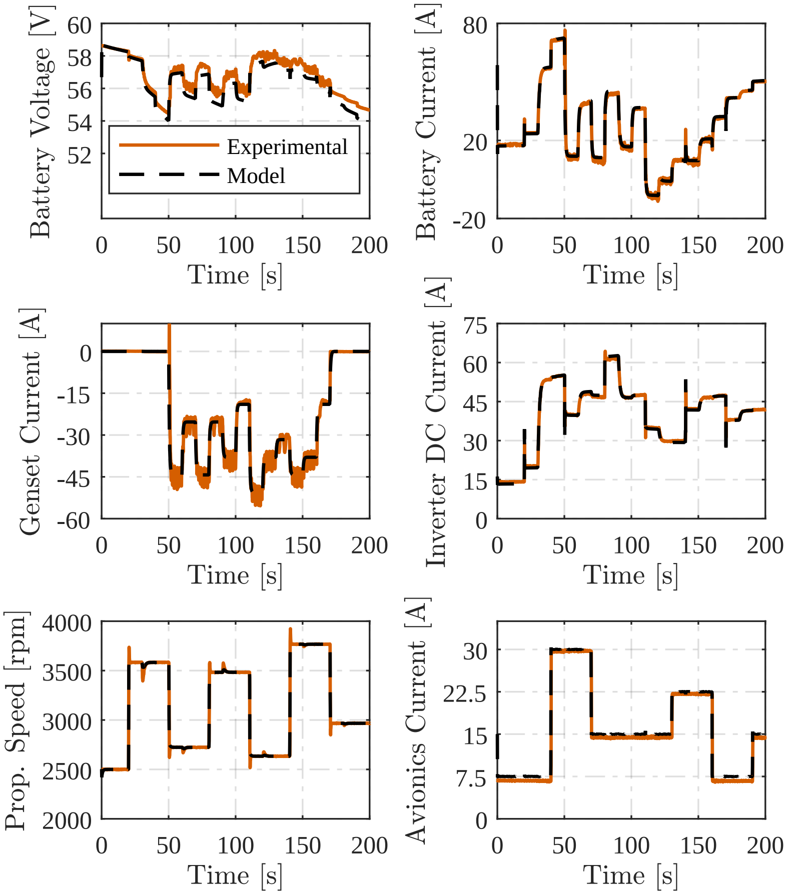
where $\{a_5, a_4, a_3, a_2, a_1, a_0\}$ are constant coefficients. The specific fuel consumption curve is illustrated in Figure 3.7 and all constant coefficient values are tabulated in Table B.4.

3.3.3 Model Validation

The system model is experimentally validated to evaluate its utility in model-based control design. But before the model can be validated, the model parameters are identified through a system identification process. This work uses the nonlinear grey-box model estimation tool from the system identification toolbox in MATLAB, which fits parameters to a structured model. The model is parameterized on a set of training data, and all model parameters are reported in Table B.4.

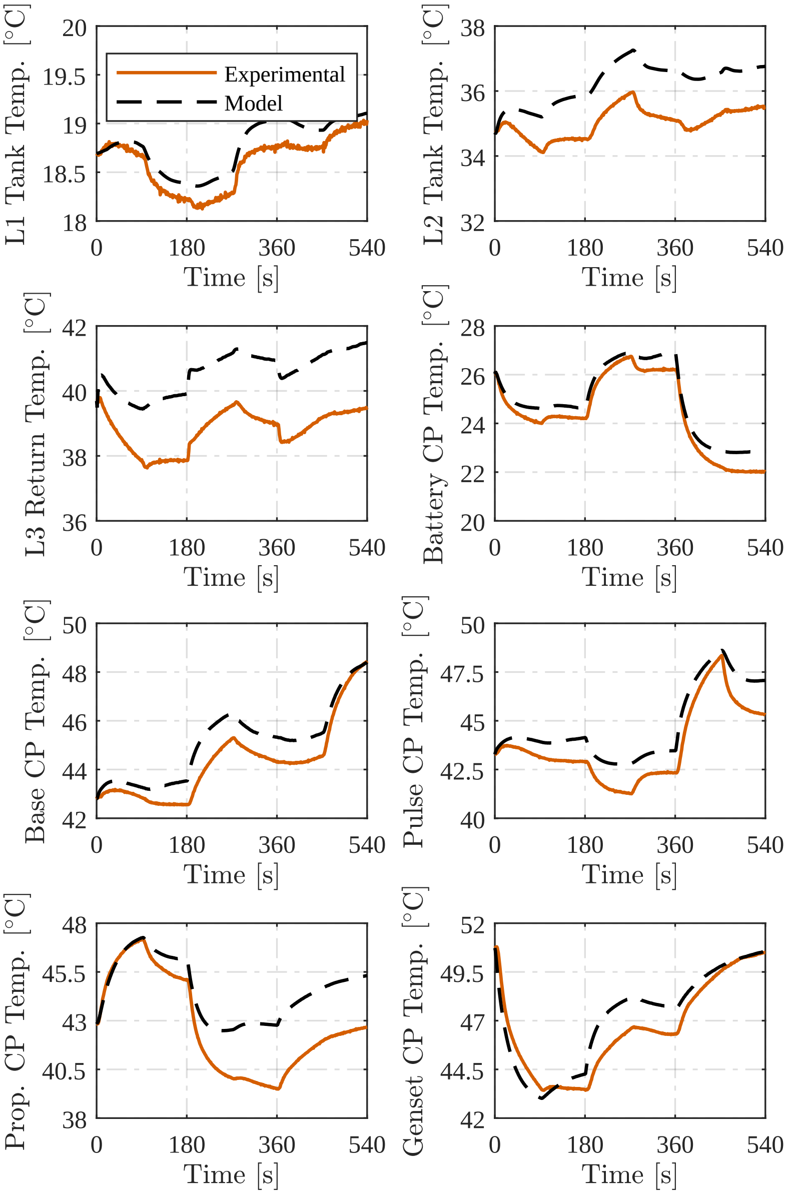
To validate the model, a second set of validation data is collected by exciting the testbed and parameterized model with the same set of inputs/disturbances. Sensor data and model outputs for key states are collected and compared as illustrated in Figure 3.8. Note that the testbeds are experimentally validated independently because they are not physically connected.

The model trajectories deviate some from the experimental results for the battery voltage and temperature states. Additionally, the genset model does not capture the start-up transient behavior at 50s. However, it is sufficient to neglect the start-up transient for control design because it lasts less than 0.5s. Otherwise, the dynamic performance for all other states is good, where both the model and experiment states change similarly when excited by the same stimuli. Overall, the modeling error is small and can be compensated for with closed-loop controllers.



(a) PT experimental system and validated model state comparison from [57].

Figure 3.8: Experimental model validation results for the (a) PT and (b) TMS states. There is sufficient matching between experimental and model data.



(b) TMS experimental system and validated model state comparison.

Figure 3.8 (cont.): Experimental model validation results for the (a) PT and (b) TMS states. There is sufficient matching between experimental and model data. L1, L2, L3 are loop 1, loop 2, and loop 3, respectively.

3.4 Control Design

MPC is a popular control strategy for energy systems because of its ability to respect operational constraints while optimizing a cost function that can be designed to align with the system's figures of merit. However, energy systems are inherently multi-timescale so solving a centralized MPC optimal control problem in real-time can be computationally challenging. To address this problem, a hierarchical MPC strategy is designed for application to an aircraft IPPTS. The following sections introduce hierarchical MPC, its application to aircraft energy management, and a baseline controller that is used for comparison.

3.4.1 Hierarchical MPC

MPC is a control strategy that identifies an optimal control action through the solution of a finite-horizon open-loop optimal control problem. As the name implies, a system model is used to predict the plant behavior for a candidate control sequence that is iterated on through conventional optimization strategies until an optimal control sequence is identified. In addition to optimizing closed-loop performance, MPC is also well known for its ability to handle operational constraints. However, applying traditional centralized MPC to multi-timescale systems can be particularly challenging because the optimization of stiff systems requires both a small discretization interval and long prediction horizon, which increases computation times and inhibits its real-time application on physical systems. The inability to efficiently handle multi-timescale dynamics limits the implementation of MPC on energy systems. For example, the real part of a model's eigenvalues represent the relative timescale of the system dynamics, and the aircraft IPPTS model from Section 3.3 has eigenvalue real parts that span approximately 10 orders of magnitude (Figure 3.9).

Hierarchical MPC addresses this limitation by decomposing a centralized MPC problem into a cascaded structure of smaller, coordinated MPC sub-problems. Distributing the decision-making problems amongst multiple coordinated MPC problems facilitates computational efficient multi-timescale control that is necessary for real-time applications. By coordinating the sub-problems, a hierarchical MPC retains much of the performance benefit

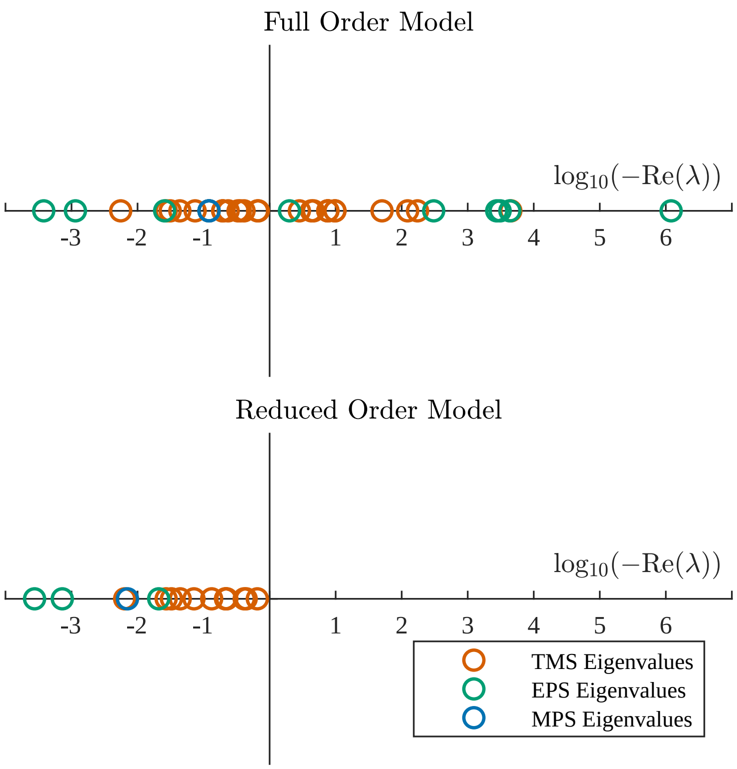


Figure 3.9: An eigenvalue comparison for the full and reduced order model of the aircraft IPPTS model from Section 3.3.

of a centralized MPC. An example hierarchical control architecture is illustrated in Figure 3.10. Each node in Figure 3.10 represents a model predictive controller with unique prediction horizon and update interval tuned to optimize the behavior of a specific aspect of the overall system.

The MPC at the top level of the hierarchy, often called the vehicle level, leverages a model of the entire system and the longest update interval and prediction horizon to plan the behavior of the slowest system dynamics over the course of a mission. Because the vehicle level controller plans the behavior of the entire system, it utilizes the complete system model. To decrease computationally complexity and facilitate multi-timescale control design, a reduced-order model is often used. Because the vehicle level controller is designed to coordinate the slowest system dynamics, the faster system dynamics can be removed using the model order reduction technique described in Section 2.3.

The vehicle level MPC solves and passes optimal trajectory information to the subsequent

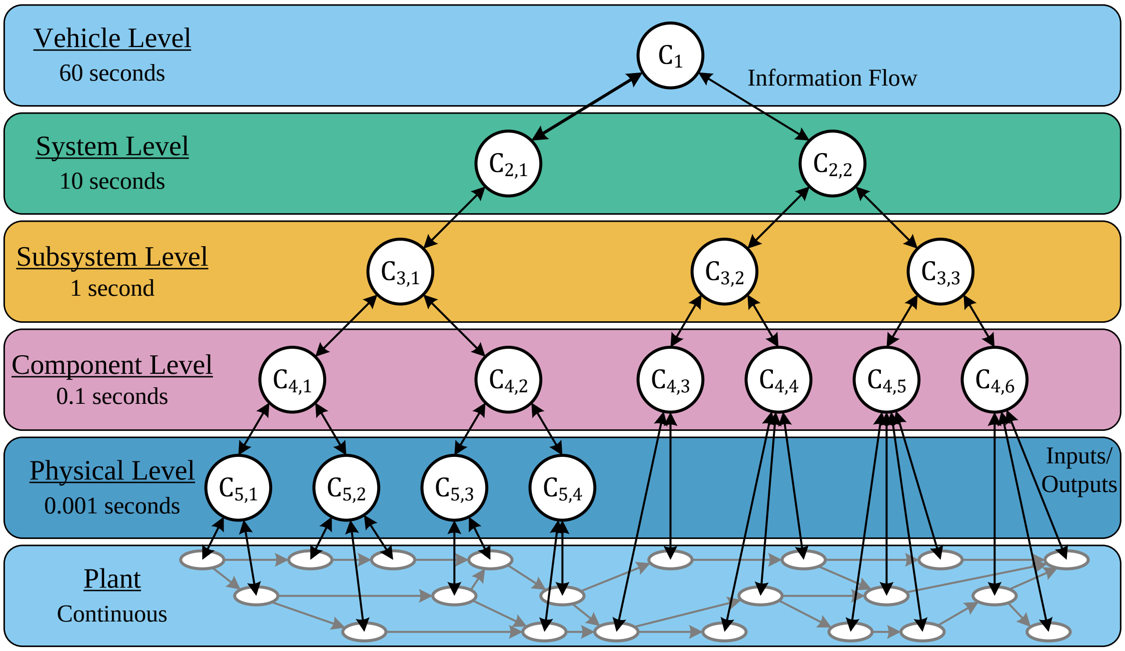


Figure 3.10: A notional 5-layer hierarchical controller from [99].

system, subsystem, component, etc. levels in the hierarchy. In general, upper level controllers in the hierarchy have longer update intervals and prediction horizon to coordinate relatively slower system dynamics, while lower levels controllers have smaller update intervals and prediction horizons to coordinate faster system dynamics. The models integrated into the MPCs below the vehicle-level are a decomposition of the complete system model. The authors of [54, 100], provide model decomposition techniques. For example, in Figure 3.10, there are 2 system level controllers. This hierarchy would require decomposing the complete system model into two separate models, each of which are integrated separately into the two system controllers. This model decomposition happens at each level in the hierarchy. Using this approach, each controller can be specifically tuned to optimize the performance of each part of the overall system. In order to complete the overall control task, upper level controllers pass planning information down the hierarchy to the lower level controllers so that the entire control system is well coordinated.

Each controller in the hierarchy controls a different part of the overall system and therefore has different control and computational requirements. To facilitate computationally efficient multi-timescale control design, upper level controllers leverage reduced-order models of the entire vehicle or systems while lower level controllers utilize higher-order, decomposed models that capture the dynamics of only subsystems and components. The model order reduction technique introduced in Section 2.3 is relevant for hierarchical control design because it allows the system modeler to reduce the state order by on timescale separation. The degree of model reduction is a key design parameter for hierarchical MPC because it provides flexibility in what system dynamics are represented and can be used to make an individual controller more computationally efficient.

3.4.1.1 Model Predictive Control

In a hierarchical MPC, each node represents a model predictive controller tuned for a specific aspect of the overall control task. Each MPC is described by an optimization routine that returns an optimal state and control sequence when solved, and the first element of the control sequence is then applied to the plant. The MPC is solved repeatedly at discrete update intervals in closed-loop with the plant.

While each MPC in the hierarchy is unique with different models, constraints, objectives, etc., each MPC optimization can be written in the same general form. An MPC is characterized by the number of prediction steps N and update interval T where NT is the total prediction time. MPC optimizes a system's dynamics given an initial state x_1, previously applied input u_0 and/or operating mode σ_0, a forecasted disturbance trajectory $\mathbf{d} = \{d_1, \ldots, d_N\}$, and desired trajectories $\mathbf{x^{ref}} = \left\{x_1^{ref}, \ldots, x_{N+1}^{ref}\right\}$ and $\mathbf{a^{ref}} = \left\{a_1^{ref}, \ldots, a_N^{ref}\right\}$, where the subscript represents the discrete time step index. The optimization routine returns optimal model trajectories $\mathbf{x} = \{x_2, \ldots, x_{N+1}\}$ and $\mathbf{a} = \{a_1, \ldots, a_N\}$, optimal control sequence $\mathbf{u} = \{u_1, \ldots, u_N\}$, and optimal mode sequence $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_N\}$. Note that the operating mode σ is generally used for switched linear models (Section 2.4.1), but is generalizable to a single linear model with one mode $N_m = 1$.

The general MPC optimization problem used in this work is

$$
\begin{aligned}
&\text{minimize} && J && \text{(3.5a)}\\
&\text{with respect to} && \mathbf{u}, \boldsymbol{\sigma}, \mathbf{s} && \\
&\text{subject to} && \forall k \in [1:N] && \\
& && \begin{aligned} x_{k+1} &= A_z^\sigma x_k + B_z^\sigma u_k + E_z^\sigma d_k + G_z^\sigma, \\ a_k &= C_z^\sigma x_k + D_z^\sigma u_k + F_z^\sigma d_k + H_z^\sigma, \end{aligned} && \text{(3.5b)}\\
& && s_k \geq 0, && \text{(3.5c)}\\
& && \begin{bmatrix} \underline{x} \\ \underline{a} \end{bmatrix} - s_k \leq \begin{bmatrix} x_{k+1} \\ a_k \end{bmatrix} \leq \begin{bmatrix} \bar{x} \\ \bar{a} \end{bmatrix} + s_k, && \text{(3.5d)}\\
& && \underline{u} \leq u_k \leq \bar{u}, && \text{(3.5e)}\\
& && u_1 = u_0, && \text{(3.5f)}\\
& && m_1 = m_0, && \text{(3.5g)}
\end{aligned}
$$

where J is the cost function to be minimized $\mathbf{s} = \{s_1, \ldots, s_N\}$ is the slack variable sequence, and $m_k \in \mathbb{R}^{N_m}$ is the controller's operating mode defined as

$$
m_{k_i} = \begin{cases} 1 & \text{if } \sigma_k = i, \\ 0 & \text{else}, \end{cases} \tag{3.6}
$$

where the subscript i is the index location of the vector m_k. This MPC problem is subject to 6 constraints. The first constraint (3.5b) is the discrete-time switched linear model dynamics introduced in Section 2.4.1. For the optimization routine to remain feasible, positive slack variables are included to allow the dynamic and algebraic states to temporarily violate their constraints if necessary (3.5c) and (3.5d). Actuator constraints are imposed with (3.5e). In much of the MPC literature [101], it is assumed that the controller computation time is negligible and inputs can be instantaneously applied to the plant. However, when applying MPC to a physical system, there is a time delay between when information is sensed and the control action is applied due to computation times of the observer and controller. To address this delay, the time delay constraints (3.5f) and (3.5g) impose a one-step delay to explicitly account for the computation time. Note that in a conventional MPC, the first element of the control sequence is applied to the plant $u = u_1$, but in this real-time capable MPC, the second element of the control sequence is applied to the plant $u = u_2$.

The MPC optimization problem identified an optimal input, switching, and slack variable sequence that minimizes an objective function

$$J = \sum_{k=1}^{N} \sum_{i=1}^{6} J_i + \sum_{k=1}^{N-1} \sum_{i=7}^{8} J_i + J_9 \tag{3.7a}$$

where,

$$J_{1,k} = \left\| x_k^{ref} - x_k \right\|_{\Lambda_x}^2 \tag{3.7b}$$

$$J_{2,k} = \left\| a_k^{ref} - a_k \right\|_{\Lambda_a}^2 \tag{3.7c}$$

$$J_{3,k} = \| u_k \|_{\Lambda_u}^2 \tag{3.7d}$$

$$J_{4,k} = \| m_k \|_{\Lambda_m}^2 \tag{3.7e}$$

$$J_{5,k} = \| s_k \|_{\Lambda_s}^2 \tag{3.7f}$$

$$J_{6,k} = \| x_{k+1} - x_k \|_{\Lambda_{\Delta x}}^2 \tag{3.7g}$$

$$J_{7,k} = \| u_{k+1} - u_k \|_{\Lambda_{\Delta u}}^2 \tag{3.7h}$$

$$J_{8,k} = \| m_{k+1} - m_k \|_{\Lambda_{\Delta m}}^2 \tag{3.7i}$$

$$J_{9,k} = \left\| x_{N+1}^{ref} - x_{N+1} \right\|_{\Lambda_T}^2 \tag{3.7j}$$

where $\|z\|_{\Lambda}^2 := z^T \Lambda z$ and $\Lambda \geq 0$ are diagonal matrices of weights for each objective. The weights can be set to zero to render specific terms in the objective function inactive. Objectives (3.7b) and (3.7c) are reference tracking objectives. Objectives (3.7d) and (3.7e) minimize the size of the control inputs and operational modes, respectively. Slack variables are included in the control formulation to allow states to violate constraints, but (3.7f) is included to minimize the size of the constraint violations. Objectives (3.7g)-(3.7i) penalize the rate of change of states, inputs, and mode switching and is used to smooth the resulting optimal trajectories. A terminal cost (3.7j) is imposed so that the trajectory ends at a desired state.

3.4.2 Hierarchical MPC for Aircraft Energy Management

The proposed hierarchical MPC for aircraft energy management is illustrated in Figure 3.11. The top three levels of the hierarchy are model predictive controllers, while the bottommost control level is regulatory controllers that handle high frequency state regulation (Section 3.4.4). The controllers are given descriptive names that describe the part of the overall system being coordinated. The air vehicle controller (AVC) coordinates the dynamics of the entire IPPTS and communicates information to the TMS controller (TMSC) and PT controller (PTC) that plan the behavior of the TMS and PT, respectively. The TMSC communicates a planned trajectory to the Loop controllers that coordinate the behavior of each fluid loop, respectively. The system was decomposed in this way based on the spatial distribution of the systems and subsystems. Controller parameters and active objectives are provided in Table 3.2. The following sections provide some additional details on the design of each of the controllers.

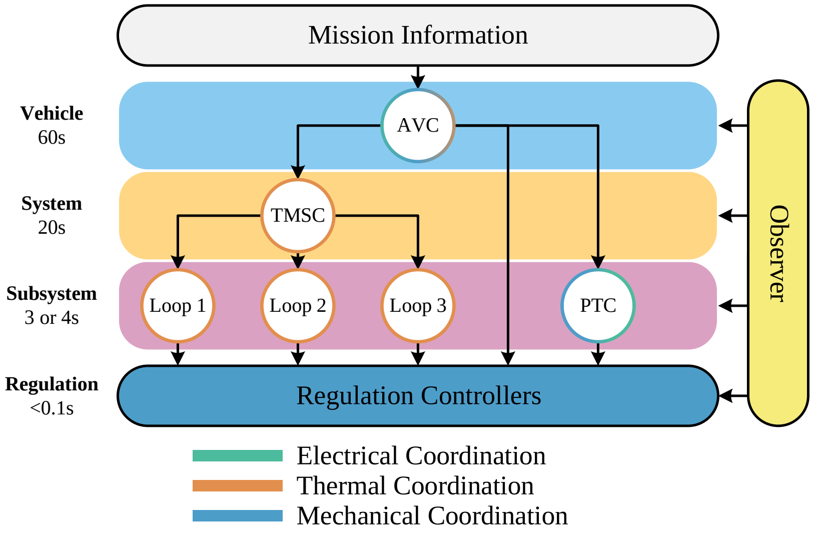


Figure 3.11: A 3-layer hierarchical model predictive controller for an aircraft IPPTS.

Table 3.2: IPPTS hierarchical MPC parameters and active objectives (3.7a).

Controller	Update Interval [s]	Horizon	Prediction Horizon [s]	Active Objectives J_i
AVC	60	7	420	1,3,4,7,8,9
TMSC	20	5	100	1,3,4,7,8,9
Loop 1	4	9	36	1,2,4,6,8,9
Loop 2	4	9	36	1,2,4,6,8,9
Loop 3	4	9	36	1,2,4,6,8,9
PTC	3	11	33	1,2,4,5,6,8,9

3.4.2.1 Air Vehicle Controller

The air vehicle controller optimizes the behavior of the entire vehicle using a reduced-order model of the entire system. Notably, the air vehicle controller selects the energy system architecture by directly commanding the system's discrete actuators: engine on/off mode and the two valve open/close modes. It also selects the vehicle's flight mode (e.g. cruise, loiter, etc.). These decisions are imposed at the top level in the hierarchy because they strongly impact coupling amongst the systems and subsystems. When considering all switched linear dynamics, there are a total of 18,144 potential operating configurations. Directly encoding all 18,144 models associated with the switching modes into the controller would result in excessive computational complexity. Instead, the method from [52] was adopted to decompose the single switching decision into multiple switching decisions that switch smaller partitions of the overall model. The single signal with 18,144 modes was decomposed into 7 signals with 8, 4, 3, 3, 3, 3, and 7 modes, respectively, where each switch impacts flow rates, valve positions, flight conditions, and heat load approximations. This approach encodes all 18,144 modes into the optimization in a more computationally efficient manner.

To optimize over discrete switching decisions, the controller is formulated as a mixed integer quadratic program (MIQP) with an update interval of $T = 60$s and prediction of $N = 7$ steps, which yields 7 minutes of mission preview. The 7 minutes of mission preview was selected assuming that the aircraft's payload would not be operated for more than 6 minutes at a time. The exact update interval and number of prediction steps were tuned to meet the 7-minute requirement and guarantee that the controller's optimization could identify a solution within the update interval.

The temperature constraints are tightened by 2.5°C in the control formulation to account for model error and disturbances. Additionally, the controller incorporates a shrinking horizon, so the mission preview does not extend past the end of the mission. The method from the appendix of [102] is adopted to implement a shrinking horizon.

3.4.2.2 Thermal Management System Controller

The air vehicle controller passes valve position, heat load, and temperature information to the TMSC, which coordinates the thermal management system over a 100s prediction horizon ($T = 20$s and $N = 5$). To retain model accuracy over the prediction horizon, the thermal management system is represented by a switched linear model with 72 unique operating modes. To optimize over the 72 discrete operating modes, the controller is formulated as a MIQP. The time step and horizon were tuned so that the optimization could be solved within one controller update interval. Similar to the AVC, all temperature constraints are tightened by 2.5°C to account for model error and disturbance uncertainty.

3.4.2.3 Thermal Management System Loop Controllers

The thermal management system comprises three fluid loops, each with its own MPC. Each loop controller is tuned identically, but the model integrated into each controller differs. For example, the loop 1 controller uses the model of only loop 1 and relies on coordination from the TMSC to account for the coupling with fluid loop 2. The loop controllers control the flow rates through the system.

Each controller has a 36s prediction horizon ($T = 4$s and $N = 9$). Because the update interval and prediction horizon is relatively short, a single linear model is sufficiently accurate for solving the MPC problem. The single linear model loop controllers contrast the AVC and TMSC, in which a switched linear model with multiple operating modes was leveraged to capture nonlinearities over a longer prediction horizon. The loop controllers are formulated as a linear quadratic program (QP) where the control model is linearized at each controller update around the system's current operating state. The update rate for the loop controllers was based on the convergence rate of the pump regulation controllers (Section 3.14), which was empirically determined to be approximately 1.0s.

3.4.2.4 Powertrain Controller

Using temperature and genset coordination from the AVC, the PTC optimizes the behavior of the powertrain states by selecting an optimal propulsion motor speed, genset power output, and avionic load power. The MPC is formulated as a QP with an $T = 3$s update interval and $N = 11$ prediction steps (33s prediction horizon). The update interval was selected to be slower than the convergence rate of the propulsion motor speed regulator (Section 3.14), which was empirically determined to be approximately 0.5s.

3.4.3 Baseline Control

A baseline controller (Figure 3.12), which is designed to compare against the hierarchical MPC, is based on conventional approaches for aircraft energy management in the literature [103, 104]. As summarized in Table 3.3, there are 9 PI controllers, 2 P load shedding controllers, and 4 logic-based controllers. The vehicle speed, avionics voltage, base load current, and pulse load current controllers are all PI controllers that track a reference trajectory specified by the mission profile.

The 5 PI pump controllers command flow rates based on temperature set points that are

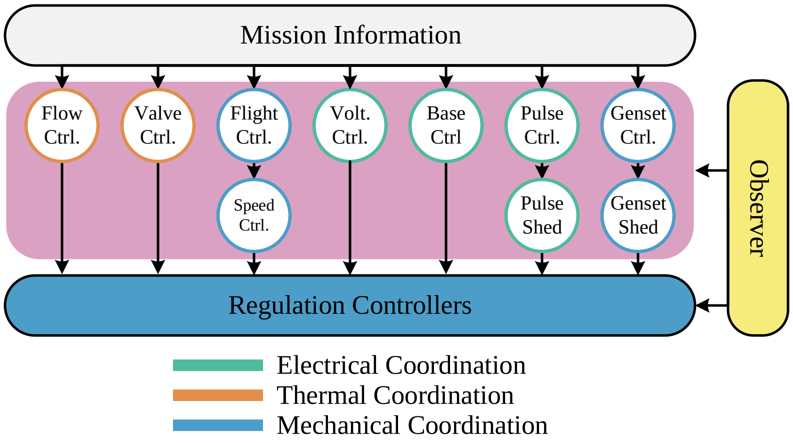


Figure 3.12: The baseline controller for the aircraft IPPTS.

Table 3.3: The IPPTS baseline controller information. Variables reference Tables B.1-B.4.

Actuator	Measurement	Type	Setpoint/Threshold
Inverter u_4	Vehicle Speed x_{14}	PI	Variable
LV Converter u_1	Avionics Voltage a_{18}	PI	40V
Base Converter u_2	Base Load Current x_{23}	PI	Variable
Pulse Converter u_3	Pulse Load Current x_{28}	PI	Variable
Loop 1 Flow u_6	Return Temp. u_{34}	PI	35°C
Loop 2 Through Line Flow u_7	Base Temp. x_{53}	PI	40°C
Loop 2 Valve Line Flow u_8	Pulse Temp. x_{54}	PI	40°C
Loop 3 Through Line Flow u_{10}	Prop. Temp. x_{55}	PI	45°C
Loop 3 Valve Line Flow u_{11}	Genset Temp. x_{56}	PI	45°C
Genset Command u_{13}	Net Load $a_6 + a_{15} + x^e_{58}$ Battery SOC x_1	Logic	Variable
Loop 2 Valve	Pulse Load Current u_3 Pulse Temp. x_{54}	Logic	43°C
Loop 3 Valve	Genset Current x^e_{57} Genset Temp. x_{56}	Logic	48°C
Pulse Load Shed	Max. Loop 2 Temp.	P	43°C
Genset Load Shed	Max. Loop 3 Temp.	P	48°C
Speed Reference	Battery SOC x_1	Logic	32%

5°C lower than the respective subsystems upper temperature limit [103]. Because fluid loops 2 and 3 have multiple flow paths, the loop 2 and 3 flow controllers must account for the flow envelopes (Figure 3.5). As shown in Figure 3.13, each path's flow rate is controlled by its own flow PI controller. The two desired flow rate inputs are saturated by the 2-dimensional saturation function that identifies a feasible flow rate combination that is closest to the desired point. For a convex flow envelope, the closest point problem is solved by projecting the unsaturated control action onto the flow envelope. In this work, the projection was computed using the *project()* function in MPT3 [105].

The genset controller, which decides the ratio of power provided by the genset and battery, is formulated as a state machine. This work uses the precise genset controller introduced in [57]. Notably, the genset control design is charge sustaining, thus biasing the system to

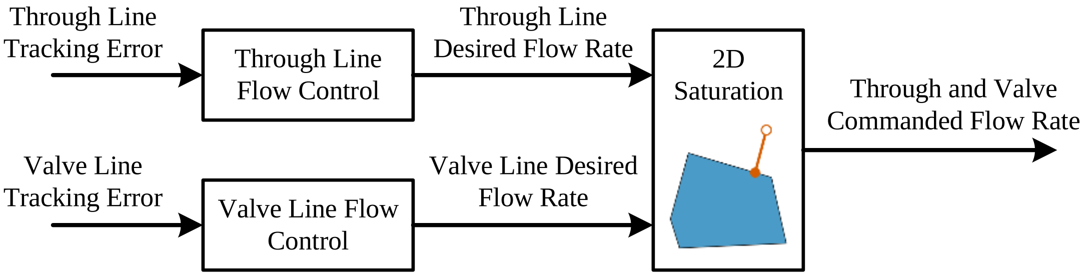


Figure 3.13: The control architecture for the MIMO pump controllers for TMS loop 2 and 3. The 2-dimensional saturation is applied by projecting the desired flow rate onto the set of feasible flow rates.

aggressively use the genset to recharge the battery if possible. Additionally, it is equipped with various emergency handling logistics to prevent the battery from under or over-charging and unsafe rates.

Two logic controllers operate the on/off position of the valves in the TMS fluid loops. Both controllers follow the same logic; if the pulse load or genset is active, the respective valve opens. If the pulse load or genset is not active and the respective load temperature decreases below a prescribed threshold, then the respective valve closes. Otherwise, the valve position remains in its current state. The temperature thresholds are listed in Table 3.3.

Two load shed controllers are designed to prevent the system from overheating. If temperatures in fluid loops 2 or 3 are too high, the load shed controllers artificially decrease the pulse load and genset commands to decrease the heat generation. The load shedding is proportional to temperature difference between the load and its constraints. Note that the P controller is designed to activate only if the maximum loop temperature exceeds the threshold listed in Table 3.3. Shedding load has undesirable consequences. If the pulse load is shed, the aircraft does not complete its energy objectives. If the genset is shed, the battery may deplete its state of charge. If the battery state of charge is depleted while in an unfavorable flight condition, a speed reference controller will return the vehicle to a flight mode better suited to recharge the battery.

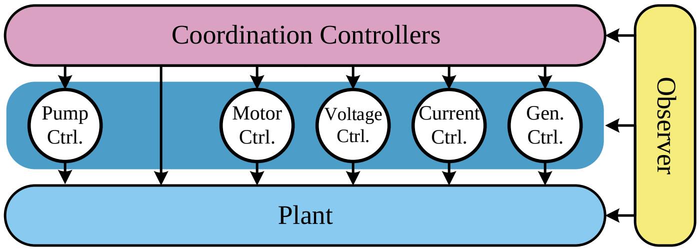


Figure 3.14: The embedded controllers that interface the hierarchical or baseline coordination controllers with the plant.

3.4.4 Embedded Controllers

While some control decisions can be directly applied to the plant (e.g. the valve position), some of the system dynamics are too fast for either the MPC or baseline controller to directly control (e.g. motor controllers can have an update interval in the MHz to GHz range). To control the fast states, a regulation layer of embedded controllers is incorporated into the complete control design (Figure 3.14). Because the embedded control systems are not published by the manufacturer, the precise control formulation is unknown. However, it is expected that they are PI-like controllers that choose a PWM input based on a user-defined setpoint. The system has 5 types of embedded regulation controllers: pump, motor, voltage, current, and generator controllers, which require a flow rate, prop. speed, DC voltage, DC current, and DC current setpoints, respectively. Because the generator is also used to start the engine, the generator controller also handles the startup condition and transition from engine startup mode to power generation mode.

3.5 Observer Design

The controllers introduced in the previous section require full state feedback to compute a control action. However, the experimental testbeds are not outfit with enough sensors to directly measure all the states. Additionally, sensor measurements are often noisy, which negatively impacts the control performance. Therefore, this work uses a decentralized observer design to both estimate unmeasured state values and reduce measurement noise.

The decentralized observer is illustrated in Figure 3.15. A decentralized structure was

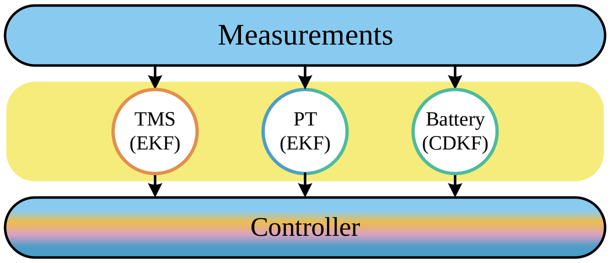


Figure 3.15: The decentralized observer for the aircraft IPPTS.

selected because it can be more computationally efficient for real-time implementation, while affording increased flexibility for observer design and tuning. As shown in Figure 3.15, the observer comprises 3 subsystem observers. The thermal management system and powertrain states (excluding the battery) are estimated using an extended Kalman Filter (EKF). It has been shown, in [85], that an EKF is sufficient for thermal management system and powertrain state estimation. However, battery state estimation is often more challenging, so a central difference Kalman Filter (CDKF) was selected to estimate the battery states [106, 107].

Discrete-time system models are required to apply the EKF and CDKF estimation technique. The nonlinear system model (Section 3.3) is discretized using a Forward-Euler approach and is put in the following form

$$x_k = f\left(x_{k-1}, u_{k-1}, d_{k-1}, w_{k-1}, \right), \tag{3.8a}$$

$$y_k = h\left(x_k, u_k, d_k, v_k, \right), \tag{3.8b}$$

where x is the system states, y is the measurements, u are inputs, d are known disturbances, and $w_K \sim N\left(0, Q\right)$ and $v_K \sim N\left(0, R\right)$ are assumed to be normally distributed process and sensor noise with covariance Q and R, respectively. Note that measured algebraic state measurements are included in y_k. The error system covariance is described by P. In practice, both an initial guess for state $\hat{x}_{0|0}$ and error covariance $P_{0|0}$ is required. An initial guess determined using system knowledge should be supplied for the initial state, and it is common practice to initialize the error covariance as the identity matrix. The noise covariance matrices Q and R are treated as tuning parameters that represent the relative degree of trust in the model and sensors, respectively. The notation $z_{i|j}$ represents the value of the signal z at time index i given data from time index j.

3.5.1 Extended Kalman Filter

The EKF is an extension of the traditional linear Kalman Filter to nonlinear systems, where the nonlinear system is linearized and the linear Kalman Filter estimation process is applied to the linearized system. The Kalman filtering process comprises prediction and correction steps. The prediction step uses the system model (3.8) to predict the system state and update the error covariance matrix

$$A = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1}, u_{k-1}, d_{k-1}, w=0}, \tag{3.9a}$$

$$\hat{x}_{k|k-1} = f\left(\hat{x}_{k-1}, u_{k-1}, d_{k-1}, 0, \right), \tag{3.9b}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q. \tag{3.9c}$$

Equation (3.9b) estimates the system state $\hat{x}_{k|k-1}$ using the nonlinear system model and previously applied system inputs and disturbances. If the model was perfect, additional computations would be unnecessary. However, model error and uncertainty is always present. To compensate for this error, sensor information is leveraged within the correction step. In this step, the measurement model is linearized to compute the observer gain L_k, which is used to correct the state estimate.

$$C = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}, u_k, d_k, v=0}, \tag{3.10a}$$

$$L_k = P_{k|k-1}C^T \left(CP_{k|k-1}C^T + R\right)^{-1}, \tag{3.10b}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k \left(y_k - h\left(\hat{x}_{k|k-1}, u_k, d_k, 0\right)\right), \tag{3.10c}$$

$$P_{k|k} = \left(I - L_k C\right) P_{k|k-1}, \tag{3.10d}$$

where y_k are the sensor measurements and I is the identity matrix. The state estimate $\hat{x}_{k|k}$ is passed to the controller for feedback.

3.5.2 Central Difference Kalman Filter

The CDKF is a variant of sigma-point Kalman filter approaches [106] with only one tuning variable. In contrast to the EKF, the CDKF retains the full nonlinear form of the system model to produce better state and covariance estimates. However, the CDKF's computational complexity hinders its application to higher dimensional systems for real-time applications. While the EKF uses a linearized system to estimate the covariance matrix (3.9c), the CDKF uses multiple function evaluations at carefully selected sigma-points to approximate the covariance matrix. An interested reader is referred to [106] for additional details on sigma-point filtering methods.

Sigma-point filtering methods are described by weighting constants γ, $\alpha^{(m)}$, and $\alpha^{(c)}$ that weigh the impact of each sigma-point on the estimated statistics. The CDKF constants are described by,

$$\gamma = g \tag{3.11a}$$

$$\alpha_0^{(m)} = \alpha_0^{(c)} = \frac{g^2 - n}{g^2} \tag{3.11b}$$

$$\alpha_j^{(m)} = \alpha_j^{(c)} = \frac{1}{2g^2} \tag{3.11c}$$

where n is the state vector dimension ($x \in \mathbb{R}^n$) and $j \in [1 : 2n]$. Note that $\gamma \in \mathbb{R}$ is a scalar and $\alpha^{(m)} \in \mathbb{R}^{2n+1}$ and $\alpha^{(c)} \in \mathbb{R}^{2n+1}$ are vectors. The variable g is the single tuning parameter for the CDKF.

When applying the CDKF, the first step is to compute the sigma points, $\mathcal{X} \in \mathbb{R}^{n\times(2n+1)}$

$$\mathcal{X}_{k-1,k-1} = \left[\hat{x}_{k-1|k-1} \quad \hat{x}_{k-1|k-1} + \gamma\sqrt{P_{k-1|k-1}} \quad \hat{x}_{k-1|k-1} - \gamma\sqrt{P_{k-1|k-1}}.\right] \tag{3.12}$$

It is recommended to compute the matrix square root $\sqrt{P_{k-1|k-1}}$ using Cholesky factorization. Note that $\hat{x}_{k-1|k-1} \pm \gamma\sqrt{P_{k-1|k-1}}$ is a matrix where each column of $\sqrt{P_{k-1|k-1}}$ is added to or subtracted from $\hat{x}_{k-1|k-1}$.

Like in the EKF, a prediction step is applied to estimate the system state and update the covariance matrix

$$\mathcal{X}_{j,k|k-1} = f\left(\mathcal{X}_{j,k-1|k-1}, u_{k-1}, d_{k-1}, 0\right), \tag{3.13a}$$

$$\hat{x}_{k|k-1} = \sum_{j=0}^{2n} \alpha_j^{(m)} \mathcal{X}_{j,k|k-1}, \tag{3.13b}$$

$$P_{k|k-1} = \sum_{j=0}^{2n} \alpha_j^{(c)} \Delta_x \Delta_x^T + Q, \tag{3.13c}$$

where $\Delta_x = \mathcal{X}_{j,k|k-1} - \hat{x}_{k|k-1}$ and the notation $Z_{j,m|n}$ is the j^{th} column of the matrix $Z_{m|n}$. The current state estimate is likely inaccurate due to model uncertainty. To correct the estimate, an estimated system output $\hat{y}_k$ is computed and compared to the sensor information. The output estimate is generated by passing the sigma-points through the measurement model,

$$\mathcal{Y}_{j,k|k-1} = h\left(\mathcal{X}_{j,k|k-1}, u_k, d_k, 0\right), \tag{3.14a}$$

$$\hat{y}_{k|k-1} = \sum_{j=0}^{2n} \alpha_j^{(m)} \mathcal{Y}_{j,k|k-1}, \tag{3.14b}$$

where $\mathcal{Y}$ are the output sigma-points. The estimated system state and covariance matrix are updated in the correction step

$$U_k = \sum_{j=0}^{2n} \alpha_j^{(c)} \Delta_y \Delta_y^T + R, \tag{3.15a}$$

$$V_k = \sum_{j=0}^{2n} \alpha_j^{(c)} \Delta_x \Delta_y^T, \tag{3.15b}$$

$$L_k = V_k U_k^{-1}, \tag{3.15c}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k\left(y_k - \hat{y}_{k|k-1}\right), \tag{3.15d}$$

$$P_{k|k} = P_{k|k-1} - L_k U_k L_k^T, \tag{3.15e}$$

where $\Delta_y = \mathcal{Y}_{j,k|k-1} - \hat{y}_{k|k-1}$. The state estimate $\hat{x}_{k|k}$ is passed to the controller for feedback.

3.5.3 Observer Tuning

The battery states are estimated using a CDKF with a $T = 0.125$s update interval and $g = 1.2$ tuning parameter. A CDKF is used to estimate the battery states because it is easy to tune and has been shown to be an effective battery state estimator in [106]. Because the battery has few states, the added computational complexity is minimal. The battery voltage is sensed as an output and the battery current is treated as a model disturbance.

The remaining powertrain states are estimated with an EKF with $T = 0.125$s update interval. The EKF was chosen for its scalability and ease of implementation. Additionally, it was shown in [57] that an EKF is sufficient for state estimation for this particular platform. Various system voltages, currents, speeds, and component temperatures are passed to the EKF as output measurements and disturbance information.

Similarly, the TMS states are estimated with an EKF with $T = 1.0$s update interval and was chosen based on its application in prior demonstrations. The slower update of the TMS EKF reflects the slower dynamics of the thermal management system in comparison to the faster dynamics of the powertrain. Measured system temperatures and mass flow rates were input into the EKF.

3.6 Control Results

The hierarchical and baseline controllers are designed and verified in simulation prior to validation on the experimental test platforms. They are evaluated on performance, efficiency, and reliability figures of merits. The following sections introduce the mission used for comparison, highlight key operational differences, and demonstrate quantifiable improvements in control behavior.

3.6.1 Mission Objectives

The aircraft's mission objectives are illustrated in Figure 3.16 and consist of a desired flight speed and avionic load current requirements. The mission considers a scenario where the aircraft cruises (high speed) to three separate locations to activate its pulsed power payload

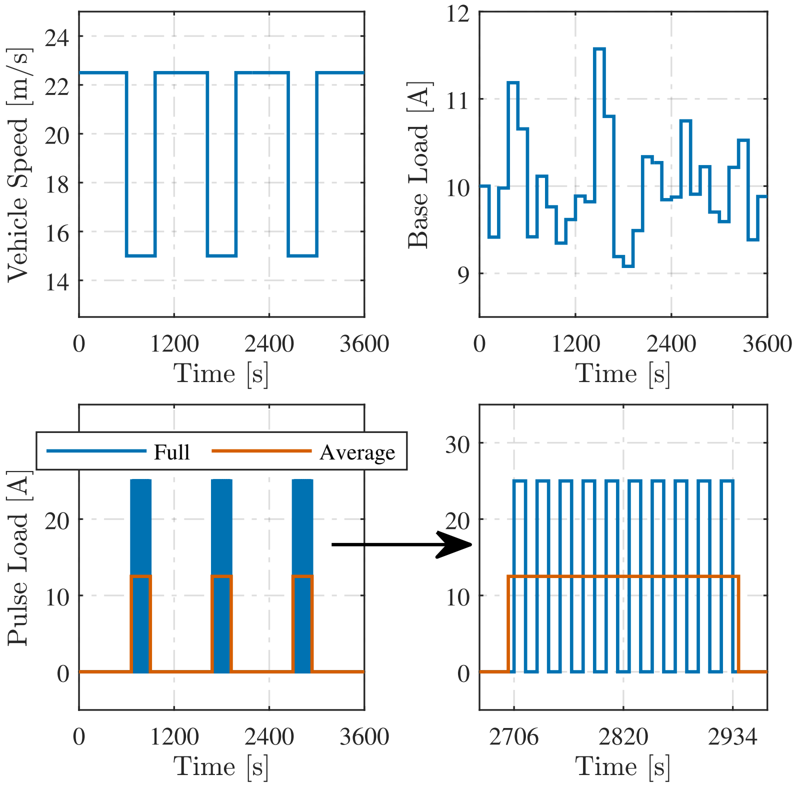


Figure 3.16: The flight and avionic load profiles for the control design evaluation case study.

during a loiter (low speed). The pulse payload power requirement is a pulse train comprised of ten evenly spaced 12s pulses (12s pulse uptime followed by 12s pulse downtime). These three mission objectives are the only time varying references passed to the controllers. Because the hierarchies AVC has a 60s update interval, an average valued variation of the pulse load is supplied to the controller (Figure 3.16). This mission assumes that the genset noise and vibration interferes with the pulsed payload equipment. Therefore, the aircraft cannot utilize the genset while the pulse load is active. Other key state and operational constraints are listed in Tables B.1 and B.3.

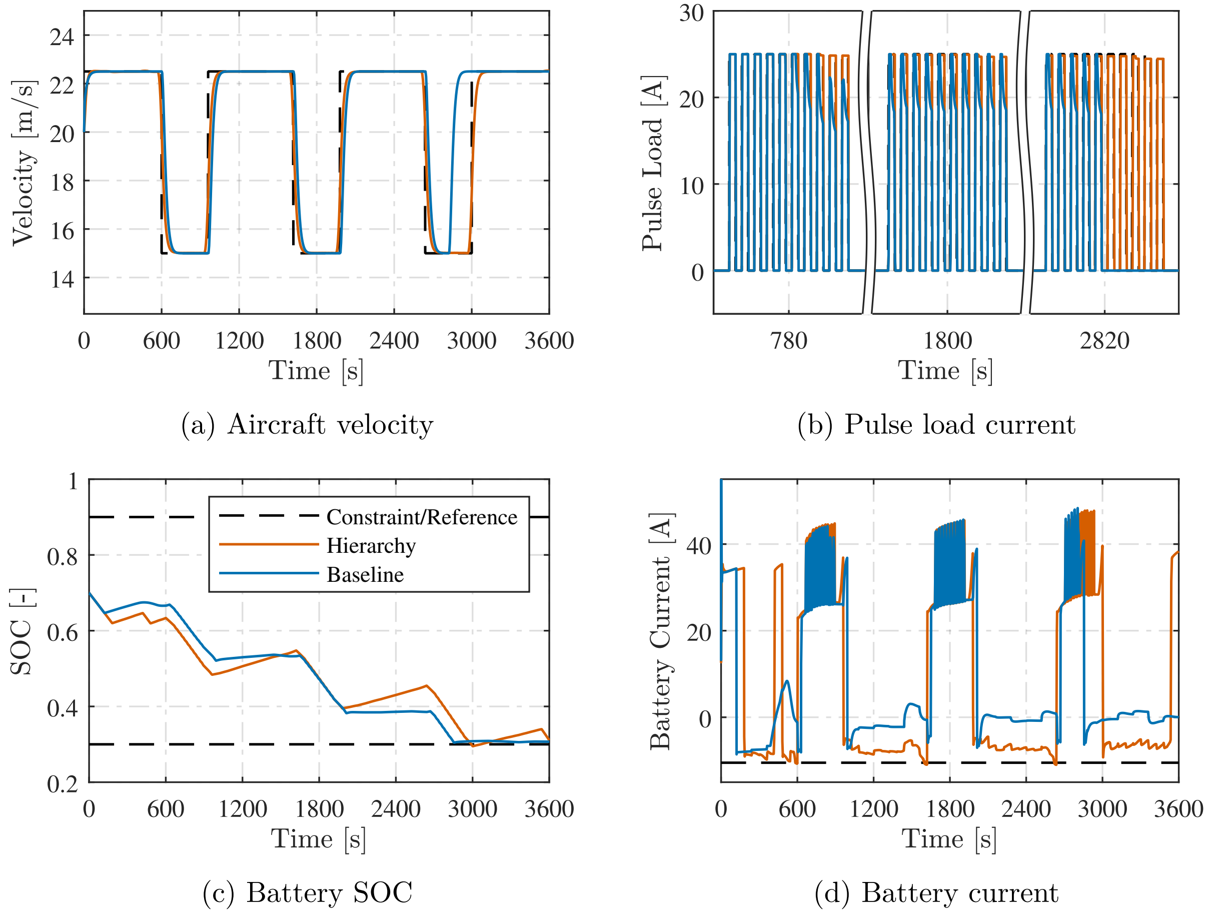
The mission objectives are classified as flight and mission-critical. The flight critical objectives include supplying power to the propulsion system and the base load, which are required to sustain flight. The mission-critical objectives are optional and can be partially completed or entirely neglected at the cost of a mission failure. The mission-critical objectives include deploying the system's pulsed payload and activating the genset. The genset is used to sustain the battery state of charge and is controlled to support the mission objectives. Because the genset power output can be changed to meet mission requirements, it is treated as a mission-critical load.

The predictive controllers were formulated using YALMIP [108] and solved using CPLEX [109] (for MIQPs) and QUADPROG (for QPs). The controllers were experimentally validated on the control computer (Figure 3.3) with an AMD Threadripper CPU and 128 GB RAM. The hardware in the loop system configuration was described in Section 3.2.2. The predictive controllers were run as parallel processes in Simulink, which allows each controller to compute its solution simultaneously for real-time operation.

3.6.2 Simulation Results

The simulated hierarchical and baseline control results are illustrated in Figures 3.17 and 3.18. Over the course of the entire mission, the baseline controller routinely sheds the pulse load in order to respect the power and thermal system constraints. Notably, at the end of the mission near 2800s, the pulse load is completely shed (Figure 3.17b) because the flight controller returns the aircraft to its cruise condition to recharge the battery (Figure 3.17a). Although the genset controller was tuned to be charge sustaining, the battery energy was mostly depleted before the last pulse train (Figure 3.17c). Inspecting the battery current (Figure 3.17d), the genset was attempting to recharge the battery during each cruise segment, but failed to recharge the battery enough. The baseline controller is limited by its thermal performance. During the pulse trains, the pulse temperature approached its temperature limit within 0.5°C (Figure 3.18c), resulting in load shed. Similarly, during the cruise segments the genset routinely operates near its thermal limit (Figure 3.18e), so the load shed controller decreases its power output and the genset is unable to sustain battery SOC. Ideally, the flow rates through each cold plate could be increased to transfer more thermal energy. However, as illustrated by Figure 3.18e, the flow rates were already at or near saturation when operating at high temperatures. The only alternative to respecting constraints is to shed the electrical system load to decrease heat generation. Although the pulse and genset temperatures are high, the load shed controllers were effective in preventing constraint violations. However, throughout the mission the base load, which is flight critical and cannot be shed, experienced temperature constraint violations.

In contrast, the hierarchical controller leverages a prediction model to explicitly account for dynamic coupling and constraints while supplying power to both the mission and flight critical loads. The hierarchy was able to follow both the mission velocity profile (Figure 3.17a) and deliver the desired current to the pulse load (Figure 3.17b), albeit a small amount of pulse load was shed towards the end of the third pulse train. The hierarchy delivered power to the PT loads because the genset adequately recharged the battery during each of the cruise segments (Figures 3.17c and 3.17d). Unlike the baseline controller, the hierarchical controller was not thermally limited and did not exhibit any significant thermal constraint violations.



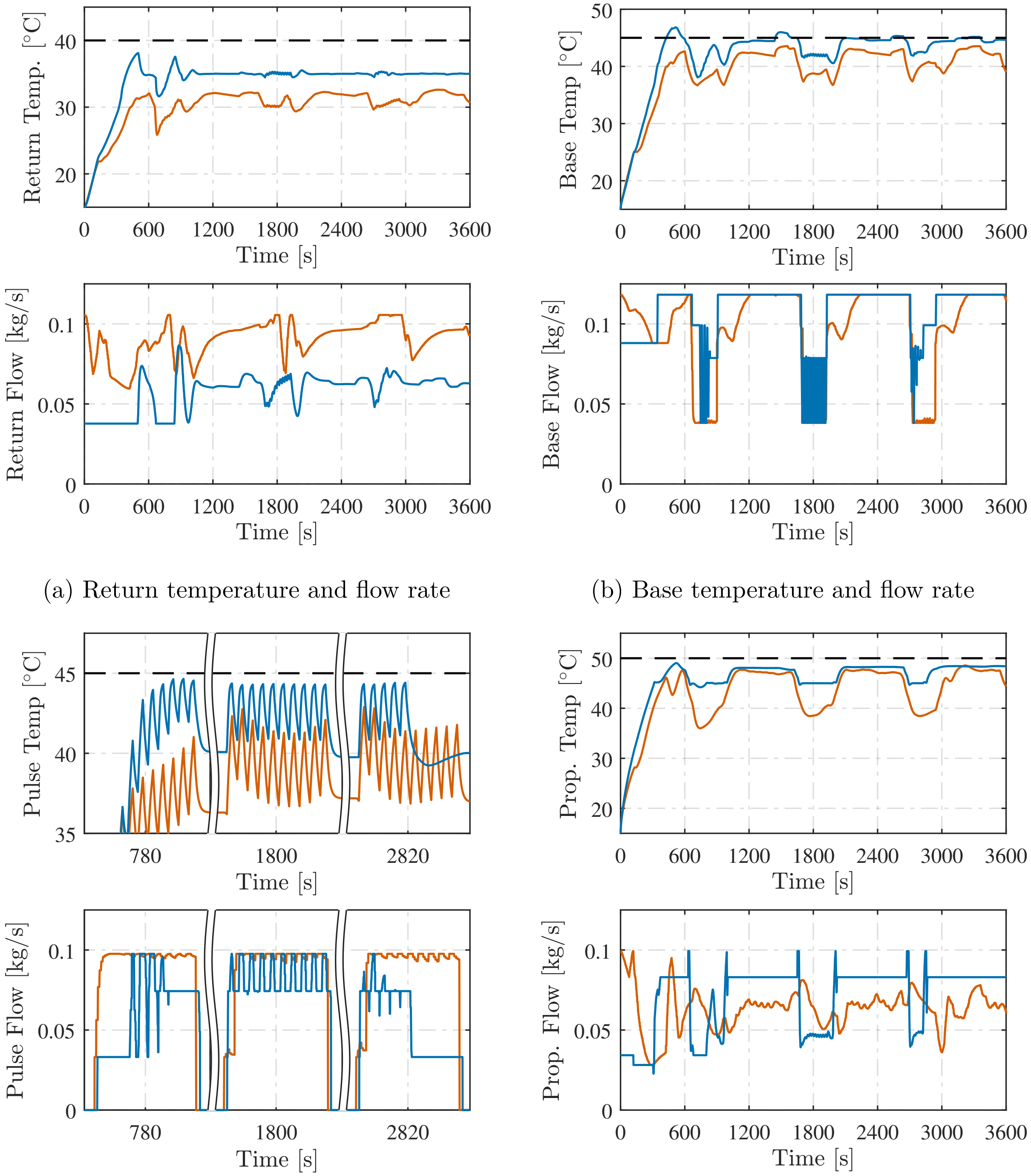
(a) Aircraft velocity

(b) Pulse load current

(c) Battery SOC

(d) Battery current

Figure 3.17: Selected simulated PT system trajectories for hierarchical and baseline controller comparison: (a) aircraft velocity, (b) pulse load current, (c) battery SOC, and (d) battery current.



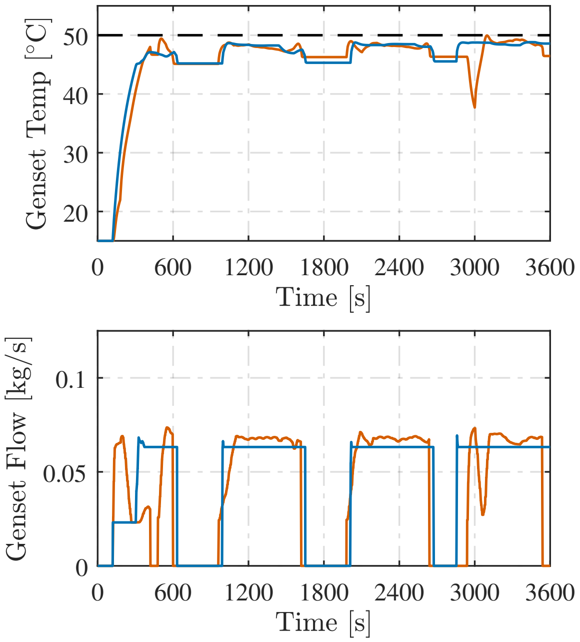
(a) Return temperature and flow rate

(b) Base temperature and flow rate

(c) Pulse temperature and flow rate

(d) Prop. temperature and flow rate

Figure 3.18: Selected simulated TMS system trajectories for hierarchical and baseline controller comparison: (a) return, (b) base, (c) pulse, (d) prop., and (e) genset temperatures and flow rates.



(e) Genset temperature and flow rate

Figure 3.18 (cont.): Selected simulated TMS system trajectories for hierarchical and baseline controller comparison: (a) return, (b) base, (c) pulse, (d) prop., and (e) genset temperatures and flow rates.

Also recall that the base and pulse loads have temperature dependent efficiencies; they are more efficient at lower temperatures. The hierarchical controller understands that coupling and reduces the avionic load temperatures to operate more efficiently.

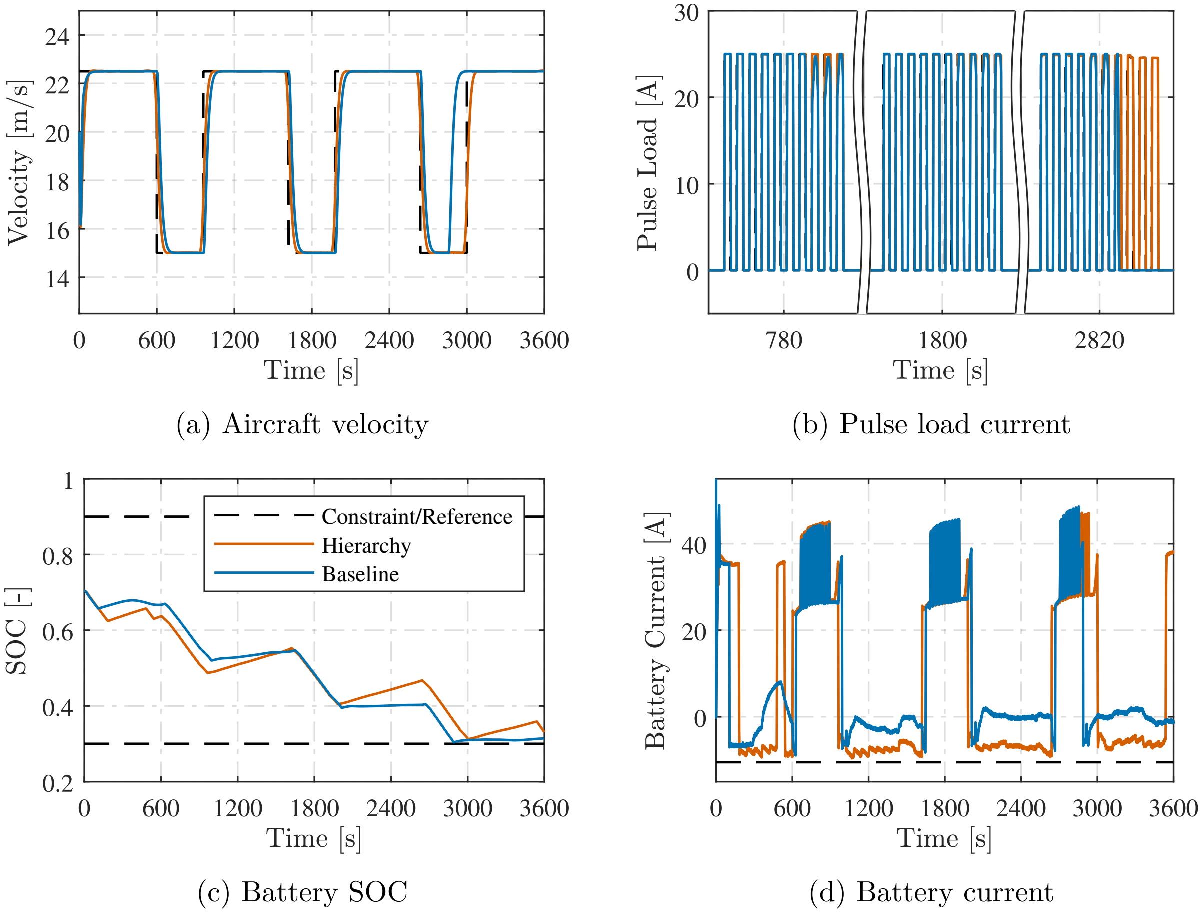
Intuitively, greater power levels result in more waste heat and consequently higher temperatures. However, that heuristic does not apply to these results. The hierarchical controller operated the genset at higher power levels and delivered more power to the pulse load than the baseline controller, yet it operated at similar, if not lower, operating temperatures. Comparing the thermal behavior of the two control approaches, the loop 1 return temperature exhibits significantly different operating behavior (Figure 3.18a). The baseline controller tracks the commanded 35°C setpoint well after initial transients, while the hierarchical controller, which is not incentivized to track a setpoint, operates closer to 31°C on average. Because the TMS architecture is serially cascaded (the chiller connects to loop 1, which connects to loop 2, which connects to loop 3), the thermal behavior of a loop is constrained by the thermal behavior its adjacent loops. For example, loop 2's lowest possible temperature is the same as loop 1's highest temperature. The cascaded TMS architecture indicates that the thermal performance between fluid loops in strongly coupled. The hierarchical controller, which explicitly accounts for this coupling through the system model, makes decisions to decrease the operating temperature in loop 1 to decrease the operating temperatures in loops 2 and 3, allowing the PT to successfully complete the mission objectives. The baseline controller, which is not designed to account for this coupling, operates and higher temperatures and is required to routinely shed loads resulting in a mission failure.

Decreasing the baseline controller return temperature setpoint would result in better baseline controller performance. However, tuning this and other control parameters would require extensive trial and error and system knowledge. For more complex systems, it could be challenging to identify which setpoints, parameters, and input-output pairings should be modified to improve operation. In contrast, the hierarchical MPC approach inherently accounts for the coupling and can be more easily tuned by modifying an intuitive objective function.

3.6.3 Experimental Results

The controllers are experimentally validated using the test platforms illustrated in Figure 3.1. The experimental results are illustrated in Figures 3.19 and 3.20. Similar to the simulation results, the baseline controller deviates from the desired trajectory (Figure 3.19a) towards the end of the mission because the battery's state of charge is depleted (Figure 3.19c). Although the genset attempts to recharge the battery during the cruise segments (Figure 3.19d), its electrical power output is thermally constrained (Figure 3.20e). Similar to the simulation results, the hierarchy accounts for the system coupling to sustain battery charge, respect thermal operating limits, and provide power to complete the mission objectives.

While the simulation and experimental control results are well validated, there are minor differences due to delays, model error or unmodeled behavior, estimation error, and sensor noise. For example, Figures 3.20c and 3.20e highlight small constraint violations in the pulse load and genset temperature. Additionally, communication and actuation delays between the testbed and control software result in more oscillatory mass flow rates in experimentation than in simulation. Despite these differences, the experimental results match the simulation results sufficiently close to consider the testbed validation a success.



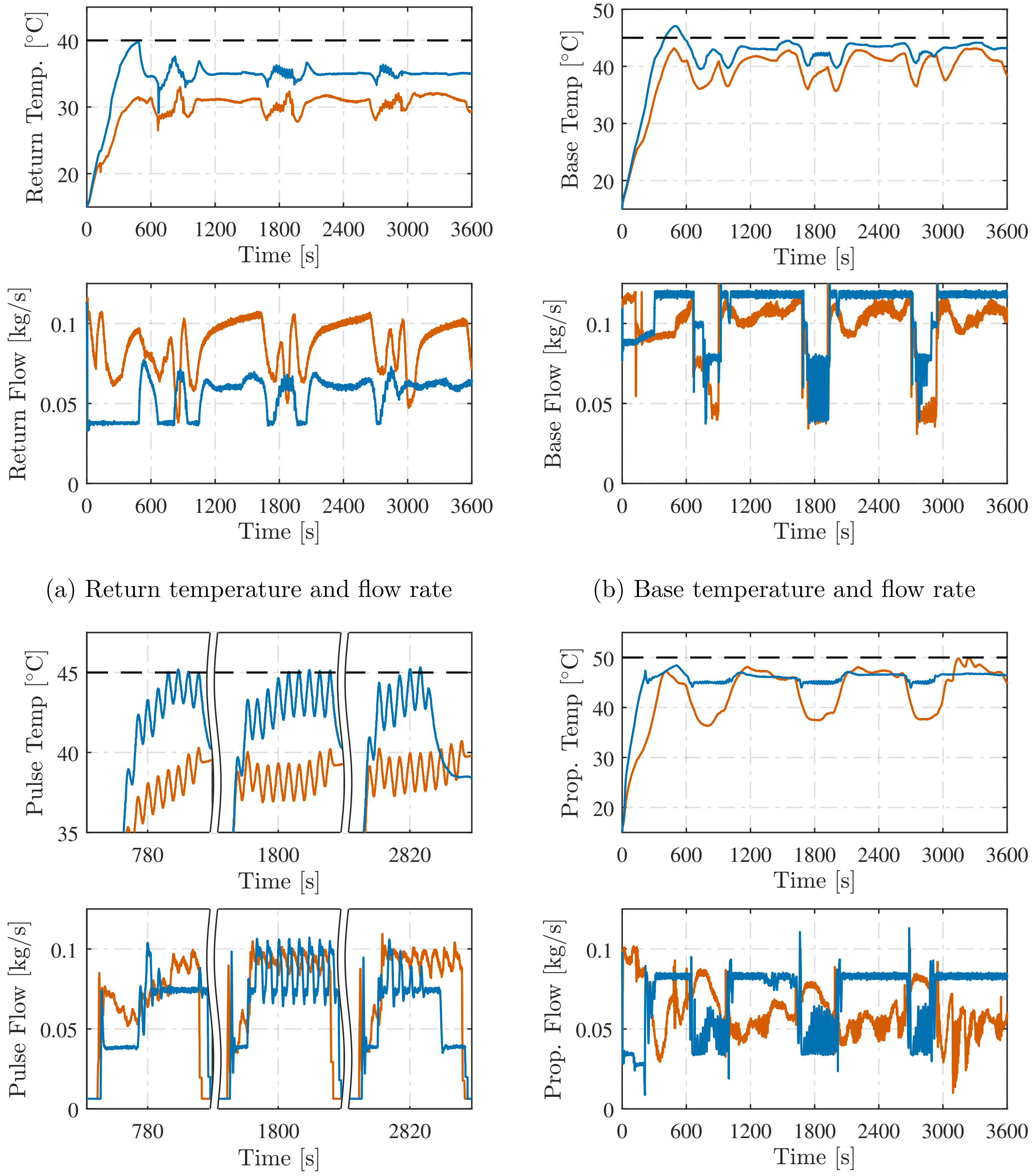
(a) Aircraft velocity

(b) Pulse load current

(c) Battery SOC

(d) Battery current

Figure 3.19: Selected experimental PT system trajectories for hierarchical and baseline controller comparison: (a) aircraft velocity, (b) pulse load current, (c) battery SOC, and (d) battery current.



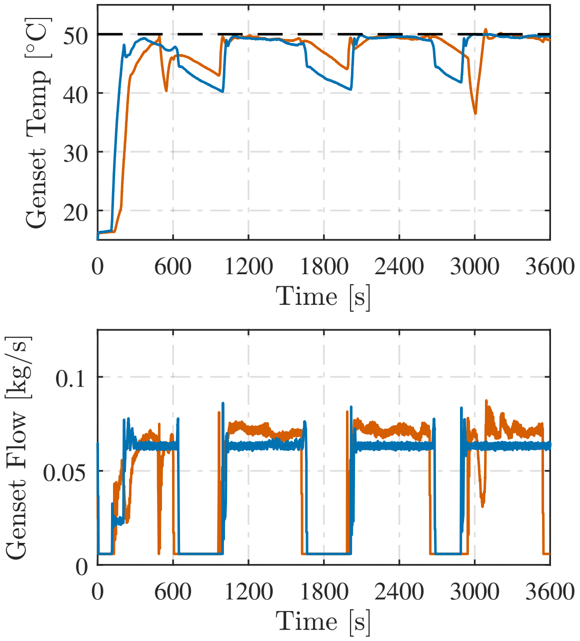
(a) Return temperature and flow rate

(b) Base temperature and flow rate

(c) Pulse temperature and flow rate

(d) Prop. temperature and flow rate

Figure 3.20: Selected experimental TMS system trajectories for hierarchical and baseline controller comparison: (a) return, (b) base, (c) pulse, (d) prop., and (e) genset temperatures and flow rates.



(e) Genset temperature and flow rate

Figure 3.20 (cont.): Selected experimental TMS system trajectories for hierarchical and baseline controller comparison: (a) return, (b) base, (c) pulse, (d) prop., and (e) genset temperatures and flow rates.

3.6.4 Controller Evaluation

Broadly speaking, aircraft energy systems can be evaluated by 1) performance, 2) efficiency, and 3) reliability criteria. Performance describes the system's ability to complete a desired class of mission or flight plans. Efficient operation is desired to minimize fuel consumption or other finite energy resources to increase range, reduce emissions, or extend capabilities. Reliability requires respecting constraints to avoid system failures or premature degradation, and is necessary for safe vehicle operation.

By failing to complete the third pulse train in both the simulation and experiment, the baseline controller failed to finish the mission. This behavior is the most apparent qualitative difference between the baseline and hierarchical control designs. To quantify the control performance, 5 figures of merit (FoM) are introduced that are associated with performance, efficiency, and reliability metrics.

The EPS performance is quantified by the pulse load tracking error

$$\mathcal{P}_{EPS} = \left\| x_{28}^{ref} - x_{28} \right\|_2^2 . \tag{3.16}$$

Tracking error generally occurs when the controller chooses to shed the pulse load to prioritize satisfying a flight critical loads or thermal constraints. The TMS performance is quantified by the average system temperature

$$\mathcal{P}_{TMS} = \text{mean}\left(\frac{1}{t_f} \int_0^{t_f} x_i dt \right) \forall i \in [29:56], \tag{3.17}$$

where t_f is the final mission time, and x_i $\forall i \in [29:56]$ is every temperature state (Table B.1).

The powertrain efficiency is quantified by the genset's fuel utilization over the entire mission.

$$\mathcal{E}_{PT} = \int_0^{t_f} (sfc)\, dt. \tag{3.18}$$

The TMS efficiency is quantified by the pump power consumption,

$$\mathcal{E}_{TMS} = \int_0^{t_f} P_{pump} dt. \tag{3.19}$$

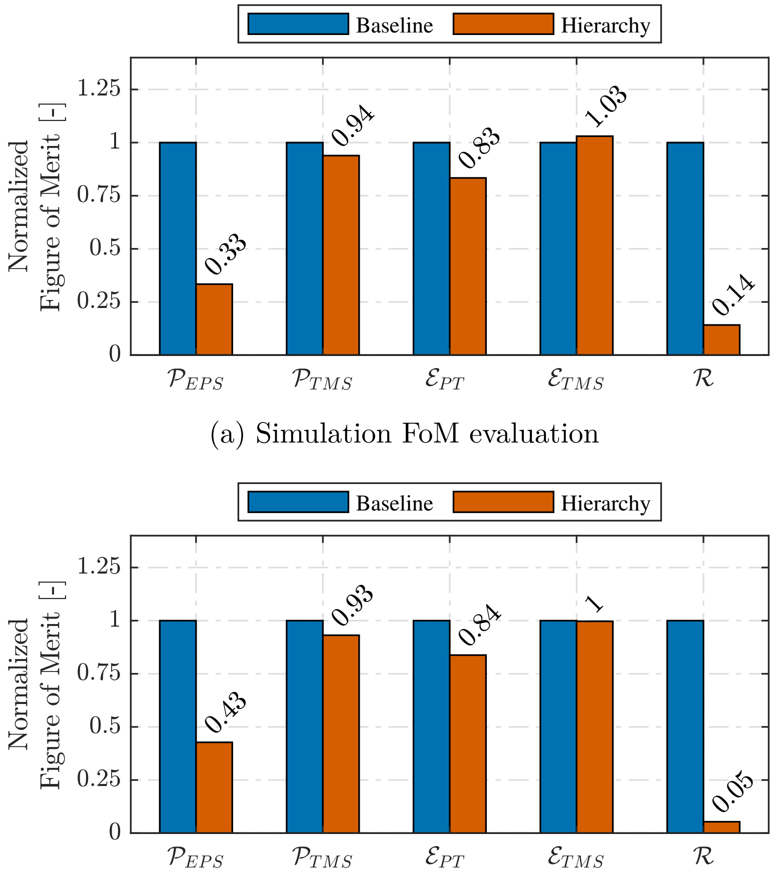
A more efficient system better utilizes fuel and requires less pumping power.

The system reliability is quantified by the amount and length of constraint violations,

$$\mathcal{R}_i = \frac{1}{\bar{x}_i} \int_0^{t_f} \max\left(0, x_i - \bar{x}_i\right) dt + \frac{1}{\underline{x}_i} \int_0^{t_f} \min\left(0, \underline{x}_i - x_i\right) dt \tag{3.20}$$

where $\mathcal{R}_i$ is a reliability metric for state x_i (state constraints are found in Table B.1). The system reliability is the sum of all individual reliability metrics $\mathcal{R} = \sum \mathcal{R}_i$.

The FoM are computed for both the simulation and experimental case studies and are presented in Figures 3.21. For illustrative purposes, the FoM are normalized such that the baseline value is one. Additionally, all FoM are designed such that a lower value is better. From Figure 3.21, the hierarchical controller outperforms the baseline controller on almost



(a) Simulation FoM evaluation

(b) Experimental FoM evaluation

Figure 3.21: Figure of merit comparison for the (a) simulated and (b) experimentally validated hierarchical and baseline controllers. The figures of merit are normalized against the baseline controller values. Smaller values indicate improvement in FoM.

all FoM. It is evident that the hierarchical controller delivered significantly more power to the pulse load because the baseline controller continually shed load to respect power and thermal constraints. The hierarchical controller, which knows the system coupling, was able to keep system temperatures 6-7% colder by increasing the flow rate in fluid loop 1. The increased flow rate increased heat transfer between fluid loops and heat rejection to the thermal sink. Although neither controller was explicitly designed to minimize system temperatures, the hierarchical controller recognized that decreasing system temperatures would improve performance. Interestingly, the hierarchical controller was able to decrease system temperatures despite using a similar amount of pumping energy as the baseline approach.

The hierarchical controller also demonstrated 16-17% improvement in fuel utilization. Although both controllers were designed to minimize fuel consumption, the baseline controller was forced to decrease the genset power output to a suboptimal operating condition to respect thermal limitations. The hierarchical system outperforms the baseline system in reliability, with only a fraction of the constraint violations. This is expected because the hierarchical controller explicitly accounts for operating constraints.

While these results illustrate qualitative advantages of a hierarchical model predictive control strategy in comparison to a conventional baseline approach, future work should consider systems with physical rather than virtual coupling between electrical, mechanical, and thermal systems to ensure that quantitative performance, efficiency, and reliability improvements translate to such conditions.

3.7 Conclusion

Increasing demand for environmentally sustainable aviation and more capable aircraft has initiated the trend for aircraft electrification. However, electrified aircraft are complex systems-of-systems with strongly coupled, constrained, multi-domain, and multi-timescale dynamics that are difficult to reliably operate using conventional control strategies without sacrificing performance or efficiency. Therefore, it is proposed to use a hierarchical MPC approach to address this challenge.

This chapter introduces an experimental hardware-in-the-loop testbed for an aircraft IPPTS. To facilitate model-based optimal control design, the testbed is modeled using the graph-based modeling framework, and the model is experimentally validated. A 3-layer hierarchical MPC for aircraft energy management is designed. The key advantage of hierarchical MPC over a centralized MPC is computationally efficient and effective coordination of multi-timescale systems. To support the experimental validation of both the hierarchical MPC and a baseline control approach, a decentralized observer using Kalman filter variants was introduced. The key contribution of this chapter was the experimental validation result that demonstrated the hierarchical controller's superior ability to account for system coupling to improve the performance, efficiency, and reliability of the closed-loop system.

In contrast, the baseline approach failed to complete all mission objectives. In particular, the hierarchical controller delivered more power to the pulse load while reducing the size of constraint violations and decreasing fuel utilization by 16-17%.

The demonstrated predictive control techniques are well suited for applications with well-known mission objectives such as commercial aircraft, which have pre-defined flight plans. However, significant uncertainty may be present in alternate applications where mission objectives and deliverables may change of the course of a mission. Therefore, it is important to investigate mission level control and rapid replanning in response to changing objectives. Additionally, electrified aircraft are designed well for autonomous and multi-agent applications where multiple aircraft work collaboratively to complete a common task. To meet this need, research should consider an additional level in the hierarchical control framework for simultaneous coordination of a multi-vehicle system.

The control results illustrated that the hierarchical controller performed better than the baseline controller because it could account from the coupling present amongst the three serially connected fluid loops. In the context of control design, it was natural to consider if the baseline controller could be redesigned to improve system performance. However, an alternative perspective is

Can the control behavior be improved by redesigning the plant?

This is the concept and objective of co-design. While this chapter focused on just control design, the following two chapters will investigate how to simultaneously design a plant and its feedback controllers.

Chapter 4

Plant Sizing and Closed-Loop Control Co-Design

4.1 Background

Electrified aircraft power systems can support a new variety of electronic capabilities for a range of applications. However, designing these systems is challenging because they are complex; comprised of various coupled energy system components, control systems, and sensor suites. The high degree of system complexity makes it difficult to evaluate how design decisions impact the performance of the end product. Therefore, this section will propose a co-design approach that systematically evaluates different design considerations and their influence on a design objective.

Control co-design is a design methodology, often posed as an optimization problem, where both plant and control design decisions are made within the same design framework. Co-design addresses the challenge of coupling between the design of a plant and the design of its controller [14]; the plant influences the control design process and the controller influences the plant design process. Traditional methods often discount this coupling by designing a plant first and the controller second, which result in suboptimal system performance that is inadequate for modern aircraft design. Much of the co-design literature is focused on open-loop co-design. In these problems, a plant and open-loop control design are identified through an optimization routine to identify a jointly optimal solution. While providing good design insights, an open-loop control action cannot be imposed on physical systems because open-loop control does not explicitly handle uncertainty or disturbances. Additionally, design studies usually consider pre-specified trajectories whereas the flight requirements are not precisely known apriori in practical applications. Some studies have explored robust OLCD methods, however, physical systems will not be operated in open-loop and require some closed-loop control.

Closed-loop co-design addresses the need to systematically identify a combined set of practical and optimal plant and control decisions. Instead of identifying open-loop control actions, CLCD methods design an optimal closed-loop control system. While the literature on CLCD is more brief than the OLCD counterpart, there have been various investigations leveraging different simultaneous [34–37], nested [38, 65], and iterative [110] optimization strategies. While these studies offer a range of contributions, they all focus on the application of CLCD to a specific subsystem with a single controller.

Therefore, the key contribution of this chapter is the formulation of a generalized and scalable CLCD optimization approach. The proposed methodology is demonstrated in a hybrid UAV design study where multiple subsystems and controllers are designed within the same optimization routine.

This chapter is outlined as follows. Section 4.2 will introduce the OLCD and CLCD formulations and a convergence analysis that can be used to systematically evaluate the tradeoff between optimization accuracy and computation time. Section 4.3 describes the hybrid electric UAV system and model that is leveraged within the design study introduced in Section 4.4. Section 4.5 summarizes the contributions and key design takeaways.

4.2 Design Methodology

Numerical optimization techniques are a useful tool to systematically evaluate and identify an optimal set of decisions that minimize an objective subject to design constraints. In the context of co-design, the optimization routine evaluates different plant and control design decisions. The following section will introduce both the open-loop and closed-loop control formulations. Because solving these optimizations can be computationally expensive, a convergence analysis technique is introduced as a guiding principle to evaluate the trade-off between optimization accuracy and computation time.

4.2.1 Open-Loop Control Co-Design

The OLCD process identifies an optimal set of plant parameters and open-loop control actions by solving the following dynamic optimization problem

$$
\begin{aligned}
&\text{minimize} && J && \text{(4.1a)}\\
&\text{with respect to} && t, \lambda_p, u &&\\
&\text{subject to} && &&\\
& && \begin{bmatrix} C_d \dot{x}(t) \\ 0 \end{bmatrix} = - \begin{bmatrix} \bar{M}_d \\ \bar{M}_a \end{bmatrix} P(t, x, a, x^e, u, \lambda_p) + \begin{bmatrix} D_d \\ D_a \end{bmatrix} P^e(t), && \text{(4.1b)}\\
& && g_j \left(t, x, a, u, \lambda_p\right) \leq 0, \forall j \in \left[1 : N_g\right], && \text{(4.1c)}\\
& && h_k \left(t, x, a, u, \lambda_p\right) = 0, \forall k \in \left[1 : N_h\right], && \text{(4.1d)}
\end{aligned}
$$

where J is the objective function to be minimized and can be a function of the optimization variables, t is time, λ_p is the plant parameters, u is the control trajectory, and g_j are h_k are inequality and equality constraints, respectively. In this problem, the decision variables are time, the plant parameters, and the open-loop control trajectory. The decisions variables are optimized with respect to the plant dynamics and other inequality and equality constraints. Notably, the plant dynamics (4.1b) are represented as a graph-based model because they are optimization-oriented; however, one could also use a general nonlinear DAE form model without loss of generality.

4.2.2 Closed-Loop Control Co-Design

Instead of identifying an open-loop control action, the CLCD process identifies an optimal plant parameterization and closed-loop control policy. The CLCD dynamic optimization problem is

$$
\begin{aligned}
&\text{minimize} && J && \text{(4.2a)}\\
&\text{with respect to} && t, \lambda_p, \lambda_c &&\\
&\text{subject to} && &&\\
& && \begin{bmatrix} C_d\dot{x}(t) \\ 0 \end{bmatrix} = -\begin{bmatrix} \bar{M}_d \\ \bar{M}_a \end{bmatrix} P(t, x, a, x^e, u, \lambda_p) + \begin{bmatrix} D_d \\ D_a \end{bmatrix} P^e(t), && \text{(4.2b)}\\
& && \dot{k} = f_k\left(t, x, a, k, \lambda_p, \lambda_c\right), && \text{(4.2c)}\\
& && u = f_u\left(t, x, a, k, \lambda_p, \lambda_c\right), && \text{(4.2d)}\\
& && g_j\left(t, x, a, u, k, \lambda_p, \lambda_c\right) \leq 0, \forall j \in \left[1 : N_g\right], && \text{(4.2e)}\\
& && h_k\left(t, x, a, u, k, \lambda_p, \lambda_c\right) = 0 \forall k \in \left[1 : N_h\right]. && \text{(4.2f)}
\end{aligned}
$$

In this problem, the control design decisions are represented by λ_c. In some cases, λ_c may represent the control gains in a feedback controller, while in other cases it may represent an open-loop control trajectory. Although this is a closed-loop control design problem, it may still be desirable to consider open-loop control behavior within the design process. For example, the reference trajectory passed to a closed-loop controller could be designed as an open-loop control action. Equation (4.2c) integrates the controller dynamics k, which are common in integral control or filter design, into the optimization problem. The feedback control law u is imposed through (4.2d). As introduced, this formulation can only be applied to structured control laws where the control policy is an explicit function of system parameters. It may be challenging to design non-structured or implicit control policies, like MPC, within the proposed framework.

The key difference between the OLCD (4.1) and CLCD (4.2) problems is in how the control behavior is optimized. The CLCD problem imposes structural constraints on the controller (i.e. the control law) that the optimization routine must consider, whereas the OLCD problem does not, thus addressing a relevant gap in the OLCD formulation. Note that generally the OLCD and CLCD problems will return different solutions. The CLCD solution will have less performance in comparison to the OLCD solution due to the additional control constraints. However, the advantage of identifying a physically implementable optimal feedback control strategy outweighs the performance degradation.

Piecewise and conditional functions are common in closed-loop control design. However, such functionality is often non-smooth and may negatively impact the convergence of a gradient-based optimization process. To address this concern, the follow section introduces smoothing functions that are useful in the CLCD process.

4.2.2.1 Smoothing Functions

The saturation function is a piecewise linear function that is often applied in closed-loop control design to limit the magnitude of the control action,

$$\mathrm{sat}\left(x\right)_L^U = \begin{cases} U & \text{if } x > U \\ x & \text{if } L \leq x \leq U \\ L & \text{if } L > x \end{cases}, \tag{4.3}$$

where U and L are the upper and lower saturation limits, respectively. Alternatively, $\mathrm{sat}(x)_L^U = \min\left(\max\left(x, L\right), U\right)$. The saturation function sensitivity $\frac{\partial\left(\mathrm{sat}(x)_L^U\right)}{\partial x}$ is discontinuous at the transition point between saturated and unsaturated outputs, which may negatively impact the convergence of a gradient-based optimization process. Instead, a smooth approximation of the saturation function is introduced,

$$\mathrm{sat}\left(x\right)_{-1}^{1} \approx \mathrm{asat}\left(x\right)_{-1}^{1} = \frac{x}{\sqrt[2p]{1+\left|x\right|^{2p}}}, \tag{4.4}$$

where $p \geq 1 \in \mathbb{Z}^+$ is a positive integer representing the degree of smoothing. While the approximated saturation function is provided for $U = 1$ and $L = -1$, it is preferred to saturate between arbitrary bounds. It is recommended to scale the saturated variable prior to saturation and then undo the scaling after applying (4.4)

$$\tilde{x} = \frac{2}{U-L}x - \frac{U+L}{U-L} \tag{4.5a}$$

$$y = \frac{U-L}{2}\tilde{y} + \frac{U+L}{2} \tag{4.5b}$$

where $\tilde{x}$ is the scaled version of x, $\tilde{y} = \text{asat}\left(\tilde{x}\right)_{-1}^{1}$, and $y = \text{asat}\left(x\right)_{L}^{U}$.

Similarly, encoding conditional logic is common in control design

$$\text{imp}\left(x, T\right)_{m_2}^{m_1} = \begin{cases} m_1 & \text{if } x \geq T \\ m_2 & \text{if } x < T \end{cases} \tag{4.6}$$

where m_i are operating modes and T is a switching threshold. The function sensitivity $\frac{\partial\left(\text{imp}(x,T)_{m_2}^{m_1}\right)}{\partial x}$ has infinite magnitude at the switching threshold and zero magnitude everywhere else, which may negatively impact the optimization process. This function can be approximated as a sigmoid function,

$$\text{imp}\left(x, 0\right)_{0}^{1} \approx \text{sig}\left(x, 0\right)_{0}^{1} = \frac{1}{1+e^{-px}}, \tag{4.7}$$

where $p \geq 1$ is a positive number representing the degree of smoothing. Similar to the saturation function, it is preferred to apply conditional logic with arbitrary operating modes and thresholds. It is recommended to scale the input variable before applying the sigmoid function, and then undo the scaling after applying (4.7)

$$\tilde{x} = x - T \tag{4.8a}$$

$$y = \left(m_1 - m_2\right)\tilde{y} + m_2 \tag{4.8b}$$

where $\tilde{x}$ is the shifted version of x, $\tilde{y} = \text{sig}\left(\tilde{x}, 0\right)_{0}^{1}$, and $y = \text{sig}\left(x, T\right)_{m_2}^{m_1}$.

The exact and approximated saturation and conditional logic functions are illustrated in Figure 4.1. The figure illustrates how the choice of smoothing factor p impacts the approximation. The smoothing factors should be held constant throughout the CLCD optimization.

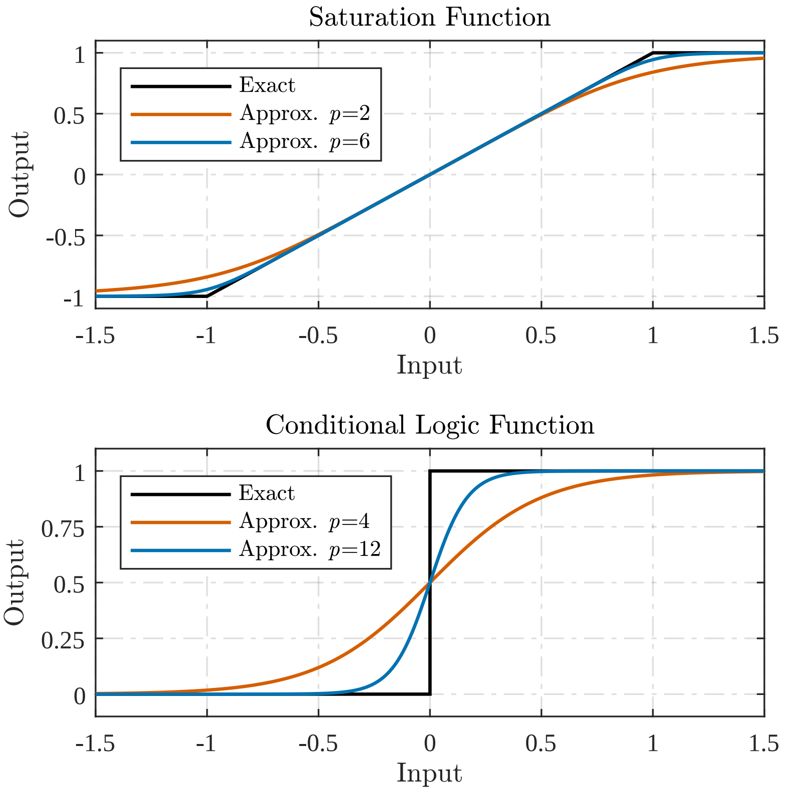


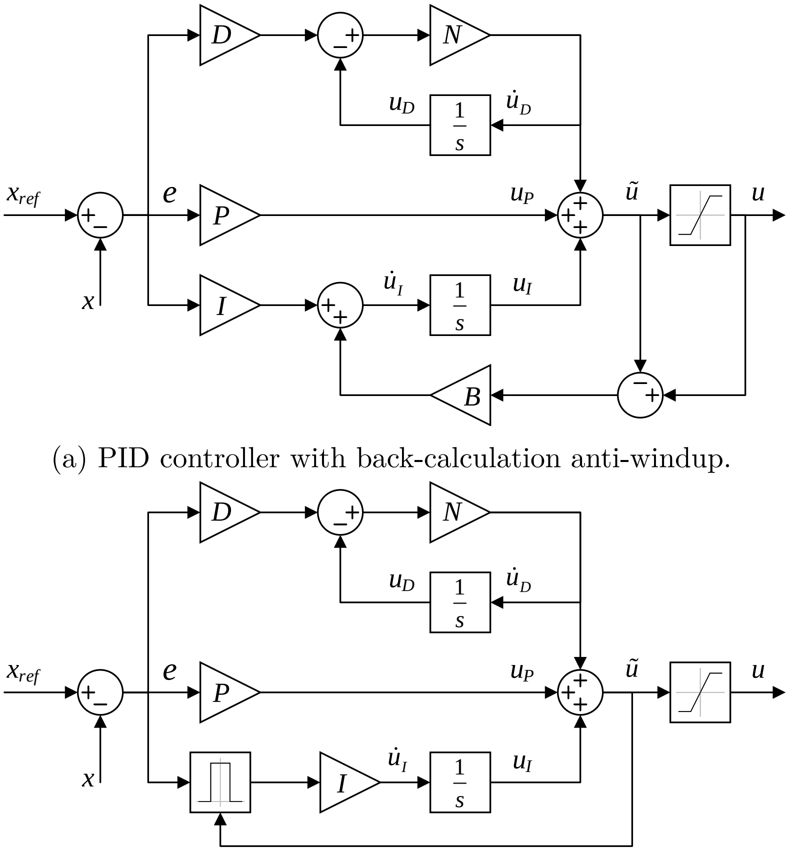
Figure 4.1: The smooth saturation and conditional logic functions for different smoothing factors.

4.2.2.2 Closed-Loop Control Examples

While any structured control law can be designed using the proposed CLCD approach, it is relevant to illustrate how control equations are integrated into the optimization problem. The process will be introduced using two common strategies relevant to energy system controls: proportional-integral-derivative control and equivalent consumption minimization strategy (ECMS) controllers. While the general form of these controllers is introduced in this section, variations of these controllers, tailored to a target application, are provided in Section 4.3.

Proportional-Integral-Derivative Control

PID control is one of the most widely used feedback control strategies in industry. A PID controller (Figure 4.2) is designed to reduce a system's regulation error to zero, which it may achieve through any combination of proportional, integral, and derivative control effort. Proportional control effort provides a control action proportional to tracking error; however,



(a) PID controller with back-calculation anti-windup.

(b) PID controller with clamping anti-windup.

Figure 4.2: PID controller block diagrams with (a) back-calculation and (b) clamping anti-windup mechanisms.

it has limited regulation applications because it often has non-zero steady state regulation error. Integral effort is often used to address this limitation by integrating the regulation error over time, resulting in zero steady-state regulation error. However, it is necessary to use anti-windup strategies when implementing integral control to avoid windup in the controller's integrator that may yield degraded or delayed responses. Derivative control slows the control response proportional to the rate of change of the regulation error and is often implemented with a low-pass filter because it is sensitive to noise and abrupt changes in references.

There are two common variations of PID controllers with different integral anti-windup methods: back-calculation and clamping. The governing equations for a PID controller with back-calculation anti-windup (Figure 4.2a) are

$$u(t) = \operatorname{asat}(\tilde{u}(t))_L^U, \tag{4.9a}$$

$$\tilde{u}(t) = u_P(t) + u_I(t) + \dot{u}_D(t), \tag{4.9b}$$

where,

$$e(t) = x_{ref}(t) - x(t), \tag{4.9c}$$

$$u_P(t) = Pe(t), \tag{4.9d}$$

$$\dot{u}_I(t) = Ie(t) + B\left(u(t) - \tilde{u}(t)\right), \tag{4.9e}$$

$$\dot{u}_D(t) = N\left(De(t) - u_D(t)\right), \tag{4.9f}$$

where u is the saturated control action, $\tilde{u}$ is the unsaturated control action, e is the tracking error, x_{ref} is the state reference, x is the feedback state, u_P, u_I, and u_D are the proportional, integral, and derivative control efforts, respectively, N is the derivative effort low pass filter, and P, I, D, and B are the proportional, integral, derivative, and back-calculation control gains, respectively.

The governing equations for a PID controller with clamping anti-windup (Figure 4.2b) are,

$$u(t) = \operatorname{asat}(\tilde{u}(t))_L^U, \tag{4.10a}$$

$$\tilde{u}(t) = u_P(t) + u_I(t) + \dot{u}_D(t), \tag{4.10b}$$

where,

$$e(t) = x_{ref}(t) - x(t), \tag{4.10c}$$

$$u_P(t) = Pe(t), \tag{4.10d}$$

$$\dot{u}_I(t) = Ie(t)\text{aclamp}\left(\tilde{u}, e\right)_L^U \tag{4.10e}$$

$$\dot{u}_D(t) = N\left(De(t) - u_D(t)\right). \tag{4.10f}$$

Note that the only difference between (4.9) and (4.10) is the computation of $\dot{u}_I$. In the clamping control formulation, the derivative of the integral effort is computed using an approximate clamping function because the clamping function is a conditional logic function

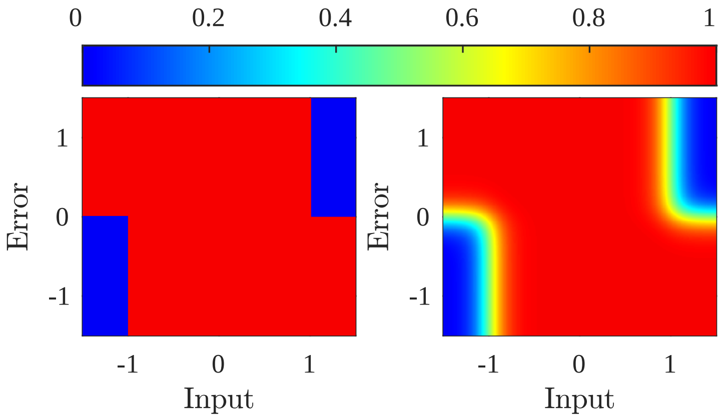


Figure 4.3: A comparison of the true and approximated smoothing clamping function for $p_x = p_y = 10$.

that prevents integration while in saturation. The clamping logic is formally written as

$$\text{clamp}\,(x, y)_L^U = \begin{cases} 1 & \text{if } (x \in [L, U]) \text{ or } (x > U \text{ and } y < 0) \text{ or } (x < L \text{ and } y > 0) \\ 0 & \text{else} \end{cases} . \tag{4.11}$$

The approximate clamping function encodes the logic statement (4.11) smoothly using the sigmoid function introduced in Section 4.2.2.1. However, the clamping function has two arguments, so a 2-dimensional sigmoid is used. The approximate clamping function is

$$\text{clamp}\,(x, y)_L^U \approx \text{aclamp}\,(x, y)_L^U = \frac{1}{1 + e^{-p_x(x-L)}} \frac{1}{1 + e^{p_y y}} + \frac{1}{1 + e^{p_x(x-U)}} \frac{1}{1 + e^{-p_y y}} \tag{4.12}$$

where p_x and p_y are positive smoothing factors along the x and y dimensions. As described in Section 4.2.2.1, it is recommended to scale the inputs before applying the approximation function. The smooth clamping function is illustrated in Figure 4.3, and the smoothing factors should be held constant throughout a CLCD optimization.

In the context of CLCD, both PID formulations have two control states u_I and u_D associated with the integral effort and derivative filter, respectively. Both control states are encoded through $k = f_k()$ in (4.2). The control laws (4.9a) and (4.10a) are encoded into (4.2) through $u = f_u()$. Note the optimization differences between the two PID formulations. The back-calculation approach implements an additional control parameter B, thus increasing the total number of design variables. In contrast, the clamping formulation has 1 fewer control parameters, but relies on a complex nonlinear function to apply anti-windup.

Equivalent Consumption Minimization Strategy

The equivalent consumption minimization strategy is popular in a hybrid energy system control approach that controls the total power contribution between a battery and genset because it provides a near global optimal control solution [111]. A notional hybrid energy system comprised of genset, battery, and electrical load is illustrated in Figure 4.4 The system stores energy in both the fuel and battery, and the ECMS strategy identifies an optimal control action that minimizes total energy consumption between the two. Because the fuel and battery energy storage mechanisms are dissimilar, an equivalence relationship is introduced. The equivalence relation converts the battery power consumption to an equivalent virtual fuel consumption metric. Then, the ECMS identifies a feasible genset command that minimizes the combined actual and virtual fuel consumption. The ECMS is posed as an optimization problem

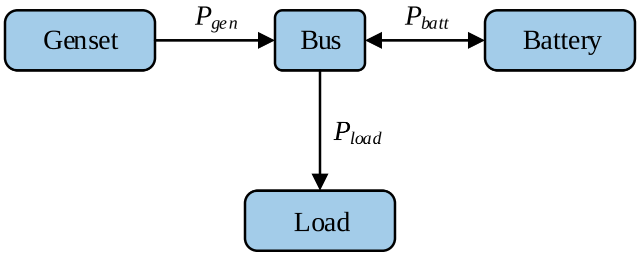


Figure 4.4: A notional hybrid energy system consisting of a genset, battery, and electrical load.

$$\begin{aligned} &\text{minimize} && \dot{m}_{fuel}\left(P_{gen}\right) + \varepsilon \frac{P_{batt}}{Q_{LHV}} && \text{(4.13a)} \\ &\text{with respect to} && P_{gen} \\ &\text{subject to} \\ & && P_{batt} + P_{gen} = P_{load}, && \text{(4.13b)} \\ & && I_{batt} \in \left[\underline{I}_{batt}, \bar{I}_{batt}\right], && \text{(4.13c)} \\ & && P_{gen} \in \left[\underline{P}_{gen}, \bar{P}_{gen}\right], && \text{(4.13d)} \end{aligned}$$

where $\dot{m}_{fuel}$ is the genset fuel consumption, $\varepsilon \frac{P_{batt}}{Q_{LHV}}$ is the battery virtual fuel consumption, ε is the equivalence factor, Q_{LHV} is the fuel's lower heating value, I is current, P is power, and the subscripts *batt*, *gen*, and *load* refer to battery, genset, and load variables, respectively. In this controller, ε is the tuning parameter that balances the relative cost of using battery versus engine power. Large ε results in less battery usage.

Assuming that $\dot{m}_{fuel}$ is convex or can be approximated as a convex function, the ECMS controller can be solved as a convex optimization problem, for which there are many efficient solution methods. By solving the ECMS as a convex optimization problem, the CLCD problem can be solved as a nested optimization. Alternatively, the problem can be reposed as an unconstrained optimization problem by substituting the power conservation constraint (4.13b) into the objective function and removing the battery current and genset power limits (4.13c) and (4.13d).

$$\begin{aligned} &\text{minimize} && \dot{m}_{fuel}\left(P_{gen}\right) + \varepsilon \frac{P_{load} - P_{gen}}{Q_{LHV}} && \text{(4.14)} \\ &\text{with respect to} && P_{gen} \end{aligned}$$

Because this is a one-dimensional optimization problem with convex cost, there exists an analytic solution to (4.14). However, the optimal genset power may not respect the battery current and genset power constraints. To respect those constraints, the optimal genset power can be saturated using (4.4). The result of this process is a static feedback control law that can be encoded into (4.2) through $u = f_u()$. This general process is provided on a more detailed system in Section 4.3.2.2,

4.2.3 Convergence Analysis

Because system-level co-design problems can be computationally expensive, a convergence analysis is proposed to evaluate the tradeoff between computation time and optimization accuracy. Solving the dynamic optimization problems (4.1)-(4.2) requires integrating the continuous-time plant dynamics to compute the state trajectories. In the context of optimization, there are two approaches to time integration: implicit and explicit [112]. Implicit methods approximate the system's continuous state trajectories $x(t)$ as a series of time-dependent polynomials $P(t)$. To solve the trajectories within the context of optimization, the states' time derivatives must equal the polynomials' time derivatives $\dot{x}(t) = \dot{P}(t)$ at discrete points in time. This relationship is treated as an equality constraint that the optimizer converges to zero. In contrast, explicit methods directly discretize the state dynamics and sequentially evaluate the discrete system model in a time-marching manner to approximate the continuous time trajectory. Common explicit integration methods include Forward-Euler or Runge-Kutta 4.

In general, both implicit and explicit methods discretize the system dynamics. The key difference is that the implicit approach integrates continuous-time dynamics into the optimization, whereas the explicit approach uses discrete time dynamics. Implicit integration problems generally have more decision variables than their explicit counterparts because the system states are treated as optimization variables. However, implicit methods can often capture the same dynamic behavior with fewer discretization steps, thus reducing the total number of decision variables and computation time for the implicit problem. Although both approaches are feasible, this work will use an implicit integration approach to optimize over the continuous time system dynamics with fewer discretization steps.

When solving a dynamic optimization problem, a key design decision is determining the number of required discretization steps to generate an accurate representation of the continuous state dynamics. While more discretization steps in a finite time interval yields a more accurate result, an excess of steps would lead to excessive computation times. In general, there is a trade-off between computational efficiency and optimization accuracy. Because co-design problems can be computationally expensive, repeatedly solving an optimization with a different number of discretization steps until the desired accuracy is reached will result in excessive design times. Therefore, the following section will introduce a convergence analysis that can be used to estimate the number of discretization steps required to meet a desired accuracy criteria before solving a full optimization problem.

The approach is as follows.

1. Formulate a more simple, but still representative, variation of the full optimization.
2. Solve the simplified optimization and record the state trajectory information.
3. Evaluate the optimization accuracy at the error nodes.
4. Repeat steps 2 and 3 until the solution accuracy meets a prespecified criterion.

There are a few approaches to posing a simple but still representative optimization problem. One approach could be to remove all decision variables from the optimization, thus reducing (4.1) and (4.2) to constraint satisfaction problems that would be easier to solve. If appropriate, a multi-phase optimization problem could be decomposed into multiple single-phase optimizations, where each of the single-phase problems represents one part of the multi-phase problem.

After posing the simplified optimization problem, it can be solved. When using implicit integration methods, the solution will return the polynomial approximation $P_i(t)$ for each state $x_i(t)$ (Figure 4.5). The accuracy of each polynomial approximation is quantified by the normalized approximation error e_i

$$e_i\left(t\right) = \frac{|\dot{P}_i(t) - \dot{x}_i(t)|}{\max\left(\dot{x}_i\right)}. \tag{4.15}$$

The optimization process constrains $e_i = 0$ at the discretization nodes (Figure 4.5). However, as illustrated by the different derivatives in Figure 4.5, the approximation error may be nonzero elsewhere along the trajectory. To evaluate the approximation error, error nodes

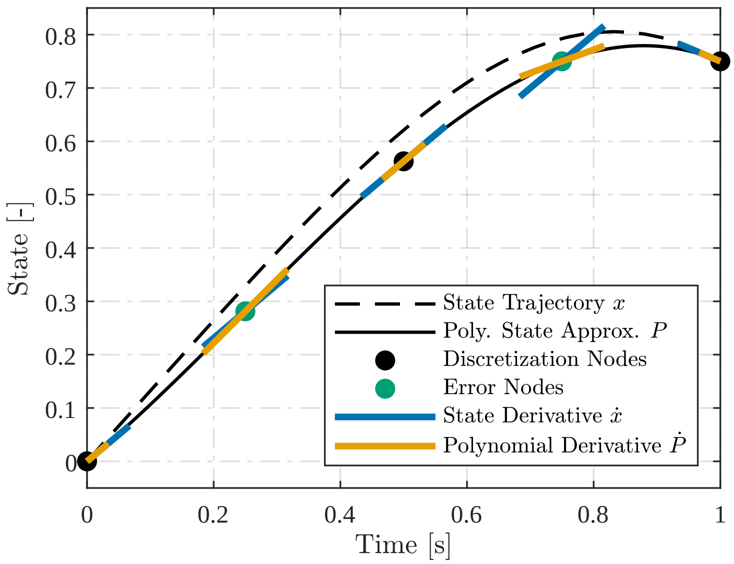


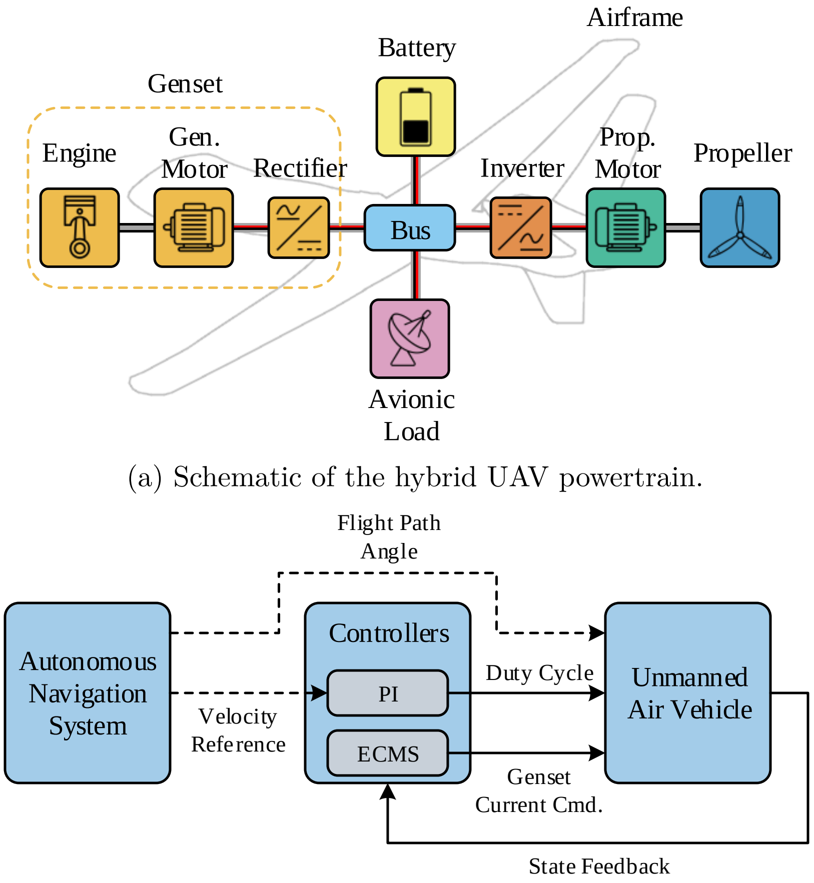
Figure 4.5: A comparison between a state trajectory and its polynomial approximation. While the state and polynomial derivatives match at the discretization nodes, they differ at the error nodes.

are placed at the midpoint between the discretization nodes, and the error is computed using (4.15). Because the simplified problems can be solved quickly, this process can be evaluated until the desired accuracy criteria is met.

4.3 Case Study System Description

To demonstrate the utility and performance of the proposed CLCD techniques, they are used to design a hybrid UAV powertrain and its respective feedback controllers. The following sections describe the system and its models.

The hybrid UAV powertrain is similar to the system introduced in Chapter 3 and is illustrated in Figure 4.6a. This hybrid UAV powertrain is similarly comprised of a battery for energy storage, a genset (engine, generator, and rectifier) for electrical power generation, a propulsion system (inverter, propulsion motor, propeller) to generate thrust, and an avionic load that emulates a high power electronic payload [96–98]. Unlike Chapter 3, this design problem focuses on powertrain design and control and assumes that all components have adequate thermal management.



(a) Schematic of the hybrid UAV powertrain.

(b) The closed-loop hybrid UAV system communication structure.

Figure 4.6: The (a) hybrid UAV powertrain schematic and (b) the communication structure of the closed-loop UAV system.

The aircraft's onboard control system comprises a PI controller that controls velocity reference tracking, and an ECMS controller that coordinates the load sharing between the battery and genset. Both controllers generate a duty cycle input that is used to control the aircraft's inverter and rectifier. Both control approaches are well utilized in the literature and are readily implemented in a closed-loop control design problem as described in Section 4.2.2.2.

While the two closed-loop controllers are sufficient to manage the aircraft's powertrain, they cannot sufficiently coordinate the vehicle performance over the course of a mission that is defined by velocity and altitude waypoints. The complete closed-loop control system is illustrated in Figure 4.6b. The remainder of this section introduces the plant and control models for this system. All model variables, descriptions, units, etc. are provided in Tables B.5-B.8 in the Appendix.

4.3.1 Plant Model

The plant model comprises three separate models: graph-based powertrain, genset, and vehicle dynamics models.

4.3.1.1 Graph-Based Powertrain Model

The UAV's powertrain is mathematically modeled in the graph-based modeling framework introduced in Chapter 2. The model was derived by connecting the component graph models from [57] according to the system schematic (Figure 4.6a). The graph model is illustrated in Figure 4.7 and its accompanying vertex, edge, input, and parameter tables are included in the Appendix (Tables B.5-B.8). Note that each cell in the battery has an open-circuit voltage V_{ocv} that is modeled as a 5^{th}-order polynomial,

$$V_{ocv}\left(x_1\right) = v_5 x_1^5 + v_4 x_1^4 + v_3 x_1^3 + v_2 x_1^2 + v_1 x_1 + v_0, \tag{4.16}$$

where v_i are coefficients reported in Table B.8.

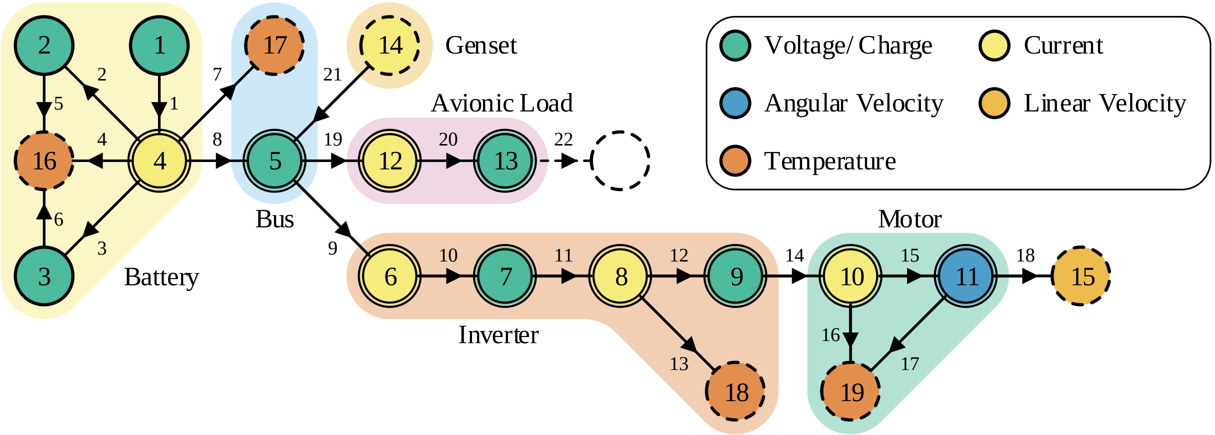


Figure 4.7: The graph model for the hybrid UAV powertrain. All graph model information is provided in Tables B.5-B.8.

4.3.1.2 Genset Model

The genset model is similar to the model introduced in Section 3.3.2, with the inclusion of fuel mass dynamics

$$\tau \dot{x}^e_{14} = \left(-x^e_{14} + K u_2\right), \tag{4.17a}$$

$$\dot{m}_{fuel} = c_1 a_5 x^e_{14} + c_0, \tag{4.17b}$$

where x^e_{14} is the genset current, a_5 is the genset operating voltage, m_{fuel} is the remaining fuel mass, τ is the genset current time constant, K is the input gain, u_2 is the genset input, and c_i are constant coefficients that describe the engine fuel consumption as a function of the genset power $a_5 x^e_{14}$. The genset parameters are values reported in Table B.8 and are derived from [57].

4.3.1.3 Vehicle Dynamics Model

The vehicle dynamics are modeled using a quasi-static longitudinal aircraft model similar to [113,114]. The quasi-static model captures the aircraft's dynamic velocity x^e_{15} and altitude h behavior and treats the vehicle's angle of attack α as a residual function of body forces and the flight path angle γ. The aircraft's free body diagram is illustrated in Figure 4.8, and the equations of motion are

$$m\dot{x}^e_{15} = T\cos\alpha - D - G\sin\gamma, \tag{4.18a}$$

$$\dot{h} = x^e_{15}\sin\gamma, \tag{4.18b}$$

$$0 = T\sin\alpha + L - G\cos\gamma, \tag{4.18c}$$

where,

$$D = \frac{1}{2}\rho A C_D {x^e_{15}}^2, \qquad L = \frac{1}{2}\rho A C_L {x^e_{15}}^2,$$

$$T = \rho d^4 C_T a_{11}^2, \qquad G = mg,$$

$$C_D = C_{D0} + K_L C_L^2, \quad C_L = C_{L0} + \frac{\alpha}{\alpha_{max}}\left(C_{L,max} - C_{L0}\right),$$

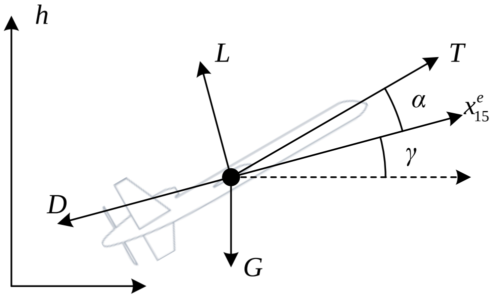


Figure 4.8: The aircraft free-body diagram used to derive the equations of motion for the vehicle dynamics model.

where T, D, L, and G are the thrust, drag, lift, and gravitational forces, respectively, ρ is the air density as a function of altitude, C_T, C_D, and C_L are the thrust, drag, and lift coefficients, A is the wing area, d is the propeller diameter, g is gravitational acceleration, C_{D0} and C_{L0} are the zero-drag and zero-lift coefficients, respectively, α_{max} is the maximum angle of attack, $C_{L,max}$ is the maximum lift coefficient, and K_L is the drag-due-to-lift factor.

The total vehicle mass is given by

$$m = m_{prop.} + m_{batt} + m_{fuel} + m_{genset} + m_{load} + m_{inv.} + m_{airframe}, \tag{4.19}$$

where the subscripts $prop.$, $batt$, $fuel$, $genset$, $load$, $inv.$, and $airframe$ refer to the mass of the aircraft's prop motor, battery, fuel, genset subsystem, avionic load, inverter, and airframe, respectively.

4.3.2 Feedback Control Model

The following sections introduce PI and ECMS control laws adapted for the hybrid UAV control system based on the examples introduced in Section 4.2.2.2.

4.3.2.1 PI Controller

A PI controller is an appropriate feedback controller to track the reference velocity profile commanded by the aircraft's autonomous navigation system. Figure 4.9 illustrates the PI controller block diagram with input saturation and clamping anti-windup, and the governing

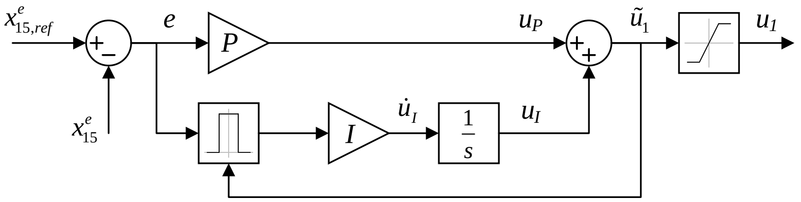


Figure 4.9: The hybrid UAV powertrain velocity tracking PI controller with clamping anti-windup.

equations are

$$u_1(t) = \text{asat}(\tilde{u}_1(t))_L^U, \tag{4.20a}$$

$$\tilde{u}_1(t) = u_P(t) + u_I(t), \tag{4.20b}$$

where,

$$e(t) = x^e_{15,ref}(t) - x^e_{15}(t), \tag{4.20c}$$

$$u_P(t) = Pe(t), \tag{4.20d}$$

$$\dot{u}_I(t) = Ie(t)\text{aclamp}\left(\tilde{u}_1(t), e(t)\right)_L^U, \tag{4.20e}$$

The PI controller seeks to reduce the tracking error e between the velocity reference $x^e_{15,ref}$ and the feedback velocity state x^e_{15} through proportional and integral control effort. The magnitude of the proportional and integral control effort is characterized by the proportional and integral gains P and I, respectively. The PI controller commands the propulsion system inverter's duty cycle u_1.

4.3.2.2 ECMS Controller

The ECMS controller and its simplification for a generic hybrid power system is introduced in Section 4.2.2.2. However, the hybrid UAV model described in Section 4.3.1 can be used to further simplify the result. As presented in Section 4.2.2.2, the process to simplify the ECMS optimization is to first remove equality constraints, then reduce the optimization to a 1 dimensional convex problem that can be solved analytically, and lastly apply the saturation functions to simplify the inequality current and power constraints.

Using the powertrain graph-based model, a power balance can be applied to vertices 4 and 5

$$N_s V_{ocv}\left(x_1\right) a_4 = R_s \frac{N_s}{N_p} a_4^2 + a_5 a_4, \tag{4.21a}$$

$$a_5 a_4 + a_5 x_{14}^e = a_5 a_6 + a_5 a_{12}, \tag{4.21b}$$

where x_1 is the battery SOC, a_4 is the battery current, a_5 is the bus voltage, a_6 is the inverter DC current, a_{12} is the avionic load current, x_{15}^e is the genset current, N_s is the number of battery series cells, N_p is the number of battery parallel cells, and R_s is that battery internal resistance (Tables B.5-B.8). The power balance (4.21a), which assumes that bulk battery inefficiency is governed by its internal resistance (Figure 4.7 edge 4), is used to compute the bus voltage a_5 as a function of the battery current a_4 and instantaneous battery SOC x_1 (4.22a). Additionally, the power balance (4.21b) is simplified to express the battery current a_4 as a function of the genset power x_{14}^e and net load current $a_6 + a_{12}$ (4.22b).

$$a_5 = N_s V_{ocv}\left(x_1\right) - R_s \frac{N_s}{N_p} a_4, \tag{4.22a}$$

$$a_4 = a_{12} + a_6 - x_{14}^e. \tag{4.22b}$$

Note that the instantaneous battery SOC and load currents are input into the controller.

Using the ECMS simplifications (4.22) and genset fuel consumption model (4.17b), the original ECMS objective function (4.13a) can be rewritten as

$$
\begin{aligned}
J &= \dot{m}_{fuel}\left(P_{gen}\right) + \varepsilon \frac{P_{batt}}{Q_{LHV}} \\
&= c_1 x_{14}^e a_5 + c_0 + \varepsilon \frac{N_s V_{ocv}\left(x_1\right) a_4}{Q_{LHV}} \\
&= c_1 x_{14}^e \left(N_s V_{ocv}\left(x_1\right) - R_s \frac{N_s}{N_p}\left(a_{12} + a_6 - x_{14}^e\right)\right) + c_0 + \varepsilon \frac{N_s V_{ocv}\left(x_1\right)\left(a_{12} + a_6 - x_{14}^e\right)}{Q_{LHV}} \\
&= \left(c_1 R_s \frac{N_s}{N_p}\right) {x_{14}^e}^2 + \left(c_1 N_s V_{ocv}\left(x_1\right) - c_1 R_s \frac{N_s}{N_p}\left(a_{12} + a_6\right) - \varepsilon \frac{N_s V_{ocv}\left(x_1\right)}{Q_{LHV}}\right) x_{14}^e + \\
&+ \left(c_0 + \varepsilon \frac{N_s V_{ocv}\left(x_1\right)\left(a_{12} + a_6\right)}{Q_{LHV}}\right)
\end{aligned}
\tag{4.23}
$$

While more complex, this equivalent form for the objective function is a one-dimensional quadratic function of the genset current x_{15}^e. The remaining variables in (4.23) are either model parameters or inputs to the optimization. Recall that the optimal genset current is the objective function minimizer ${x_{15}^e}^* = \arg\min(J)$, which can be computed analytically because (4.23) is quadratic.

While computing the minimizing genset current is straightforward, the simplification process has ignored the battery current and genset power constraints (4.13c) and (4.13b) and that the genset requires a duty cycle command and not a current command. The prior is addressed through saturation functions that guarantee a feasible genset current (4.24b)-(4.24c), while the latter current command is converted to a duty cycle using the genset's input gain K (4.24d).

$$
{x_{15}^e}^* = -\frac{N_p}{N_s} \frac{c_1 N_s V_{ocv}\left(x_1\right) - c_1 R_s \frac{N_s}{N_p}\left(a_{12} + a_6\right) - \varepsilon N_s V_{ocv}\left(x_1\right) Q_{LHV}^{-1}}{2 R_s c_1}, \tag{4.24a}
$$

$$
{x_{15,s1}^e}^* = \operatorname{asat}\left({x_{15}^e}^*\right)_{a_{12}+a_6-\underline{a}_4}^{a_{12}+a_6-\bar{a}_4}, \tag{4.24b}
$$

$$
{x_{15,s2}^e}^* = \frac{1}{a_5} \operatorname{asat}\left(a_5 {x_{15}^e}^*\right)_{\underline{P}_{gen}}^{\bar{P}_{gen}}, \tag{4.24c}
$$

$$
u_2 = \frac{{x_{15,s2}^e}^*}{K}. \tag{4.24d}
$$

The ECMS control solution (4.24) can be directly encoded into the CLCD problem (4.2).

The controller is tuned by varying the equivalence factor ε.

4.3.3 System Summary

The closed-loop hybrid UAV system comprises 5 separate models: powertrain, genset, airframe, PI controller, and ECMS. Figure 4.10 is included to illustrate the interactions and information passing between the 5 models and the autonomous navigation system. To summarize, the powertrain exchanges motor speed a_{11} and vehicle velocity x_{15}^e states with the airframe model and bus voltage a_5 and genset current states x_{14}^e with the genset model. Additionally, the altitude state h is passed to the powertrain model to determine the air density value required to compute the total propulsion power. The aircraft model communicates the velocity state x_{15}^e to the PI controller to compute the inverter duty cycle input u_1 that is applied to the powertrain model. The ECMS controller requires the instantaneous battery state of charge x_1, inverter DC current a_6, avionic load current a_{12}, and bus voltage a_5 to compute the genset command u_2. The autonomous navigation system commands the aircraft's flight path angle γ and velocity trajectory $x_{15,ref}^e$. Recall that all model variable and parameter names are in Tables B.5-B.8.

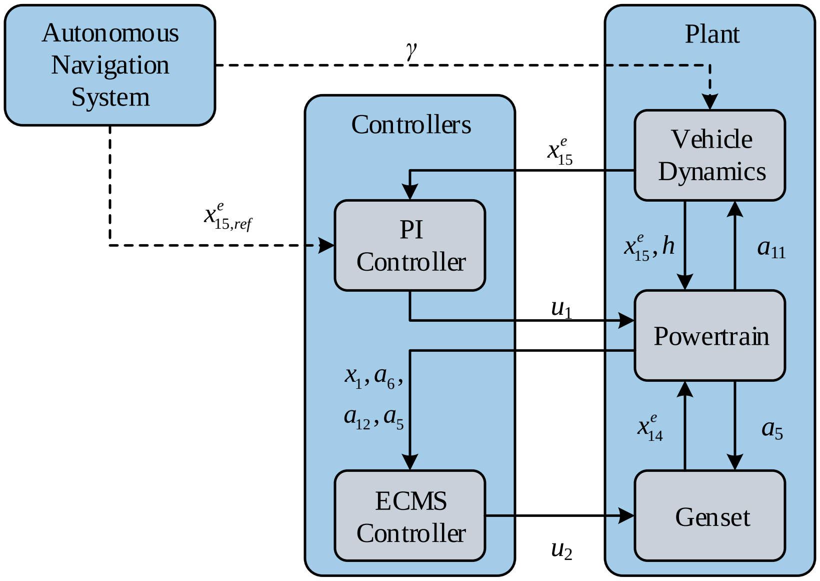


Figure 4.10: Variable information passing between the 5 models that comprise the closed-loop hybrid UAV system.

4.4 Design Study

Electrified aircraft offer opportunities for more efficient and capable vehicles. In particular, there is interest in faster aircraft across different industries because, depending on the application, a faster aircraft may deliver more packages, transport more people, or survey more land/crops in a fixed amount of time. When designing aircraft, the battery and propulsion motor selections are two key design considerations that dictate the overall vehicle performance. While larger batteries and motors may deliver more energy and power, the increased component weight may result in suboptimal system performance. Additionally, the control design strongly governs the dynamic system performance, influencing how power is both generated, stored, and delivered to the subsystems. Therefore, this study will investigate the design of a hybrid UAV's battery pack, propulsion motor, and their respective controllers. To demonstrate the effectiveness of the introduced CLCD approach, the hybrid UAV is designed using 5 different co-design optimization strategies:

1. **Case 1: Simultaneous open-loop co-design (SOL)**

   Identify an optimal plant and open-loop control decisions as described in (4.1). While the open-loop control is not physically applicable, this case will illustrate the *best case design*.

2. **Case 2: Sequential co-design (SQ)**

   The sequential design study represents a conventional design approach wherein the plant is designed first, and the controller designed second. The plant design is identified using the optimal plant parameters from Case 1 (SOL). Then, the plant parameters are fixed and a second CLCD optimization (4.2) identifies the optimal closed-loop control parameters.

3. **Case 3: Simultaneous closed-loop co-design (SCL)**

   The simultaneous closed-loop problem identifies a set of optimal plant and closed-loop control parameters at the same time by solving (4.2).

4. **Case 4: Siloed plant design (PCL)**

This study identifies an optimal plant design by solving the CLCD optimization problem (4.2) where the closed-loop control parameters are fixed.

5. **Case 5: Siloed control design (CCL)**

   This study identifies an optimal closed-loop control parameterization by solving the CLCD optimization problem (4.2) where the plant parameters are fixed.

The merit of each optimized design is evaluated against a baseline aircraft design (BASE). The baseline aircraft system was designed based on the experimentally validated powertrain parameters from Chapter 3. The PI controller was tuned via pole placement methods by linearizing the system around the cruise condition (Table 4.1), generating the root locus, and using MATLAB's control design toolbox to change the closed-loop pole locations until a desired fast, but damped, step response was produced. The ECMS controller was tuned to be charge sustaining by changing the equivalence factor until the battery depleted ∼10% SOC over the course of the mission (Table 4.1).

The plant design parameters are the number of series battery cells N_s, number of parallel battery cells N_p, prop. motor constant k_v, and the prop. motor coil resistance R_m. The control design parameters are the PI controller tuning gains P, and I, and the ECMS equivalence factor ε. The number of battery cells determine the battery capacity and operating voltage, and the motor parameters determine the electro-mechanical conversion and electrical efficiencies. The nominal battery cell and parameters are from a 3Ah Samsung INR18650-30Q cell. While a physical battery pack may only have an integer number of cells, this design study will permit a continuous number of battery cells. Discrete optimization methods could be used to convert the continuous design solution to an integer solution [33]. The motor parameter design space is described by the reported parameter values from the NeuMotors 8000 series brushless DC motors. These motors have a power level range from 4.5-21.4kW, which is well suited for the proposed application. To formulate the motor parameter design space, the NeuMotors 8000 series motor database is downloaded and each model's motor constant, coil resistance, mass, and current limit are recorded. The feasible motor parameter design space is defined by the convex hull H_m of all motor possible constant and coil resistance pairs. Figure 4.11 illustrates the feasible motor parameter design space and the

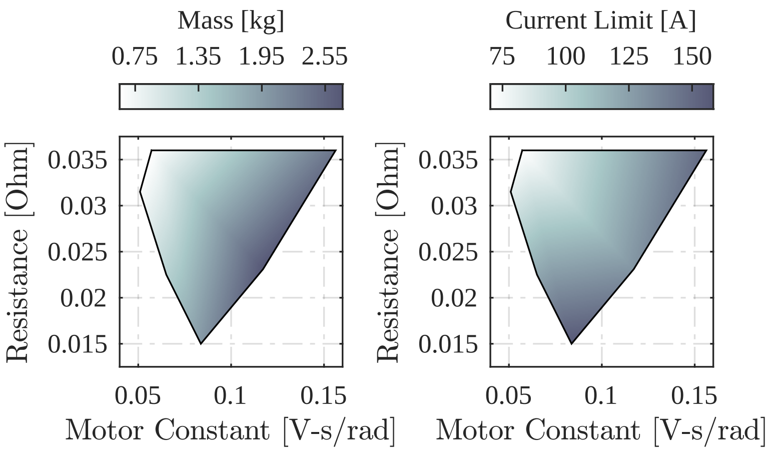


Figure 4.11: The feasible motor constant and resistance design space for the propulsion motor based on the NeuMotors 8000 series motors. The shading illustrates the changes in mass and maximum coil current.

relationship with motor mass and current limit.

By designing the battery and prop. motor, the system mass and operating limits change. The battery and prop. mass and current limits change as a function of their design parameters N_s, N_p, k_v, and R_m

$$m_{batt} = N_s N_p m_{cell}, \tag{4.25a}$$

$$\underline{a}_5 = N_p \underline{I}_{cell}, \tag{4.25b}$$

$$\bar{a}_5 = N_p \bar{I}_{cell}, \tag{4.25c}$$

$$m_{prop.} = m_0 + m_1 k_v + m_2 R_m, \tag{4.25d}$$

$$\underline{a}_{10} = 0, \tag{4.25e}$$

$$\bar{a}_{10} = i_0 + i_1 k_v + i_2 R_m, \tag{4.25f}$$

where a_5 is the battery current and a_{10} is the prop. current.

The aircraft is designed for a specified mission characterized by the velocity, altitude, and avionic load profiles illustrated in Figure 4.12. The mission begins in a cruise condition before a dive into a sensing segment where the aircraft activates its avionic load that could represent a high power sensing device [96–98]. The sensing segment is followed by a climb

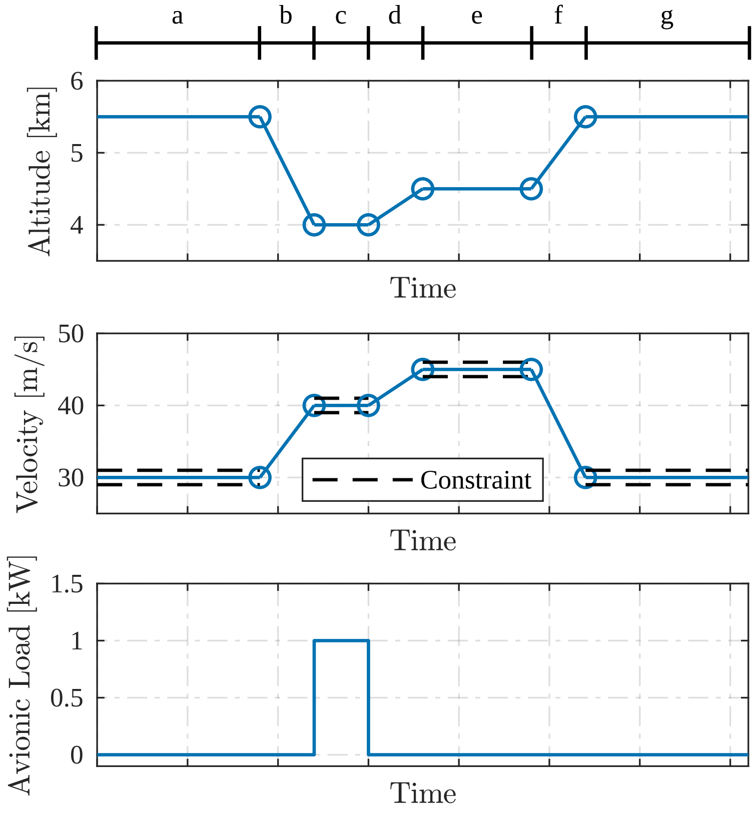


Figure 4.12: The desired aircraft altitude, velocity, and avionic load power for the hybrid UAV powertrain design study. Table 4.1 provides additional specifications for each mission segment.

to dash, and then another climb back to cruise. The mission segment classification, length, and boundary conditions are described in Table 4.1. Note that the climb and dive segments are variable length and do not end until the aircraft meets the prescribed flight conditions. For example, the sensing segment does not start until the vehicle reaches 4 km altitude at 40 m/s velocity.

Table 4.1: Mission profile conditions for the Figure 4.12.

Mission Segment	Description	Discretization Steps	Duration [min]	Velocity [m/s]	Altitude [km]	Load Power [kW]
a	Cruise	10	6	30	5.5	0
b	Dive	20	t^b	30-40	5.5-4.0	0
c	Sensing	10	2	40	4.0	1.0
d	Climb	22	t^d	40-45	4.0-4.5	0
e	Dash	10	4	45	4.5	0
f	Climb	22	t^f	45-30	4.5-5.5	0
g	Cruise	10	6	30	5.5	0

4.4.1 Co-Design Optimization Problems

The proposed design study requires solving both the OLCD and proposed CLCD problems (4.1) and (4.2). The objective is to design a vehicle that can complete a mission in minimum time by selecting the battery, motor, and control parameters. The OLCD problem is

$$\text{minimize} \quad t_f^g \tag{4.26a}$$

$$\text{with respect to} \quad t^b, t^d, t^f, N_p, N_s, R_m, k_v, u_1, u_2, \gamma$$

$$\text{subject to}$$

$$\left.\begin{bmatrix} C_d \dot{x} \\ 0 \end{bmatrix} = - \begin{bmatrix} \bar{M}_d \\ \bar{M}_a \end{bmatrix} P + \begin{bmatrix} D_d \\ D_a \end{bmatrix} P^e, \right\} \text{Powertrain Model} \tag{4.26b}$$

$$\left.\begin{aligned} &\tau \dot{x}_{14}^e = \left(-x_{14}^e + K u_2\right), \\ &\dot{m}_{fuel} = c_1 a_5 x_{14}^e + c_0, \end{aligned}\right\} \text{Genset Model} \tag{4.26c}$$

$$\left.\begin{aligned} &m \dot{x}_{15}^e = T \cos\alpha - D - G \sin\gamma, \\ &\dot{h} = x_{15}^e \sin\gamma, \\ &0 = T \sin\alpha + L - G \cos\gamma, \end{aligned}\right\} \text{Vehicle Dynamics Model} \tag{4.26d}$$

$$\left.\begin{array}{ll} x_{15}^{e}\left(t_{f}^{a,f,g}\right)=30\text{m/s}, & x_{15}^{e}\left(t^{a},t^{g}\right)=30\pm1\text{m/s} \\ x_{15}^{e}\left(t_{f}^{b,c}\right)=40\text{m/s}, & x_{15}^{e}\left(t^{c}\right)=40\pm1\text{m/s} \\ x_{15}^{e}\left(t_{f}^{d,e}\right)=45\text{m/s} & x_{15}^{e}\left(t^{e}\right)=45\pm1\text{m/s} \end{array}\right\}\text{Mission Velocity} \tag{4.26e}$$

$$\left.\begin{array}{l} h\left(t^{a,g}\right)=5.5\text{km}, \\ h\left(t^{c}\right)=4.0\text{km}, \\ h\left(t^{e}\right)=4.5\text{km} \end{array}\right\}\text{Mission Altitude} \tag{4.26f}$$

$$\left.\gamma\left(t^{a,c,e,g}\right)=0^{\circ}\right\}\text{Mission Flight Path Angle} \tag{4.26g}$$

$$\left.P_{22}^{e}\left(t^{c}\right)=1kW\right\}\text{Mission Avionic Power} \tag{4.26h}$$

$$\text{Tables B.5, B.7, and B.8 upper and lower bounds.} \tag{4.26i}$$

Note the following notation: t_f^i is the final time of mission segment i and t^i is the time duration of mission segment i. Here, the optimization routine is written for readability, where each set of constraints are labeled. Because the duration of the cruise and dive segments are variable in length, those time durations t^b, t^d, and t^f are treated as decision variables in addition to the plant and open-loop control design variables. The powertrain, genset, and vehicle dynamics models are included by (4.26b)-(4.26d). The mission structure defined by Figure 4.12 is encoded into the optimization using (4.26e)-(4.26h). Together, the velocity, altitude, and flight path angle constraints (4.26e)-(4.26g) require the aircraft to start and end each mission segment at a specified velocity and altitude while in steady level flight. Constraint (4.26e) requires the aircraft fly within a velocity threshold for the duration of the cruise, sensing, and dash segments. The avionic load power is enforced through (4.26h). For brevity, all other general lower and upper bound constraints are summarized in Tables B.5, B.7, and B.8.

Similarly, the CLCD problem is formulated as

$$
\begin{aligned}
&\text{minimize} && t_f^g && \text{(4.27a)}\\
&\text{with respect to} && t^b, t^d, t^f, N_p, N_s, R_m, k_v, P, I, \varepsilon, x_{15,ref}^e, \gamma && \\
&\text{subject to} && && \\
& && \text{Equation (4.26b)}\Big\}\ \text{Powertrain Model} && \text{(4.27b)}\\
& && \text{Equation (4.26c)}\Big\}\ \text{Genset Model} && \text{(4.27c)}\\
& && \text{Equation (4.26d)}\Big\}\ \text{Vehicle Dynamics Model} && \text{(4.27d)}\\
& && \left.\begin{aligned}
&u_1(t) = \text{asat}(\tilde{u}_1(t))_L^U,\\
&\tilde{u}_1(t) = u_P(t) + u_I(t),\\
&e(t) = x_{15,ref}^e(t) - x_{15}^e(t),\\
&u_P(t) = Pe(t),\\
&\dot{u}_I(t) = Ie(t)\text{aclamp}\left(\tilde{u}_1(t), e(t)\right)_L^U,
\end{aligned}\right\}\ \text{PI Controller Model} && \text{(4.27e)}\\
& && \left.\begin{aligned}
&\text{Equation 4.24a}\\
&{x_{15,s1}^e}^* = \text{asat}\left({x_{15}^e}^*\right)_{a_{12}+a_6-N_p\underline{a}_4}^{a_{12}+a_6-N_p\bar{a}_4},\\
&{x_{15,s2}^e}^* = \frac{1}{a_5}\text{asat}\left(a_5{x_{15}^e}^*\right)_{\underline{P}_{gen}}^{\bar{P}_{gen}},\\
&u_2 = \frac{{x_{15,s1}^e}^*}{K},
\end{aligned}\right\}\ \text{ECMS Controller Model} && \text{(4.27f)}\\
& && \text{Equation (4.26e)}\Big\}\ \text{Mission Velocity} && \text{(4.27g)}\\
& && \text{Equation (4.26f)}\Big\}\ \text{Mission Altitude} && \text{(4.27h)}\\
& && \text{Equation (4.26g)}\Big\}\ \text{Mission Flight Path Angle} && \text{(4.27i)}\\
& && \text{Equation (4.26h)}\Big\}\ \text{Mission Avionic Power} && \text{(4.27j)}\\
& && \text{Tables B.5, B.7, and B.8 upper and lower bounds.} && \text{(4.27k)}
\end{aligned}
$$

For brevity, the plant model and mission constraints are not restated in the CLCD formulation. The key differences between the OLCD and CLCD problems are the control decision variables and the inclusion of the controller dynamics. Variations of the OLCD and CLCD

Table 4.2: The free design variables for the hybrid UAV co-design studies from List 4.4. The SQ design problem is initialized with the optimal plant parameters from the SOL design problem.

Design Study	Optimization Problem	Plant Decisions	Control Decisions	Duration Decisions
BASE	(4.27)	-	-	t^b,t^d,t^f
SOL	(4.26)	N_p,N_s,R_m,k_v	u_1,u_2,γ	t^b,t^d,t^f
SQ	(4.27)	-	γ,P,I,ε,$x^e_{15,ref}$	t^b,t^d,t^f
SCL	(4.27)	N_p,N_s,R_m,k_v	γ,P,I,ε,$x^e_{15,ref}$	t^b,t^d,t^f
PCL	(4.27)	N_p,N_s,R_m,k_v	-	t^b,t^d,t^f
CCL	(4.27)	-	γ,P,I,ε,$x^e_{15,ref}$	t^b,t^d,t^f

optimizations are solved for the various design scenarios in List 4.4. Table 4.2 lists the design variables considered in each design scenario.

These optimization problems are built in the python-based optimization packages OpenMDAO and Dymos [115,116]. The tools facilitate a modular representation of the optimization problem through its efficient assembly of total derivatives of coupled subsystems. A key advantage of these tools is the option to define analytical partial derivatives, which may result is faster and more robust optimizations when using gradient-based optimizers in contrast to less efficient finite differencing methods. The optimizations are solved using the sequential least squares programming (SLSQP) gradient-based optimizer from SciPyOptimize [117]. They are solved on a workstation with an Intel i7 2.90 GHz CPU and 32 GB of RAM. Each design case is warm-started from the baseline optimization solution.

4.4.2 Convergence Study

The convergence analysis methods introduced in Section 4.2.3 are used to select the number of discretization steps necessary to solve (4.26) and (4.27). The mission comprises 3 types of mission phases: constant speed and altitude flight (Case 1), dive (Case 2), and climb (Case 3). To apply the proposed method, the full optimization is reduced to 3 separate optimizations that optimize the system over each of the mission phase types. Additionally, all design variables are held constant except for the velocity reference and flight path angle. For the dive and climb optimizations (Cases 2 and 3), the final phase time is also allowed to vary.

Each optimization case is solved for 10 to 50 discretization steps in 2 node increments. In Dymos, the number of discretization steps is defined through the *num_segments* parameter. The aircraft velocity and fuel mass error statistics are computed using (4.15). These two states' statistics are presented because they are critical vehicle states and demonstrated the largest discretization error amongst all other states. Each case is designed with an average 0.1% accuracy specification.

The aircraft velocity and fuel mass error statistics are presented in Figure 4.13. For all discretization steps from 10 to 50, the velocity and fuel mass error statistics for Case 1 are below the predesignated 0.1% accuracy specification. Case 1, which represents a

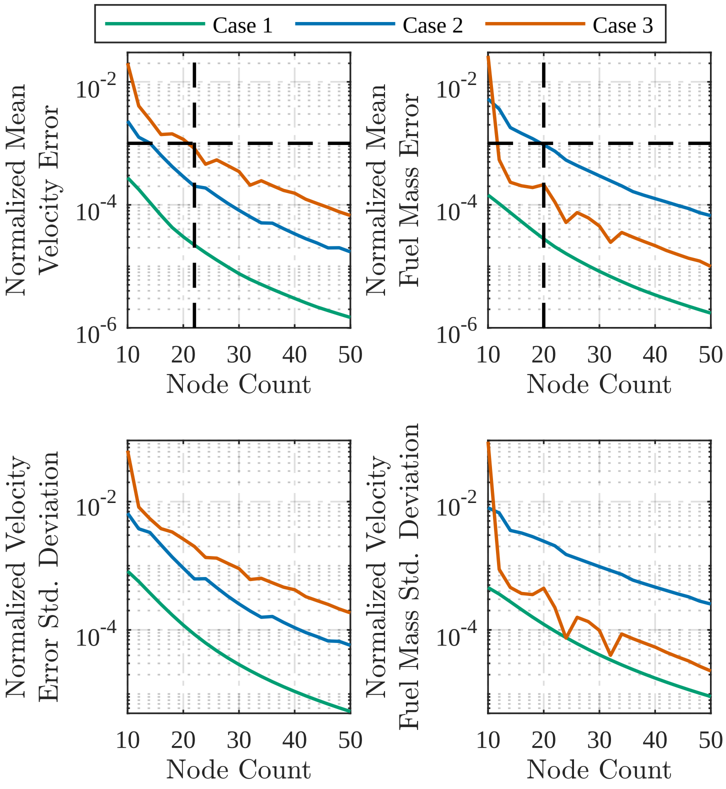


Figure 4.13: The UAV velocity and fuel mass convergence analysis error statistics used to evaluate the number of required discretization steps.

constant velocity and altitude flight condition, does not have significant fast transients, thus it intuitively requires fewer discretization steps. In contrast, the dive and climb segments are significantly more transient, thus requiring more discretization steps. In particular, Figure 4.13 indicates that dive (Case 2) and climb (Case 3) segments require at least 20 and 22 discretization steps, respectively, to meet the accuracy specification. These values are reported in Table 4.1.

4.4.3 Results

Each design study in List 4.4 is solved using (4.26) and (4.27) with the design variables in Table 4.2. The quantitative results optimization results are listed in Table 4.3 and the normalized mission time performance is illustrated in Figure 4.14. Recall that the aircraft and its controllers are designed to reduce the total mission time. The baseline design completes the mission in 2,438s, while all other optimized designs complete the mission 5-35% faster. Intuitively, the simultaneous open-loop (SOL) design offers the greatest improvement of 35.6% in performance because it can directly choose the optimal open-loop control behavior. While high performing, an open-loop control system cannot be put on a physical

Table 4.3: A comparison of the optimal UAV and control design parameters and performance for each design study.

	BASE	SOL	SQ	SCL	PCL	CCL
Motor Constant [V-s/rad]	0.1238	0.0870	0.0870	0.0926	0.0988	0.1238
Motor Resistance [Ω]	0.0292	0.0158	0.0158	0.0171	0.0186	0.0292
Series Cells [-]	16	20	20	20	20	16
Parallel Cells [-]	7	9.3	9.3	9.8	19.3	7
Equivalence Factor [-]	18.0	-	16.6	16.9	18.0	19.3
P Gain [-]	0.0371	-	0.001	0.001	0.0371	0.0335
I Gain [-]	0.0020	-	0.0065	0.010	0.0020	.0042
GTOW [kg]	50.5	53.7	53.7	54.2	50.5	50.5
Mission Time [s]	2438	1570	1604	1589	1890	2278
Improvement [%]	-	35.6	34.2	34.8	22.5	6.60

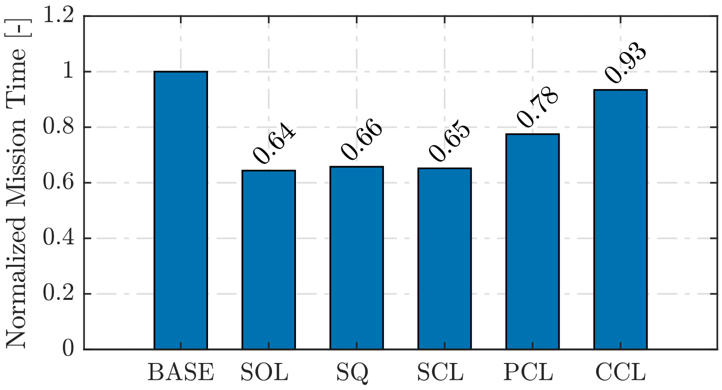


Figure 4.14: The optimal mission time for each of the design cases. The times are normalized against the BASE (baseline) design.

system, so the SOL design serves as the *best case* benchmark. The sequential optimal design (SQ) also demonstrates good performance, however, it does perform worse than the simultaneous closed-loop optimal design (SCL). This demonstrates that designing the plant and closed-loop controller at the same time does yield improved system performance. In contrast, both siloed design approaches (PCL) and (CCL) provide comparatively less improvement in comparison to the baseline because they do not account for the coupling between plant and control designs.

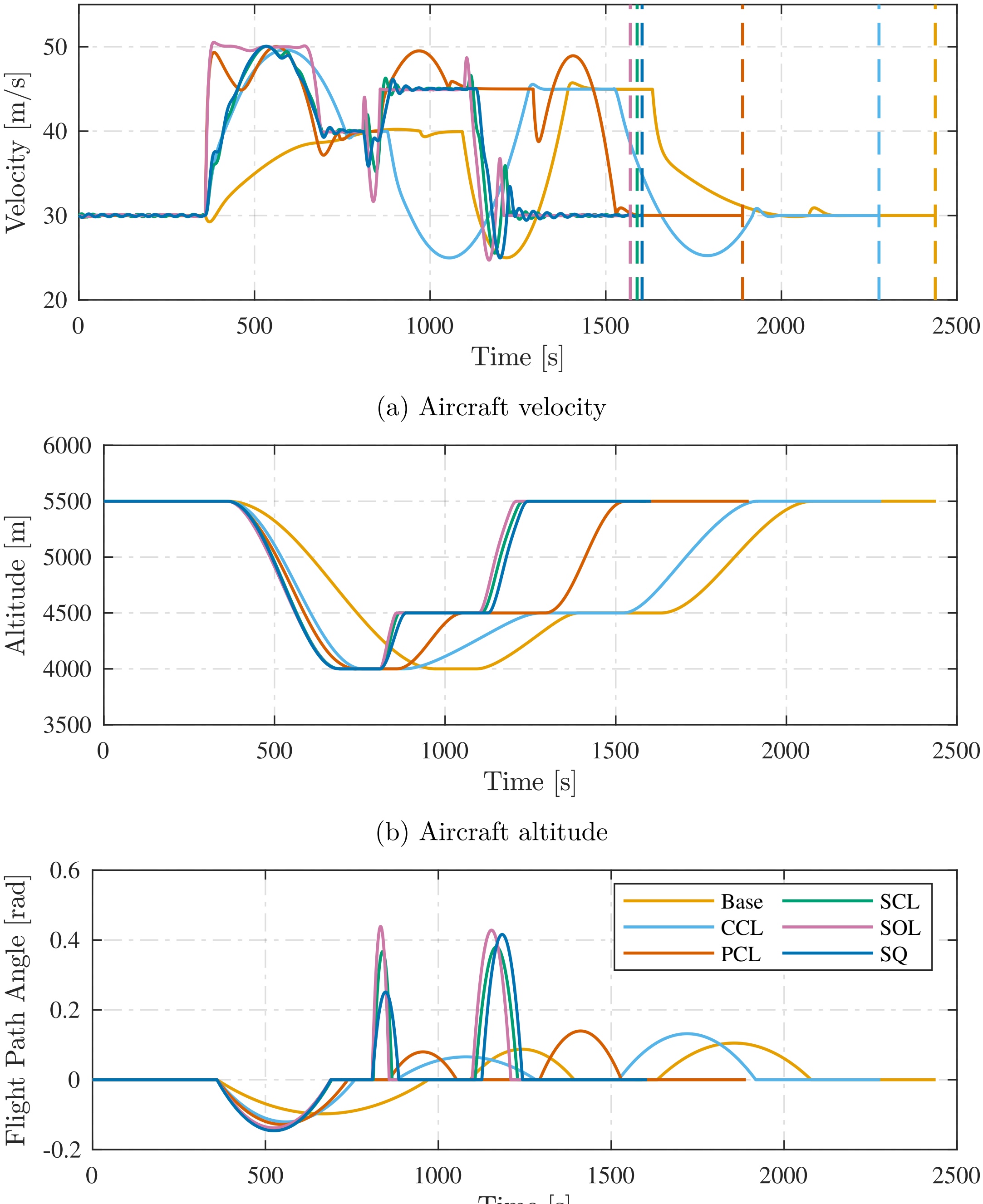
Key state and input trajectories are illustrated in Figure 4.15. The velocity and altitude profiles (Figures 4.15a-4.15b) show that each design successfully completes the mission requirements. The key to completing the mission quickly is to have aggressive dives and climbs, as illustrated in Figure 4.15c, which shows that the high performing SCL and SQ designs reach significantly greater flight path angles. While a greater flight path angle is preferred, it may result in a stall condition if not enough thrust is produced. As such, the vehicle's instantaneous thrust-to-weight ratio is illustrated in Figure 4.15d, which shows that the high performing designs nearly double the maximum thrust-to-weight ratio of the other designs. To understand the decisions made to increase the thrust-to-weight ratio, investigate the thrust equation (4.18). Thrust is increased by increasing the propeller speed a_{11}^2. By decreasing the motor constant, the motor's back-emf decreases, allowing the motor to spin faster at a given terminal voltage. Similarly, decreasing the motor's resistance increases the efficiency, thus improving the conversion from electrical to mechanical power.

However, decreasing the motor resistance increases the motor mass (4.25d), thus decreasing the thrust-to-weight ratio. Systematically evaluating that trade-off is one of the key merits of using an optimization-based design approach. According to Table 4.3, the SCL, SQ, and PCL designs improved the system performance by decreasing both the motor constant and resistance values.

By increasing the number of series battery cells, the system's operating voltage increases, thus increasing the voltage that can be applied to the prop. motor and the resulting prop. speed. At the expense of increased system mass, all design studies maximized the number of series battery cells to 20 cells except for the CCL study, which could not change plant parameters (Table 4.3). However, there is some variance in the choice of number of parallel battery cells. The SCL and SQ designs select approximately 9-10 parallel cells, while the PCL design selects 19.3. Increasing the battery voltage increases performance only if the high battery voltage can be sustained. The PCL design, which can only optimize plant parameters, can sustain the system voltage by increasing the number of parallel cells at the expense of additional weight. In contrast, the SCL and SQ designs have additional control design freedom, allowing those designs to complete the mission faster and therefore require less battery energy and weight.

The CCL design increases the ECMS equivalence factor to better sustain the system voltage. While this is a fair comparison because the baseline and CCL designs have the same battery size, the same direct comparison is improper for the SCL and SQ designs that have different battery sizes. However, by investigating Figures 4.15e-4.15f, it is clear that both the SCL and SQ design have charge depleting ECMS designs. Although the PI control gains do not impact thrust like the other design variables, they do strongly affect the system's transient response. By increasing the I gain and decreasing the P gain, the system response is more aggressive. As a result, the systems are more responsive but with some oscillatory velocity behavior.

Notably, the SCL and SQ closed-loop controllers behave similarly to SOL open-loop control action (Figures 4.15g-4.15h). For the ECMS control design, this is not unexpected because it has been shown that a well tuned ECMS controller can approximate the open-loop optimal control behavior [111]. Additionally, the SCL and SQ designs are coordinated by the autonomous navigation system that commands the aircraft velocity reference as an open-loop control action. Because the controllers are optimally tuned, they are able to track

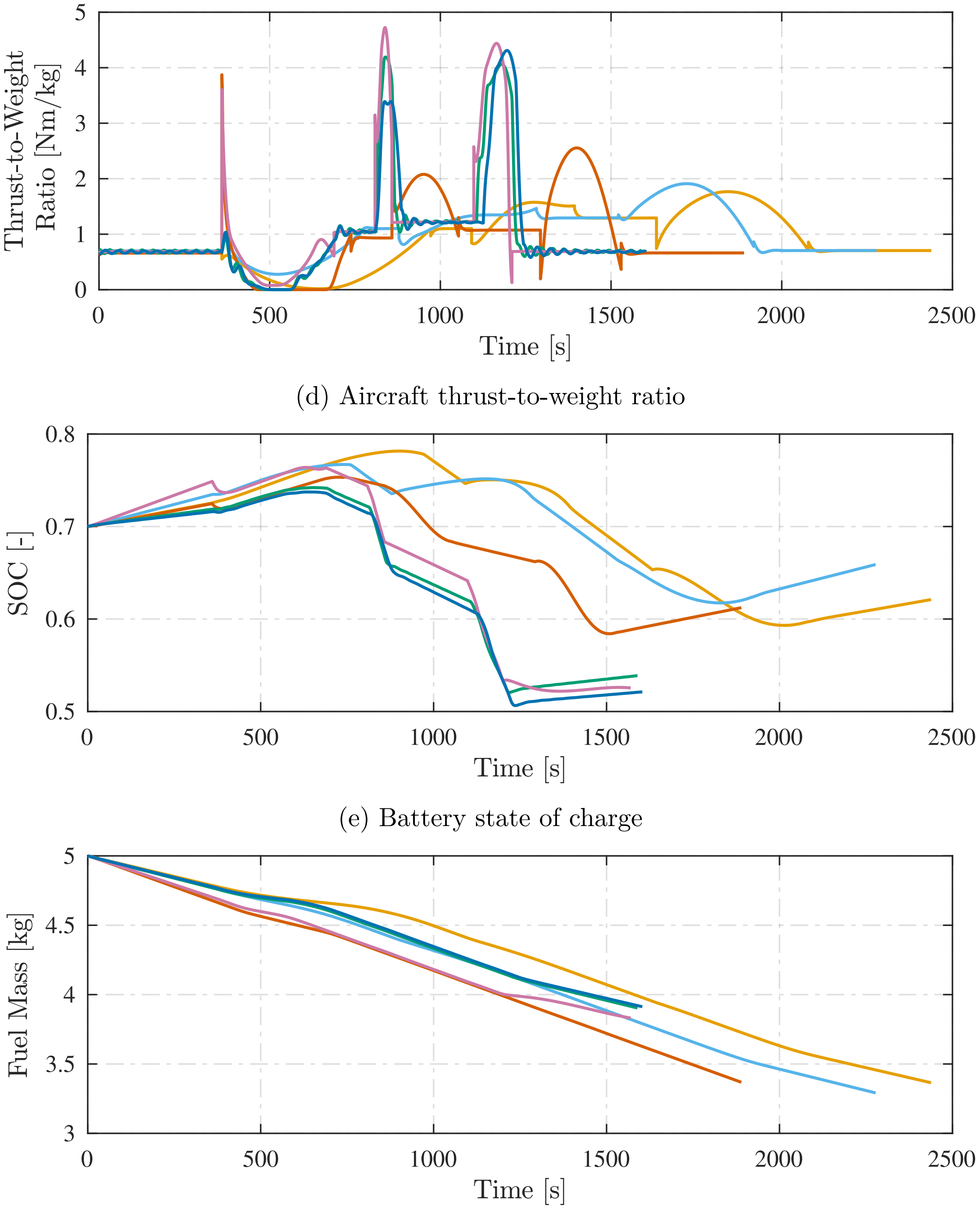


(a) Aircraft velocity

(b) Aircraft altitude

(c) Aircraft flight path angle

Figure 4.15: Selected optimized trajectories for the hybrid UAV design studies: (a) aircraft velocity, (b) aircraft altitude, (c) aircraft flight path angle, and (d) aircraft thrust-to-weight ratio, (e) battery SOC, (f) fuel mass, (g) PI control command, (h) genset command.

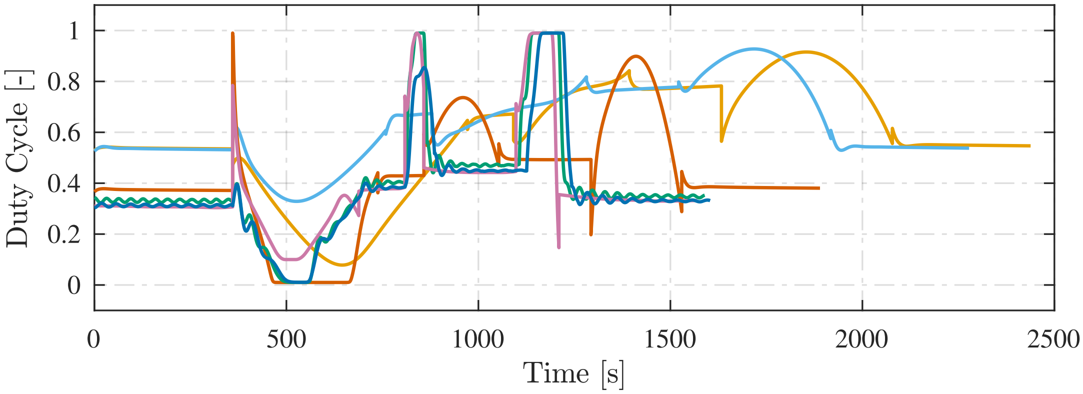


(d) Aircraft thrust-to-weight ratio

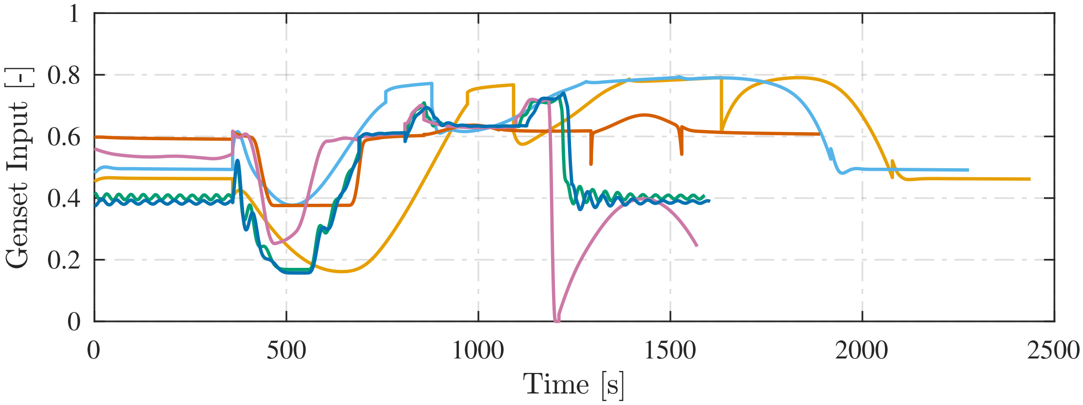
(e) Battery state of charge

(f) Fuel mass

Figure 4.15 (cont.): Selected optimized trajectories for the hybrid UAV design studies: (a) aircraft velocity, (b) aircraft altitude, (c) aircraft flight path angle, and (d) aircraft thrust-to-weight ratio, (e) battery SOC, (f) fuel mass, (g) PI control command, (h) genset command.



(g) PI controller duty cycle command



(h) ECMS input command

Figure 4.15 (cont.): Selected optimized trajectories for the hybrid UAV design studies: (a) aircraft velocity, (b) aircraft altitude, (c) aircraft flight path angle, and (d) aircraft thrust-to-weight ratio, (e) battery SOC, (f) fuel mass, (g) PI control command, (h) genset command.

the optimal velocity reference trajectory well and perform similarly to the SOL design.

Using the full optimization results, the velocity and fuel mass error statistics are computed and tabulated in Table 4.4. The average error is approximately 0.1% for each case, which meets the accuracy criteria defined in Section 4.4.2. Additionally, the computational metrics for each closed-loop optimization study is provided in Table 4.5. Given the large number of design variables, constraints, and long computation times, it should be clear that the proposed convergence analysis is a valuable prepossessing step for this scale of problem that saves time while quantitatively evaluating the design accuracy.

Table 4.4: The optimization accuracy error statistics for aircraft velocity and fuel mass states.

	Velocity Error		Fuel Mass Error	
	Mean [%]	Std. Dev. [%]	Mean [%]	Std. Dev. [%]
BASE	0.01	0.03	0.01	0.04
SQ	0.08	0.32	0.09	0.29
SCL	0.02	0.06	0.02	0.11
PCL	0.05	0.47	0.27	1.33
CCL	0.11	0.53	0.12	0.41

Table 4.5: The computation metrics for the closed-loop optimization problems. Note that the optimizer treats the dynamic states as optimization variables.

	SQ	SCL	PCL	CCL
Design Variables	2155	2159	2156	2155
Constraints	3204	3204	3204	3204
Iterations	79	55	46	98
Function Evaluations	98	62	52	196
Gradient Evaluations	79	55	46	98
Solution Time [hr]	5.5	3.8	3.2	6.6

4.5 Conclusion

While electrified aircraft offer a novel suite of impactful capabilities, the inherent complexity and coupling between subsystems and their controllers makes them difficult to design. Using traditional sequential design strategies is possible, but results in underperforming designs inadequate for modern vehicle requirements. Therefore, it is proposed to use co-design methods to design electrified aircraft

The key contribution of this chapter was the introduction of a generalizable and scalable CLCD approach that simultaneously optimizes both the plant and closed-loop controller configuration. While the approach is general, the proposed CLCD problem is introduced with conventional controllers from the literature to illustrate how feedback controllers are integrated into the design process. To investigate system-level design, a convergence analysis technique was introduced that allows designers to evaluate the tradeoff between optimization accuracy and computation time. The capabilities and practicality of the proposed approach was demonstrated through the design of a hybrid UAV and its feedback controllers. In particular, the design study investigated the design of the vehicle's battery, propulsion motor, speed controller, and ECMS controller. The optimization process identified a set of optimal design parameters that increased the vehicle's thrust-to-weight ratio and dynamic response to complete a specified mission in minimum time.

While these results demonstrated improvements in system performance for a single mission, there is significant uncertainty present in practical applications. Therefore, integrating formal robustness metrics and constraints into the optimization is necessary to design a robust plant and closed-loop control design. Additionally, the design process should be iteratively solved for different profiles to identify a set of plants and controllers that are best-suited for a range of missions. Monte Carlo is one approach to meet this need, however, further consideration should be given to more advanced algorithms that scale better to more complex systems. Chapter 3 demonstrated the need to account for multi-domain coupling in control design; coupling that should be considered in a co-design process as well. Similarly, a natural extension is to investigate other aspects of the hybrid aircraft design, such as the airframe or wing, and understand how different objectives and constraints impact the optimal system design.

The case study presented in this chapter focused on combined sizing and control design for a standard series hybrid UAV. For this particular system architecture and mission, an optimal plant and closed-loop control design was identified. However, there are a variety of UAV architectures such as parallel hybrid, power split, all-electric, etc. that could potentially perform better. It is also likely that the optimal system design will also change if the mission requirements are modified. The next chapter will investigate these topics and introduce methods for mission-specific architecture and closed-loop control co-design.

Chapter 5

Plant Architecture and Closed-Loop Control Co-Design

5.1 Background

The trend of electrification has enabled the exploration of novel new aircraft system architectures/topologies with unique capabilities, such as those illustrated in Figure 1.3. While these novel architectures are diverse with different airframe, propulsion, power, and thermal management systems, it is challenging to determine which designs are best or appropriate for a given application because of the numerous design decisions. Therefore, this chapter will introduce methods to more efficiently parse through the large and complex system-level design space.

As illustrated in Chapter 1, system level design can be characterized by topology generation, topology optimization, sizing optimization, and control optimization (Figure 5.1). In general, design decisions are either *continuous*, where a decision is made from a continuum

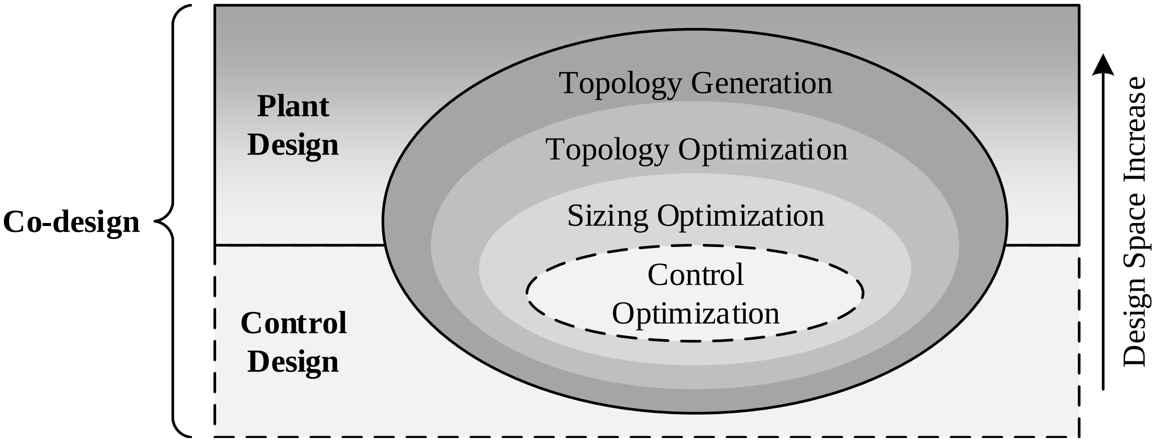


Figure 5.1: Illustration of the plant and control design optimization decisions that comprise the co-design process. The different design disciplines are sized according to the relative size of the design space. Modified from [15].

of options, or *discrete*, where a decision is made from a discrete number of finite options. An example of a continuous decision may be the length of a beam that can vary between 1 and 3 meters, while a discrete decision is the choice between either a 1, 2, or 3 meter beam. Discrete design decisions are most often considered when integrating commercial-off-the-shelf (COTS) parts or discrete actuators (e.g. valves) into the decision-making process.

While many efforts have focused on the plant sizing and control optimization co-design problem (Table 1.1), fewer studies have focused on integrating architecture optimization into the design routine. Of these few studies, some have focused explicitly on topology generation [60–62] while others have focused on topology optimization [16, 18, 63]. Architecture optimization is particularly challenging because it is an entirely discrete design problem, often solved using a limited set of tools. To solve these problems for energy system design, the current state of the art is through an exhaustive search; evaluate the performance of every design and select the best option. While a valid approach, the increasingly large topology design space makes it computationally intractable to evaluate the performance of numerous possible system configurations. In sizing and control design, discrete decisions can be handled through a relaxation, where the discrete problem is reposed as a continuous one that can be more efficiently solved with a greater breadth of optimization tools. However, a relaxation approach has not yet been developed for topology design of energy systems.

Therefore, the key contribution of this chapter is a relaxed representation of the combined plant architecture and closed-loop co-design problem. The proposed approach is enabled by the modular and computationally efficient graph-based models (Chapter 2) wherein multiple architectures are encoded into a single model, thus facilitating a continuous search of the relaxed design space. Similarly, this approach also introduces an architecture-adaptive feedback controller that is simply reconfigured to rapidly evaluate the performance of any candidate architecture design. The methods are applied to create a mission-specific aircraft thermal management design. The proposed case study seeks to address the following questions:

Which class of system architectures is best suited for a range of missions?

How is a given topology and its controller optimized for any given mission?

Does relaxing the traditional discrete topology selection problem improve the optimization process?

This chapter is outline as follows. Section 5.2 introduces a generic system-level co-design problem formulation. Section 5.3 establishes the novel relaxed architecture design problem, a representative graph-based model, optimization methods, and the architecture-adaptive closed-loop controller. To illustrate the effectiveness of the novel design approach, the methods are used to design a thermal management system in Section 5.4. The mission-specific optimization results are outlined in Section 5.5, and Section 5.6 concludes the chapter and suggests areas for future work.

5.2 The System-Level Co-Design Problem

Chapter 4 introduced a co-design problem formulation (4.2) for combined sizing and closed-loop control design. This chapter extends that formulation to include architecture/topology optimization in the decision-making process

$$\begin{aligned}
&\text{minimize} && J && \text{(5.1a)}\\
&\text{with respect to} && \rho, \lambda_p, \lambda_c &&\\
&\text{subject to} && &&\\
& && \dot{x} = f_x\left(x, a, u, d, \lambda_p, \rho\right) && \text{(5.1b)}\\
& && 0 = f_a\left(x, a, u, d, \lambda_p, \rho\right) && \text{(5.1c)}\\
& && \dot{k} = f_k\left(x, a, k, \lambda_p, \lambda_c, \rho\right), && \text{(5.1d)}\\
& && u = f_u\left(x, a, k, \lambda_p, \lambda_c, \rho\right), && \text{(5.1e)}\\
& && \rho \in \mathcal{A} && \text{(5.1f)}\\
& && g_j\left(x, a, u, k, \lambda_p, \lambda_c\right) \leq 0, \forall j \in \left[1 : N_g\right], && \text{(5.1g)}\\
& && h_k\left(x, a, u, k, \lambda_p, \lambda_c\right) = 0 \forall k \in \left[1 : N_h\right], && \text{(5.1h)}
\end{aligned}$$

where J is the objective function to be minimized, ρ, λ_p, and λ_c are the topology, plant, and control design variables, respectively, x and a are the system dynamic and algebraic states, respectively, k are the control dynamics, u are the control inputs, d are the system disturbances, and $\mathcal{A}$ is the feasible topology design space. Note that the model's time dependency is implied. The functions f_x, f_a, f_k, and f_u are the dynamic plant state, algebraic plant state, dynamic controller state, and control input equations, respectively. Thus, the plant equations are represented by (5.1b)-(5.1c) and the controller equations are represented by (5.1d)-(5.1e). The constraint functions g_j and h_k represent inequality and equality constraints, respectively.

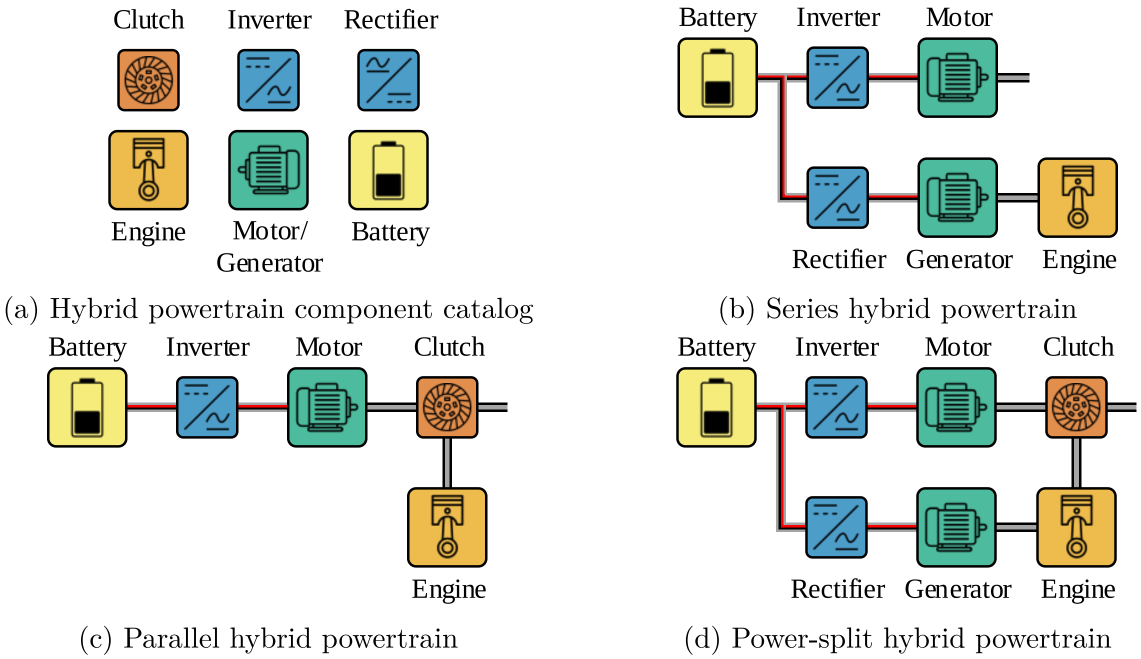
Equation (5.1) has a few differences from (4.2). The plant and controllers equations are parameterized as a function of the architecture design variables ρ to capture the relationship between system behavior and topology. While it is reasonable to suggest that only the plant equations change as a function of the plant topology, it is necessary to consider structural controller changes as well. Different plants have different control needs, and thus the control system will also change as a function of the topology design variables. This formulation also poses the plant equations as a nonlinear DAE system instead of a graph-model. While a graph-based representation will become useful in Section 5.3.2, this problem definition is designed for a generic plant and control model. Lastly, constraint (5.1f) guarantees that a candidate topology design is feasible. The following section will describe the novel methods used to define and solve this optimization problem for architecture and closed-loop control design.

5.3 Design Methodology

As illustrated in Figure 5.1, system level design comprises 4 steps: topology generation, topology optimization, sizing optimization, and control optimization. While sizing optimization is necessary for system level design, those methods were discussed in Chapter 4 and will be omitted here. Therefore, the following sections will describe methods for topology generation, topology optimization, and control optimization. In particular, the novelty of the proposed methods is the relaxation of the topology design problem and the formulation of an architecture-adaptive feedback control law that enables the application and development of efficient gradient-based optimization algorithms.

5.3.1 Architecture Generation

The architecture generation problem is primarily focused on identifying and forming the design space of all unique feasible architectures $\mathcal{A}$, which is described as a unique arrangement/connection of components from a *component catalog*. Figure 5.2 illustrates 3 unique hybrid powertrain architectures (series, parallel, power-split) that are generated from a component catalog of batteries, engines, clutches, power electronics, and motors. While the illustration only demonstrates architecture design decisions, sizing decisions, like those discussed in Chapter 4, could be incorporated as well. In the context of energy system design, the architecture design problem is *discrete* (or *integer*) wherein a candidate design is part of a finite discrete set of feasible topologies. The discrete problem contrasts *continuous* design problems, in which all feasible designs are part of a continuous set. In general, continuous design problems are easier to solve using an assortment of optimization tools, such as gradient-



(a) Hybrid powertrain component catalog

(b) Series hybrid powertrain

(c) Parallel hybrid powertrain

(d) Power-split hybrid powertrain

Figure 5.2: A hybrid powertrain component catalog (a) and three unique powertrain architectures: (b) series hybrid, (c) parallel hybrid, and (d) power-split hybrid.

based optimization methods. Conversely, discrete design problems are NP-complete [118], so a feasible solution can be easily verified, but it is challenging to efficiently identify a solution. Because topology design problems are discrete, they are particularly challenging to solve, a problem that becomes more difficult when the design process considers continuous sizing and control decisions too. Problems that consider both discrete and continuous design decisions are often referred to as *mixed-integer* design problems.

The most common discrete optimization technique is an exhaustive search, where every unique architecture is evaluated, and the best design is selected. However, exhaustive search is not scalable and its computational expense limits it practicality. To address this limitation, other discrete optimization methods have been developed that rely on heuristics or approximations to more efficiently search the discrete design space. While such techniques work well on some problems, they are not guaranteed to outperform an exhaustive search (it is NP-complete). Despite the lack of guarantees, there is still interest in using alternative discrete optimization techniques to more efficiently search the design space.

Many of the alternative discrete optimization techniques, such as branch and bound, annealing, penalty, or surrogate methods [118–120], use a relaxed representation of the discrete design space to improve search efficiency. Relaxation is the process by which a discrete set $\{1, 2, 3, \ldots, N\} \subset \mathbb{Z}$ is converted into a continuous set $[1 : N] \subset \mathbb{R}$. While it is possible to relax an integer discrete set, it may be more appropriate to relax a *binary* discrete set $\{0, 1\} \subset \mathbb{Z}$ where topology variables take on only one of two values: 0 or 1. For topology optimization of energy systems, it is recommended to use a binary set representation. An integer set representation is possible, but the ordering of the candidate design architectures can negatively impact the search of the design space.

The discrete topology design space must be formulated before it can be relaxed into the continuous topology design space. The following sections describe these processes and provides a notional example of the proposed methods. Note that the unique contribution of this section is the formulation of the binary and continuous topology design spaces.

5.3.1.1 Integer Topology Design Space

Recall that a unique architecture is described by a unique set of connections between a unique set of components, and the set of all unique architectures is the discrete topology design space. Unique architectures can be built using either generative [58,59] or exhaustive [60–62] construction methods. Generative techniques use rules to construct potential system architectures, while exhaustive techniques enumerate all possible configurations and then impose rules posteriori to evaluate structural feasibility. As the name suggests, exhaustive techniques identify all possible system topologies, while generative techniques may not. This works uses the perfect matching enumeration approach from [60] for architecture generation because it guarantees a complete representation of the topology design space and is provided as an open-source tool. It has also been used in other energy system design problems [17]. It should be clear that the perfect matching approach is not a contribution of this dissertation, and that an interested reader is referred to [60, 61] for more information. A brief overview of the approach is described below.

The perfect matching approach is input a component catalog C defined by the list of components L and a set of *structural constraints*. Each component can have R replicates and P ports, where ports are interfaces that connect components. Structural constraints impose compatibility between components. For example, a propeller could be mounted on a motor, but not on a battery pack. Operational constraints, such as a topology's ability to keep a temperature below a specified threshold, would also be relevant to consider at this stage. However, evaluating those constraints may be challenging or computationally expensive at this step in the design process. Conversely, structural constraints can be more efficient to evaluate.

The perfect matching is input the component catalog and structural constraints, and the algorithm enumerates every system topology. Very efficient and parallelizable evaluations of structural constraints allows the algorithm to enumerate millions of possible design architectures on the scale of seconds. The approach is based in graph-theory, so efficient isomorphism checks can evaluate uniqueness of a given topology. The output of the perfect matching algorithm is a set of adjacency matrices that define the discrete topology design space,

$$\mathcal{A} \coloneqq \{A_1, A_2, \dots, A_N\} = \{A_i \mid i \in \{1, 2, 3, \dots, N\}\} \tag{5.2}$$

where N is the total number of unique system architectures. In graph theory, the adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N_c \times N_c}$ is a symmetric metric that represents the connectivity between components in a system,

$$a_{ij} = \begin{cases} 1 & \text{if component } L_i \text{ is connected to component } L_j, \\ 0 & \text{else,} \end{cases} \tag{5.3}$$

where N_c is the number of components in the candidate architecture. The adjacency matrix for the example shown in Figure 5.3 is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{5.4}$$

The discrete design space (5.2) is currently an integer set because a candidate topology A_i is identified by choosing an integer variable $i \in \{1, 2, 3, \dots, N\} \subset \mathbb{Z}$. To relax the design problem, the integer discrete set must first be converted to a binary discrete set.

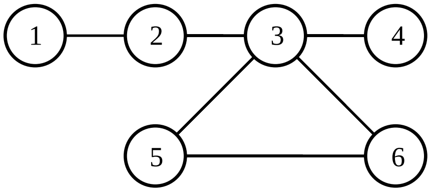


Figure 5.3: A notional graph used to show the construction of an adjacency matrix.

5.3.1.2 Binary Topology Design Space

To convert the integer design space into a binary space, a similar approach to [121] is adopted by converting the adjacency matrix to a bitstring representation. Recognize that each integer topology A_i comprises binary variables 0 or 1. Also, note that A_i is symmetric, so all the topology information is encoded into only half of the full matrix (upper or lower triangular half). Without loss of generality, assume that none of the candidate topologies have components that connect to themselves or have self loops (i.e. the diagonal elements of A_i are zero: $a_{ij} = 0$ for $i = j$). Let $\mathrm{P}_i \in \mathbb{R}^{N_c(N_c-1)/2}$ be the column vector of all elements from the lower triangular partition, excluding the diagonal, of A_i. Therefore, an equivalent representation for the integer topology design space is

$$\mathcal{A} \coloneqq \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_N\} = \{\mathrm{P}_i \mid i \in \{1, 2, 3, \ldots, N\}\}\,. \tag{5.5}$$

The binary topology design space is formulated by concatenating each candidate topology vector P_i into a matrix and introducing the binary topology selection variable $\lambda \in \mathbb{R}^N$

$$\mathcal{A} \coloneqq \left\{\rho \mid \rho = \begin{bmatrix}\mathrm{P}_1 & \mathrm{P}_2 & \cdots & \mathrm{P}_N\end{bmatrix}\lambda, \sum \lambda = 1, \lambda \in \{0, 1\}\right\}. \tag{5.6}$$

In this representation $\rho \in \mathbb{R}^{N_c(N_c-1)/2}$ are the topology variables describing connections between components and λ is a vector where a single element has value 1 and all other elements are 0. When λ is selected in this fashion, ρ is also a binary vector that directly corresponds to one of the topologies P_i.

While (5.2), (5.5), and (5.6) are all equivalent representations of the discrete topology design space, they uniquely encode the design space as an integer set of adjacency matrices (5.2), an integer set of topology vectors (5.5), or as a binary set with a topology matrix (5.6). In (5.2) and (5.5), the design problem has a scalar design variable i that can be any of N integer values while in (5.6), the design problem has a N-length vector design variable λ where each element can be a binary value (0 or 1). The binary design space is well suited for relaxation.

5.3.1.3 Continuous Topology Design Space

Relaxing the binary set into a continuous set requires removing the binary constraint on λ

$$\mathcal{A}_c \coloneqq \left\{ \rho \mid \rho = \begin{bmatrix} \mathrm{P}_1 & \mathrm{P}_2 & \cdots & \mathrm{P}_N \end{bmatrix} \lambda, \sum \lambda = 1, \lambda \geq 0 \right\}, \tag{5.7}$$

where $\mathcal{A}_c$ is the continuous topology design space. This notation is otherwise known as the vertex representation (V-rep) for convex sets. Note that the only difference between (5.6) and (5.7) is the constraint on the topology selector variable λ. In the continuous representation, λ can vary continuously, which results is a continuous set of feasible system topologies ρ. In the discrete representation, ρ was binary and indicated if components were connected $\rho = 1$ or not $\rho = 0$. In the continuous representation, the continuous ρ can be interpreted as facilitating a partial connection between components. In the context of optimization, the optimizer could select intermediate values of ρ to determine the connections that yield the best performance. Note the following definitions

Definition 5.1 (Continuously feasible)**.** A candidate topology ρ is *continuously feasible* for a given λ if $\rho \in \mathcal{A}_c$ as defined in (5.7).

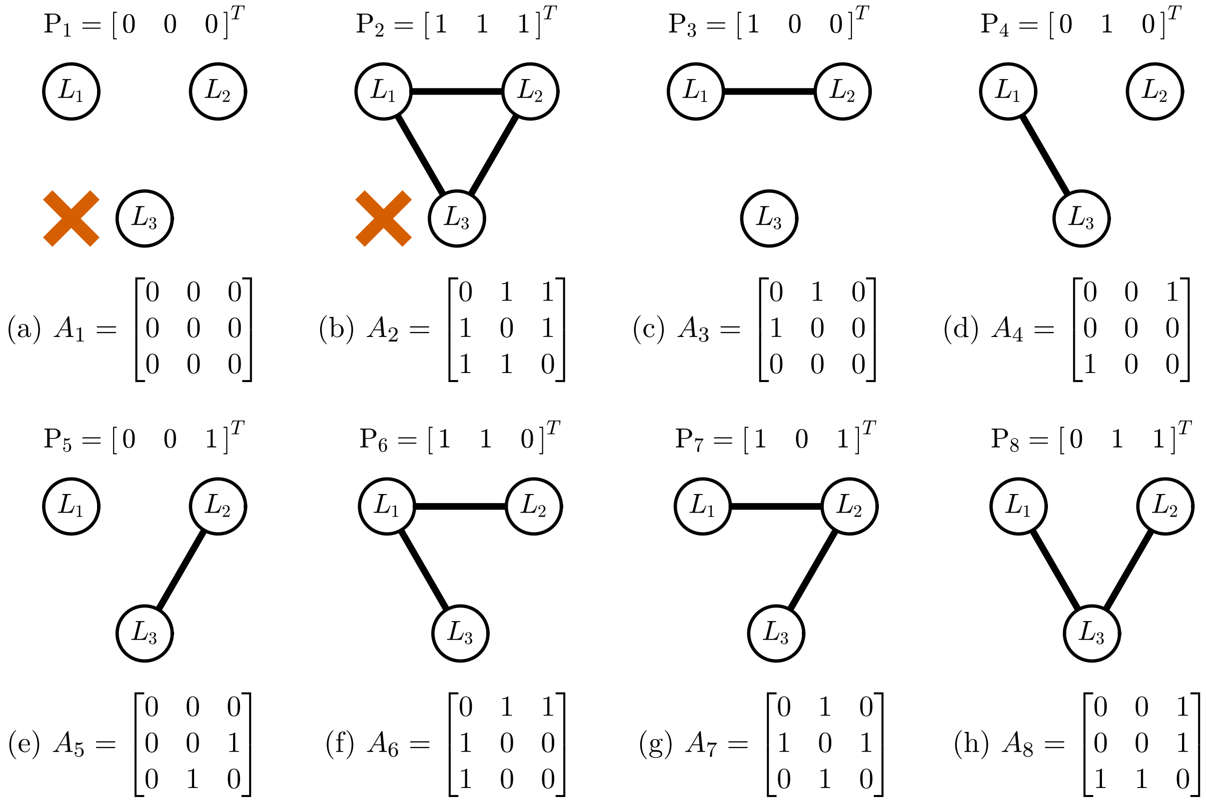
Definition 5.2 (Discretely feasible)**.** A candidate topology ρ is *discretely feasible* for a given λ if $\rho \in \mathcal{A}$ as defined in (5.6). Note that all discretely feasible topologies are also continuously feasible $\mathcal{A} \subset \mathcal{A}_c$.

5.3.1.4 Notional Low-Dimensional Example

The following example is constructed to illustrate how the topology design space is generated. Consider a component catalog of 3 unique components L_1, L_2, and L_3 without replicates $R = 0$ where each component may have up to 2 ports. In this example, the components are arbitrary and do not represent any physical system. Apply the following structural constraints:

1. There must exist at least 1 connection between components.

2. No more than 2 connections can be present.



$\mathrm{P}_1 = [0 \quad 0 \quad 0]^T$

(a) $A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$\mathrm{P}_2 = [1 \quad 1 \quad 1]^T$

(b) $A_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

$\mathrm{P}_3 = [1 \quad 0 \quad 0]^T$

(c) $A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$\mathrm{P}_4 = [0 \quad 1 \quad 0]^T$

(d) $A_4 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$\mathrm{P}_5 = [0 \quad 0 \quad 1]^T$

(e) $A_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$\mathrm{P}_6 = [1 \quad 1 \quad 0]^T$

(f) $A_6 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$\mathrm{P}_7 = [1 \quad 0 \quad 1]^T$

(g) $A_7 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

$\mathrm{P}_8 = [0 \quad 1 \quad 1]^T$

(h) $A_8 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Figure 5.4: Eight unique system architectures for the example architecture generation problem. The crosses indicate the designs (a) and (b) are structural infeasible. Each architecture's adjacency matrix A and topology variables ρ are provided.

The perfect matching approach enumerates and generates the 8 candidate system topologies, illustrated in Figure 5.4. However, structural constraints 1 and 2 are applied to reduce the 8 candidate topologies to 6 because architecture 1 (Figure 5.4a) contains zero connections and architecture 2 (Figure 5.4b) contains 3 connections.

The lower triangular elements of each adjacency matrix are vectorized and the discrete topology design space is represented as

$$\mathcal{A} = \{\mathrm{P}_i \mid i \in \{3,4,5,6,7,8\}\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\}. \tag{5.8}$$

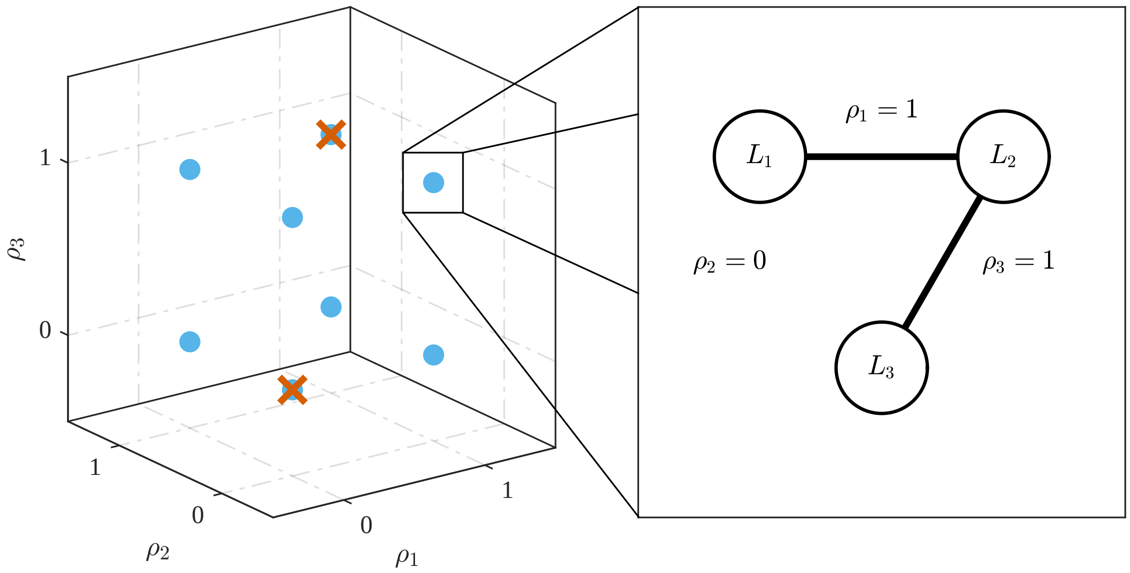


Figure 5.5: A visualization of the 3-dimensional discrete topology design space. The structurally infeasible architectures are crossed out. The topology at $\rho = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T$ is shown.

The discrete design space is visualized in Figure 5.5 where each feasible topology is represented by a single point in a 3-dimensional space, and the structurally infeasible designs are crossed out. Relax the design space by concatenating the topology design vectors P_i according to (5.7)

$$\mathcal{A}_c = \left\{ \rho \mid \rho = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \lambda, \sum \lambda = 1, \lambda \geq 0 \right\} \tag{5.9}$$

where any continuously feasible topology is generated through the choice of a valid λ. The continuous topology design space is visualized as the interior of the convex hull of the discrete topology design space (Figure 5.6). Three examples of continuously feasible topologies are visualized in Figure 5.7 where the candidate topologies marked by a ■ and ▲ in the interior of the topology design space yield partial connections between components represented by the shaded lines between vertices. Additionally, the candidate topology marked by the ★ is at the vertex of the continuous design space, which indicates a discretely feasible topology.

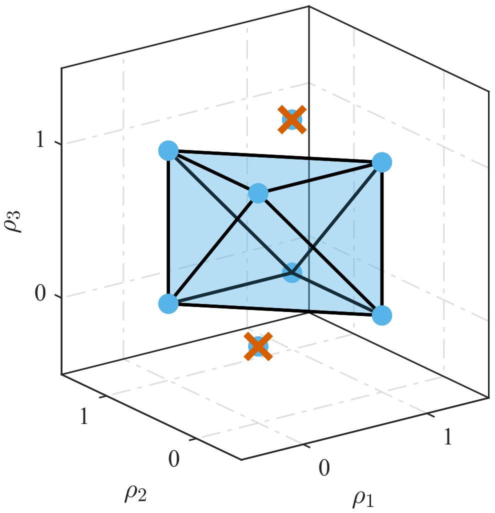


Figure 5.6: A visualization of the 3-dimensional continuous topology design space as the convex hull of the discrete topology design space.

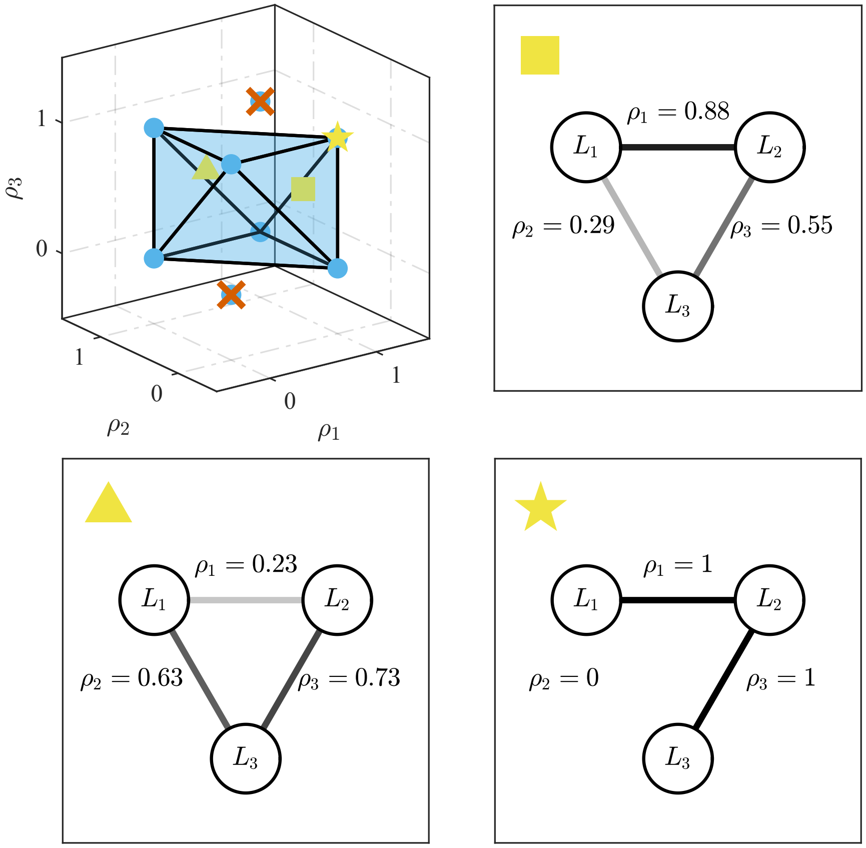


Figure 5.7: A visualization of 3 topologies in the continuous design space. The topologies marked by a ■ and ▲ are continuously feasible while ★ is discretely feasible.

5.3.2 Superset Model Generation

If an exhaustive search is used to solve the topology design problem, N unique plant models could be constructed and optimized. However, the proposed relaxed design approach requires a single plant that can be continuously varied to reflect the continuous topology design space. Therefore, this dissertation adopts a superset modeling approach, inspired from [18], where the dynamics for every continuously feasible topology are captured in a single monolithic model. Graph-based models, which are modular, are well suited for this application because the component connections described in Section 5.3.1 are analogous to component connections in the graph-based modeling framework. Therefore, a single graph-model containing all feasible components connections can be constructed, and the topology design variables ρ can be used to activate $\rho_i = 1$ or deactivate $\rho_i = 0$ the connections to represent a continuum of system dynamics.

As presented in [91], a system graph model comprises various modular component graph models that are connected according to a user defined specification. In particular, graph edges, which capture energy transfer, represent the connection between two energy storage elements. If two components are not connected, energy cannot be transferred and vice versa. This is illustrated by the notional example in Figure 5.8 from [91] where the two component models are connected along edges 3 and 1, respectively, resulting in a system where edge 3 represents the energy transfer between the two components. In the context of continuous topology design, the existence of edge 3 can vary between $\rho \in [0, 1]$ depending on whether the two component are or are not connected (e.g. grey connections in Figure 5.7). All other edges in Figure 5.8 are always present because they represent energy transfer internal to the components and not external connections between components.

A superset graph model captures all possible connections between components, where each element of the topology design vector ρ_i is associated with the connection/energy transfer i between components. The graph-based model equation (2.9) can be augmented with the topology design variables. Recall that $\rho \in \mathbb{R}^{N_c(N_c-1)/2}$ where $N_c\left(N_c - 1\right)/2$ is the total number of component connections and $N_{e,i}$ is the total number of graph model edges. Without loss of generality, let edges e_j for $j \in [1 : N_c\left(N_c - 1\right)/2]$ be the edges associated

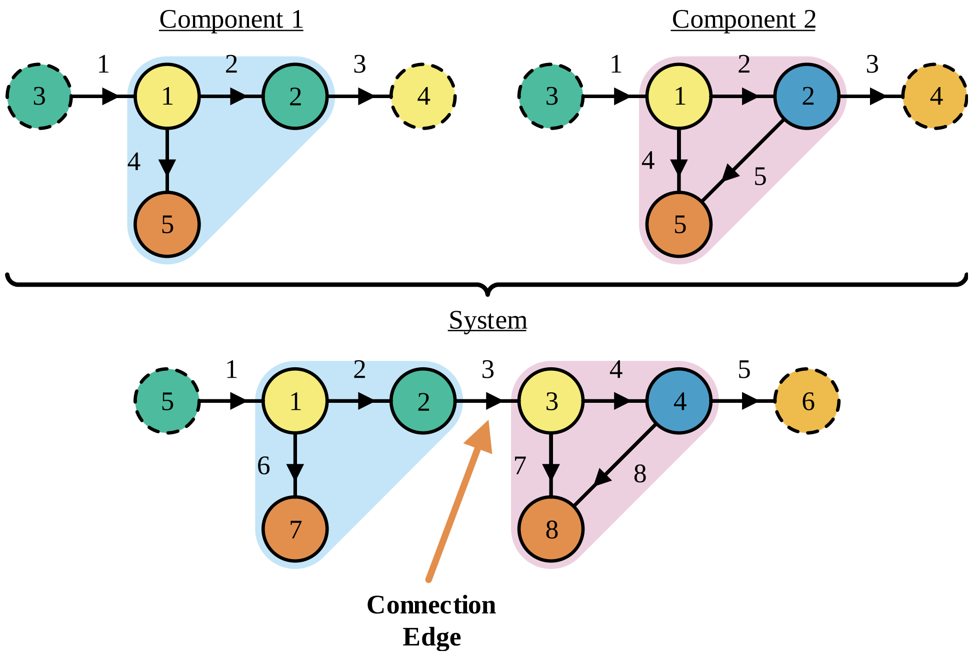


Figure 5.8: An example connection between two graph model components into a graph model system. Components 1 and 2 are connected along edges 3 and 1, respectively, resulting in a system model where 3 represents the connection edge.

with connections ρ_i. Similarly, let edges e_j for $j \in [N_c(N_c-1)/2+1 : N_{e,i}]$ be all other edges not associated with component connections. Therefore, the superset graph model is given by

$$\begin{bmatrix} C_d\dot{x} \\ 0 \end{bmatrix} = -\begin{bmatrix} \bar{M}_d \\ \bar{M}_a \end{bmatrix} \begin{bmatrix} \text{diag}\,(\rho) & 0 \\ 0 & I_{N_{e,i}-N_{nc}} \end{bmatrix} P(x,a,x^e,u,\lambda_p) + \begin{bmatrix} D_d \\ D_a \end{bmatrix} P^e \tag{5.10}$$

where diag$\,(\rho)$ is a diagonal matrix of the topology design variables ρ and $I_{N_{e,i}-N_{nc}}$ is the identify matrix of size $N_{e,i}-N_c$. By choice of ρ, the magnitude of the power flows P are modified, where $\rho=0$ implies that the power flow is 0 (i.e. components are not connected) while $\rho=1$ implies full power flow (i.e. components are connected). A superset graph model is illustrated in the case study problem is Section 5.4.

5.3.3 Topology Optimization

As illustrated in Figure 5.1, the topologies can be optimized after they are generated and a model is constructed. An exhaustive search approach could be used to enumerate the performance of every discrete design. However, an exhaustive method is computationally expensive and does not scale well to large design problems. To leverage the novel continuous topology design space representation, a hybrid continuous-discrete search method is proposed to more efficiently search the discrete topology design space.

Because a hybrid optimization does not necessarily guarantee improved performance over an exhaustive approach, both optimization strategies are investigated in this dissertation. The following sections describe the two approaches in more detail.

5.3.3.1 Enumeration

The enumeration approach, which is simple and easy to implement, is commonly applied to discrete problems. It identifies the optimal topology $\rho^* \in \mathcal{A}$ that minimizes an objective function J

$$\rho^* = \arg\min\left(J\left(\rho\right)\right) \tag{5.11}$$

where,

$$\rho^* \in \mathcal{A}$$

The discrete topology design space $\mathcal{A}$ is defined in (5.2), (5.5), or (5.6). In the enumeration algorithm, the integer representation (5.5) is used. The algorithm is as follows:

Enumeration Algorithm

1. **Given:** N candidate topologies $\rho_i \in \mathcal{A}$

2. **For each** candidate topology, evaluate the optimal objective function value $J_i\left(\rho_i\right)$.

3. **Return:** $\rho = \arg\min\left(J\left(\rho\right)\right)$

The exhaustive approach efficiently uses parallel processing techniques. However, for very large problems, the computational impact of parallel processing may be negligible.

5.3.3.2 Distance-Sorted Search

The distance-sorted search algorithm is a hybrid algorithm that solves the continuous topology design problem to formulate a distance metric that represents the relative expected performance of the discrete topologies. Then an exhaustive search is conducted that starts by evaluating the true performance of the highest ranked topology and sequentially evaluating the remaining topologies until a stopping condition is met. The continuous topology problem identifies the optimal continuous topology $\rho_c^* \in \mathcal{A}_c$ that minimizes an objective function J

$$\rho_c^* = \arg\min\left(J\left(\rho_c\right)\right) \tag{5.12}$$

where,

$$\rho_c \in \mathcal{A}_c$$

Various distance metrics can be applied. In this work, the distance metric is the Euclidean distance between the optimal continuous topology ρ_c^* and each discretely feasible topology, $\rho_i \in \mathcal{A}$

$$D_i = \left\|\rho_c^* - \rho_i\right\|_2^2, \tag{5.13}$$

where $D \in \mathbb{R}^N$ is the distance metric vector. Sort D from smallest to largest and let $R \in \mathbb{R}^N$ be the ranked index of each design. For example, suppose that design $i = 627$ has the 10th smallest distance. Then $R_{10} = 627$. Then apply the exhaustive search algorithm, starting with R_1 and sequentially evaluating topologies R_i for $i \in [1 : N]$. The algorithm is as follows:

Distance-Sorted Search Algorithm

1. **Given:** N candidate topologies $\rho_i \in \mathcal{A}$

2. **Evaluate** the optimal objective function value $J\left(\rho_c\right)$ for $\rho_c \in \mathcal{A}_c$.

3. **If:** the optimal continuous design is discretely feasible $\rho_c^* = \arg\min\left(J\left(\rho\right)\right) \in \mathcal{A}$, then
   Return: $\rho_c^* = \arg\min\left(J\left(\rho\right)\right)$
   Else If: $\rho_c^* = \arg\min\left(J\left(\rho\right)\right)$ meets a stopping criterion, then
   Return: $\rho_c^* = \arg\min\left(J\left(\rho\right)\right)$

4. **Compute** the distance metric D and ranked index vector R

5. **For each:** $i \in [1 : N]$
   For each: candidate topology ρ_{R_i}, evaluate the optimal objective function value $J_i\left(\rho_{R_i}\right)$

6. **If:** the stopping criterion is met
   Return: $\rho_{R_i} = \arg\min\left(J\left(\rho\right)\right)$

Multiple stopping criterion can be used. For example, if the evaluated cost function is less than a desired threshold $J\left(\rho\right) < J_{desired}$ (or $\Delta J \leq \varepsilon$), then the algorithm can be stopped. An alternative approach could be to limit the number of searches $N_{search} < N$. It is also common to stop the search if the current best design has not been improved upon over the last N searches. Note that while the distance-sorted search can be solved sequentially, it is recommended to parallelize the search process.

5.3.4 Control Optimization

In most applications, a feedback controller is designed for a specific system, whereas in a co-design approach, the controller is designed at the same time as the system. However, when architecture considerations are integrated into the co-design process, challenges arise because a general controller needs to be developed for a general class of systems instead of just the singular system. Thus, an architecture adaptive control strategy is developed.

In a discrete topology design problem, a unique controller could be synthesized for each discrete topology. However, the proposed continuous topology design problem requires that a controller be synthesized for any of a continuum of continuously feasible architectures. Similar to the superset modeling method, a monolithic control design approach is adopted where a single controller, parameterized as a function of the topology design variables ρ, is synthesized to operate over a continuum of topologies. It also enables the use of efficient gradient-based optimizers because sensitivity information can be computed to understand the impact the topology variables have on the control design.

An architecture adaptive linear quadratic regulator (AALQR) approach is proposed to meet this need. LQR is a common optimal feedback control approach for linear time-invariant systems. Notably, an LQR controller can be efficiently synthesized and is consequently well-suited for integration into a co-design problem [65, 66, 68–71]. While LQR is used in this work, other similar feedback control approaches could be explored as well. The novelty of the proposed AALQR is the sensitivity computations that can be used by a gradient-based optimizer to evaluate the impact of the topology design variables on the control design, and thus the closed-loop system performance. Additionally, systems commonly have input constraints, so an input constrained variation of the AALQR is also presented. The following sections provide a brief review of the LQR problem and its extension to nonlinear systems and the AALQR formulation with input constraints.

5.3.4.1 Linear Quadratic Regulator

The infinite horizon LQR is an optimal control strategy for unconstrained linear time-invariant systems. The infinite horizon LQR problem is commonly posed as

$$\begin{aligned} &\text{minimize} && J\left(u\right) = \int_0^\infty \left(x^T Q x + u^T R u\right) dt && \text{(5.14a)} \\ &\text{with respect to} && u && \\ &\text{subject to} && && \\ & && \dot{x} = Ax + Bu && \text{(5.14b)} \end{aligned}$$

where J is the quadratic cost function, $A \in \mathbb{R}^{N_x \times N_x}$ and $B \in \mathbb{R}^{N_x \times N_u}$ are the linear state space matrices, $Q \in \mathbb{R}^{N_x \times N_x}$ is a symmetric positive semi-definite state weight matrix, and $R \in \mathbb{R}^{N_u \times N_u}$ is a symmetric positive definite input weight matrix. Equation (5.14b) is the linear time-invariant system dynamics where (A, B) are stabilizable and (A, C) are observable. The solution to (5.14) is the unique optimal feedback control law

$$u = -R^{-1}B^{T}Px, \tag{5.15}$$

where P is the unique symmetric positive semi-definite solution to the Algebraic Riccati Equation (ARE)

$$0 = PA + A^{T}P + Q - PBR^{-1}B^{T}P. \tag{5.16}$$

Often, the optimal control law is written as

$$
\begin{aligned}
u &= -Kx, \qquad (5.17a)\\
&\text{where,}\\
K &= R^{-1}B^{T}P. \qquad (5.17b)
\end{aligned}
$$

While LQR techniques are developed for efficient and optimal control synthesis for linear time-invariant systems, they can also be applied to nonlinear systems, though the control solution is no longer globally optimal. Consider a general nonlinear state-space form model.

$$\dot{x} = f\left(x, u, d\right), \tag{5.18}$$

where x is the system state, u is the input, and d are the disturbances. Linearizing (5.18) about an equilibrium condition $\left(\bar{x}, \bar{u}, \bar{d}\right)$ yields a linear state space model (excluding disturbances)

$$
\begin{aligned}
\Delta\dot{x} &= A\Delta x + B\Delta u, \qquad (5.19)\\
&\text{where,}\\
\Delta x &= x - \bar{x},\\
\Delta u &= u - \bar{u}.
\end{aligned}
$$

With a system in the form (5.19), the standard LQR synthesis techniques can be applied. When applying the optimal linear control solution to the nonlinear system, the control action

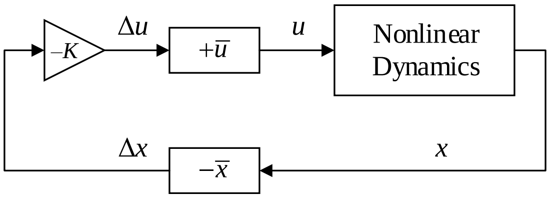


Figure 5.9: The feedback block diagram for an LQR applied to a nonlinear system.

is modified to

$$u = -K\left(x - \bar{x}\right) + \bar{u}, \tag{5.20}$$

where the matrix K is computed as described in (5.17). The general LQR feedback control system diagram is illustrated in Figure 5.9.

5.3.4.2 Architecture-Adaptive LQR

As introduced in Section 5.3.2, the system model may be a function of the system topology variables

$$\dot{x} = f\left(x, u, d, \rho\right), \tag{5.21a}$$

or,

$$\dot{x} = A\left(\rho\right)x + B\left(\rho\right)u \tag{5.21b}$$

Therefore, the LQR problem can be reposed as the AALQR problem

$$\text{minimize} \quad J\left(u\right) = \int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{5.22a}$$

with respect to u

subject to

$$\dot{x} = A\left(\rho\right)x + B\left(\rho\right)u \tag{5.22b}$$

with the corresponding optimal control solution

$$u = -K\left(\rho\right)x, \tag{5.23a}$$

where,

$$K\left(\rho\right) = R^{-1}B\left(\rho\right)^{T}P\left(\rho\right), \tag{5.23b}$$

$$0 = P\left(\rho\right)A\left(\rho\right) + A\left(\rho\right)^{T}P\left(\rho\right) + Q - P\left(\rho\right)B\left(\rho\right)R^{-1}B\left(\rho\right)^{T}P\left(\rho\right) \tag{5.23c}$$

The optimal feedback control gain $K\left(\rho\right)$ is a function of the system topology. The AALQR synthesis technique is a reparameterization of the traditional LQR synthesis approach to be a function of the system topology. Notably, this technique can be applied to a continuum of systems, therefore, it is possible to compute the sensitivity of the feedback control gain with respect to the topology variables $\frac{\partial K(\rho)}{\partial \rho}$. In the context of co-design, the sensitivity of the feedback control gain with respect to the LQR state and input weights $\frac{\partial K(Q)}{\partial \rho}$ and $\frac{\partial K(R)}{\partial \rho}$ can also be computed. As previously mentioned, this parameterization is particularly important for gradient-based optimization to facilitate a more efficient search of the feasible design space. Because these sensitivity evaluations require knowledge of matrix calculus and are lengthy, the computations are provided in Appendix C.

5.3.4.3 Input Constrained AALQR

While LQR identifies an optimal feedback control law, it does not explicitly handle input constraints that may be present in real systems, such as in the case study presented in

Section 5.4. To account for input constraints, the revised AALQR problem is

$$\begin{aligned} &\text{minimize} && J(u) = \int_0^\infty \left(x^T Q x + u^T R u\right) dt && \text{(5.24a)} \\ &\text{with respect to} && u && \\ &\text{subject to} && && \\ & && \dot{x} = A(\rho) x + B(\rho) u, && \text{(5.24b)} \\ & && \underline{u} \leq u \leq \bar{u}, && \text{(5.24c)} \\ & && M(\rho) u = 0 && \text{(5.24d)} \end{aligned}$$

where $\underline{u}$ and $\bar{u}$ are lower and upper input constraints, respectively, and $M(\rho) \in \mathbb{R}^{m \times N_u}$ is the linear equality constraint matrix that may change as a function of the system topology. Note that $m < N_u$. Constraints (5.24c) and (5.24d) are input inequality and equality constraints, respectively. Unlike the LQR and AALQR problems, there does not exist an explicit solution to (5.24). However, (5.24) can be simplified to approximate the optimal control solution.

The equality constraints are eliminated using the process outline in Section 10.1.2 in [122]. To eliminate the equality constraints, identify a matrix $F(\rho) \in \mathbb{R}^{N_u \times (N_u - m)}$ whose range in the nullspace of $M(\rho)$ (i.e. $F(\rho) = \text{null}(M(\rho))$ such that

$$\{u \mid M(\rho) u = 0\} = \left\{F(\rho) z + \hat{u} \mid z \in \mathbb{R}^{N_u - m}\right\} \tag{5.25}$$

where $\hat{u}$ is any particular solution to $M(\rho) u = 0$. From (5.25) with $\hat{u} = 0$, recognize that

$$u = F(\rho) z. \tag{5.26}$$

The equality constraint is eliminated by substituting (5.26) into (5.24)

$$\begin{aligned}
&\text{minimize} && J(z) = \int_0^\infty \left(x^T Q x + z^T \bar{R}(\rho)\, z\right) dt && (5.27\text{a})\\
&\text{with respect to} && z &&\\
&\text{subject to} && &&\\
& && \dot{x} = A(\rho)\, x + \bar{B}(\rho)\, z, && (5.27\text{b})\\
& && \underline{z} \leq z \leq \bar{z}, && (5.27\text{c})
\end{aligned}$$

where,

$$\bar{R}(\rho) = F(\rho)^T R F(\rho),$$

$$\bar{B}(\rho) = B(\rho)\, F(\rho),$$

By eliminating the input equality constraints, the LQR problem is now a function of the intermediate control variable z. Note that (5.24) and(5.27) are equivalent problems. To eliminate the input inequality constraints (5.27c), saturate the optimal control action after the optimal control solution is synthesized. Therefore, the solution to the input constrained LQR problem (5.24) is

$$u = F(\rho)\, \mathrm{sat}\left(-K(\rho)\, x\right)_{\underline{z}}^{\bar{z}}, \tag{5.28a}$$

where,

$$K(\rho) = \bar{R}(\rho)^{-1} \bar{B}(\rho)^T P(\rho), \tag{5.28b}$$

$$0 = P(\rho)\, A(\rho) + A(\rho)^T P(\rho) + Q - P(\rho)\, \bar{B}(\rho)\, \bar{R}(\rho)^{-1} \bar{B}(\rho)^T P(\rho) \tag{5.28c}$$

While this technique was applied to a linear time-invariant system, it can also be extended to nonlinear systems using the process provided in Section 5.3.4.1.

5.4 Thermal Management System Design Case Study

Growing interesting in electrified aircraft is supported by various performance, ecological, and economic benefits [4]. However, thermal management remains a key challenge that inhibits widespread development and adoption of electrified vehicles. Inherent inefficiencies in the novel electrical systems are manifested as heat that, if not managed, could result in degraded performance, damaged components, or complete system failures. This heat generation is particularly important to consider for pulsed electrical components that generate significant thermal loads [47, 97, 98]. To address this reliability challenge, electrified power systems are equipped with a thermal management system to regulate system temperatures.

These novel electrified power systems require novel thermal management system designs. Traditional design methods for worst-case operation are no longer suitable for the increasing power and thermal requirements for electrified aircraft [123]. The proposed plant topology and closed-loop control co-design techniques can be used to explore the design space of novel thermal management systems.

The following sections will demonstrate how the novel techniques introduced in Section 5.3 are applied to a relatively simple thermal management system design problem. The case study investigates a mission-specific TMS design where the optimal thermal management system design changes as a function of varying power and energy requirements. The results will highlight a class of system designs well suited for a variety of applications, generate design insights, and evaluate the performance of the distance-sorted search optimization approach.

5.4.1 Thermal Management System Architecture Generation

As introduced in Section 5.3.1, the architecture generation problem requires a catalog of components and a set of structural constraints to define the topology design space. The component catalog for TMS design is illustrated in Figure 5.10 and comprises a heat sink, one heat exchanger, and four unique cold plates. The heat sink comprises a heat exchanger that can reject absorb heat to a chiller and a tank for bulk thermal energy storage. The single heat exchanger is used to develop multiple fluids loops. Each of the four cold plates will have unique heat loads and temperature constraints to represent different thermal requirements for different electrical equipment. This catalog was selected because it produces a thermal management system design suitable for a small UAV powertrain like the system introduced in Chapter 4.

To facilitate the architecture design problem, the structure of feasible topologies are represented as labeled rooted trees, as introduced in [16]. In graph theory, a rooted tree is an undirected graph where a simple path connects any pair of vertices. In the context of TMS design, the connections between components are represented by tubes in which fluid flows to transfer energy. However, note that rooted trees do not contain any cycles, meaning that they can not inherently represent fluid loops present in TMS design. To address this limitation, a post-processing modification is applied to the generated tree structures to connect a tree's leaves (a vertex with only one connection) to the root, thus representing a closed fluid loop. An example of a tree and its closed loop representation is illustrated in Figure 5.11. In the TMS design problem, the heat sink is treated as the tree's primary root, and the secondary side of the heat exchanger is treated as a secondary root. When creating the closed-loop rooted tree TMS, the leaves are connected to their nearest parent root. To facilitate using the perfect matching architecture generation algorithm, the heat exchanger is treated as two separate components: a primary and secondary side. This separation improves the scalability of the approach to a catalog with more components. The structural constraints imposed for the perfect matching enumeration process are as follows:

1. *Simple Graph*: The topologies can not contain any multi-edges (two connections between the same components) or loops (a component is connected to itself).

2. *Connected*: The topologies must be connected; there exists a path from each component to any other component.

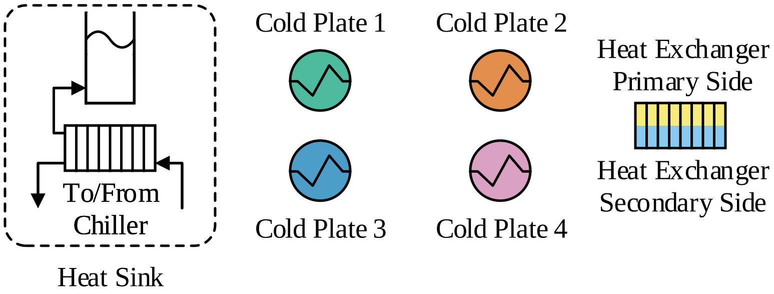


Figure 5.10: The TMS design problem component catalog consisting of a heat sink, a heat exchanger, and 4 cold plates.

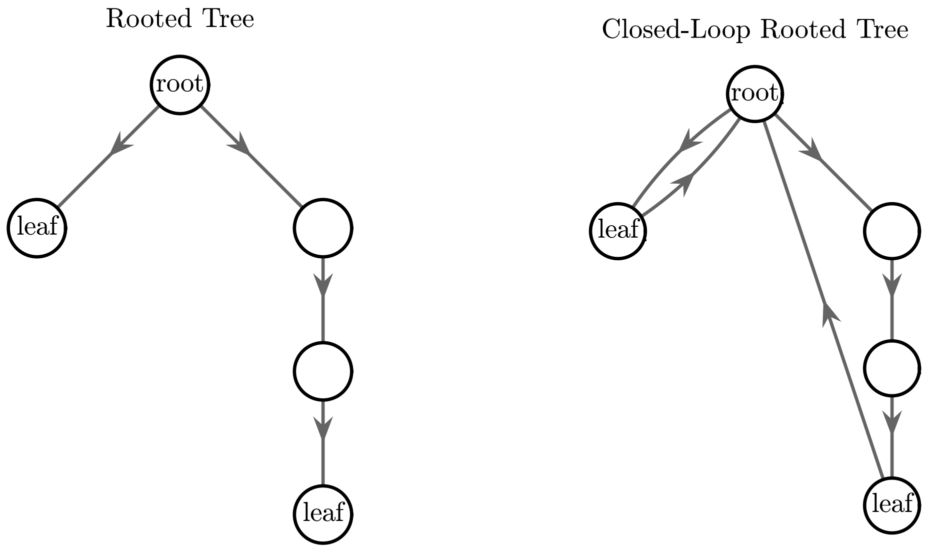


Figure 5.11: An example of a rooted tree (left) and its corresponding closed-loop representation (right).

3. *No Loops*: There cannot be a path that starts and ends at the same vertex.

4. *Tree Condition*: A candidate topology is a tree if $N_p = 2(N_c - 1)$ where N_p is the number of ports in the topology and N_c is the number of components.

5. *Heat Exchanger Side Constraint*: The primary and secondary sides of the heat exchanger must be connected. This is necessary because each side of the heat exchanger is treated as two separate components.

6. *Load Placement*: There must exist at least one path between any cold plate and the heat sink that does not pass through the secondary side of the heat exchanger. There must also exist at least one path between any cold plate and the secondary side of the heat exchanger that does not pass through the primary side of the heat exchanger. In other words, all candidate topologies will have at least two fluid loops and there must exist at least one cold plate in each loop (See Figure D.1 for examples).

By applying these constraints, 682 feasible system topologies are identified. As previously mentioned, each topology represented as a rooted tree is post-processed to convert the trees

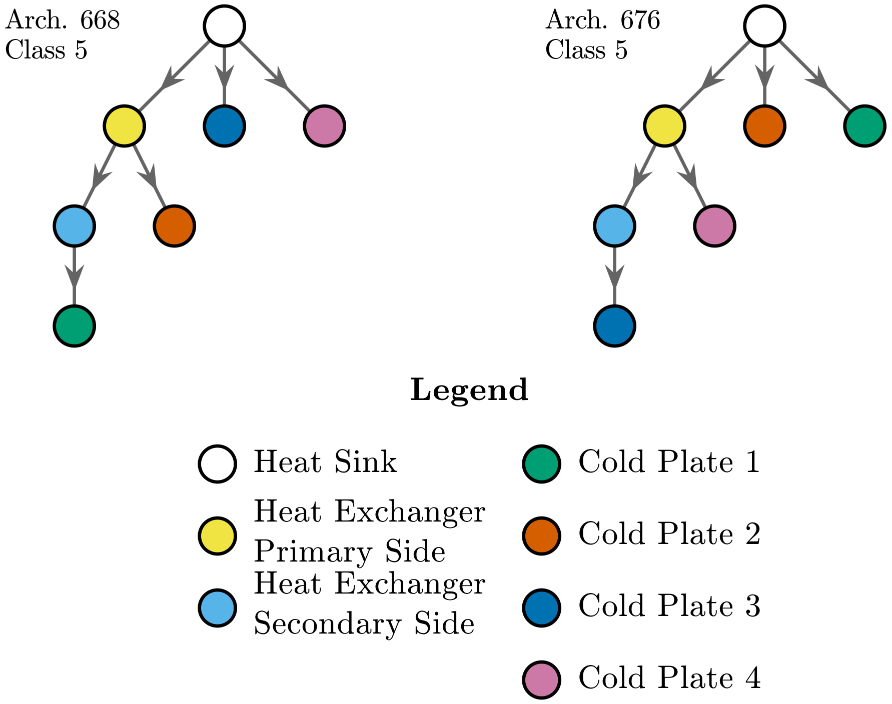
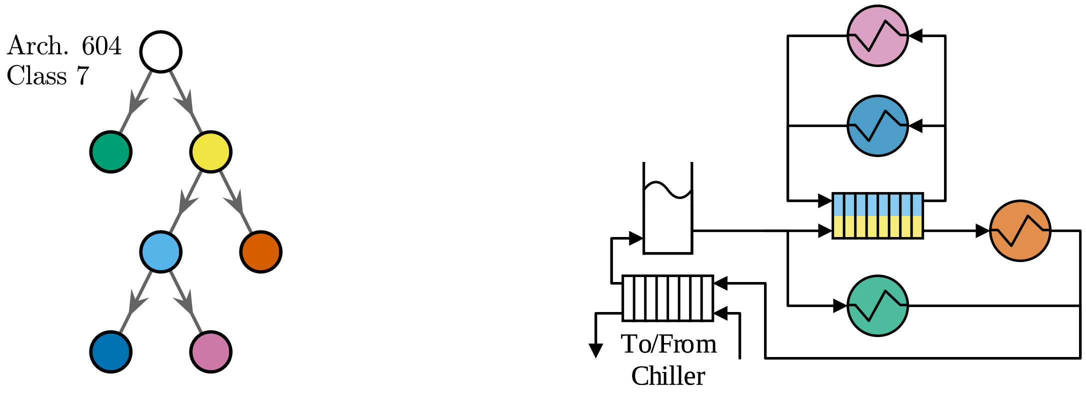


Figure 5.12: An example of two TMS architectures in the same class. They differ by cold plate connections.

into closed loop systems (Figure 5.11). Illustrating all 682 system would be complex, but they can be categorized into 37 unique topology classes. All architectures within a class have the same structure, but differ in the cold plate arrangement. All architecture classes are illustrated in Figure D.1. An example of two unique architectures in the same class is depicted in Figure 5.12. The relationship between the rooted tree representation and TMS schematic is shown in Figure 5.13.

The perfect matching algorithm outputs the adjacency matrix information for each topology that is used to define both the discrete and continuous topology design spaces $\mathcal{A}$ and $\mathcal{A}_c$, respectively, using the methods from Section 5.3.1. The superset system model is formed using the topology design space.



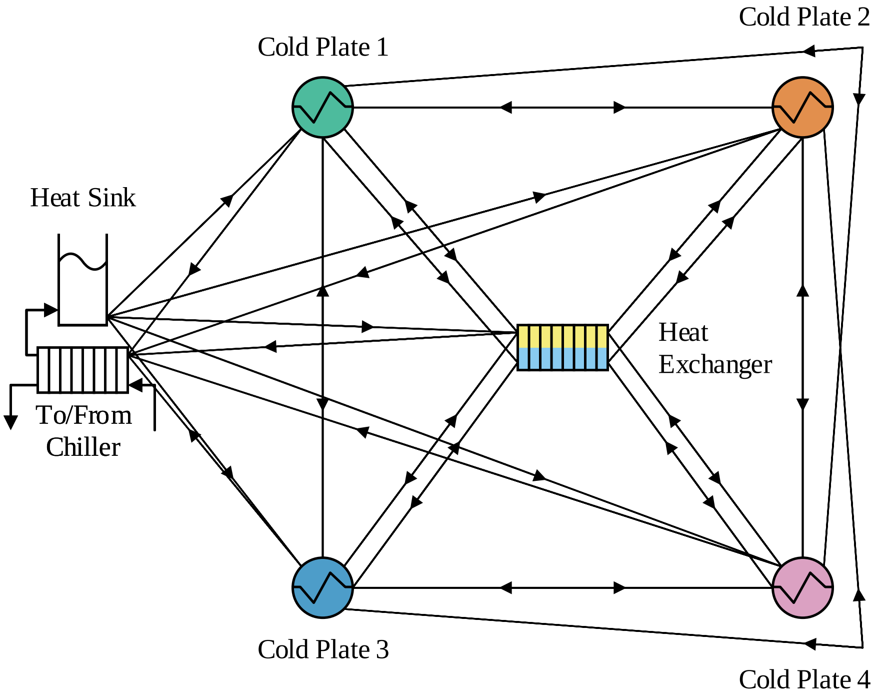
(a) Rooted tree representation

(b) TMS representation

Figure 5.13: An example relating a (a) rooted tree topology representation to its (b) TMS schematic representation.

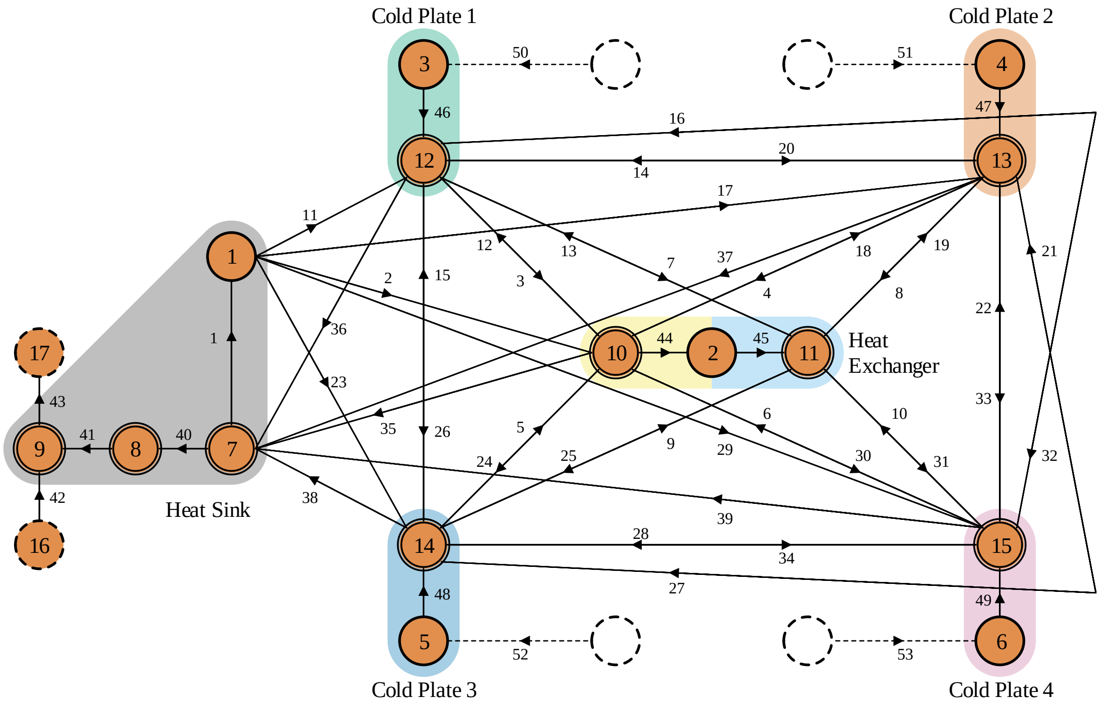
5.4.2 Graph-Based Thermal Management System Model

The TMS component catalog comprises a heat sink, a heat exchanger, and four cold plates, so the superset graph-based system model comprises heat sink, heat exchanger, and cold plate component graph-based models [53]. Using the set of all discretely feasible topologies, the superset system schematic and system graph-based model are generated (Figure 5.14). Because the superset system approach captures every possible system architecture in a single monolithic model, the schematic and model are complex with numerous connections between components and graph vertices. Note that only a subset of all the connections are present in a discretely feasible architecture, as seen in Figure 5.13b. The graph-based model's vertex, edge, input, and parameter information is provided in the Appendix (Tables B.9-B.12). The model dynamics are provided in the general form (5.10).



(a) Superset TMS schematic

Figure 5.14: The superset TMS (a) schematic and (b) graph-based model.



(b) Superset TMS graph-based model

Figure 5.14 (cont.): The superset TMS (a) schematic and (b) graph-based model.

5.4.3 Control Design

The thermal management system is controlled by varying the mass flow rates through the fluid loops via an input constrained AALQR as introduced in Section 5.3.4. There are 2 key considerations when synthesizing the controller. First, the thermal management system has nonlinear dynamics (bilinear to be precise), so the system needs to be linearized about an equilibrium condition before the controller can be synthesized. Second, the system must respect conservation of mass and flow limit constraints (i.e. input constraints). The remainder of this section will describe the methods to identify the system's equilibrium, linearize the dynamics, and synthesize an input-constrained AALQR.

Every system architecture has a unique steady-state operating condition, so the system's equilibrium point will be a function of the topology variables ρ. To identify the general form equilibrium point, choose a set of nominal input, disturbance, and parameter values $\left(\hat{u}, \hat{x}^e, \hat{P}^e, \hat{\lambda}_p\right)$ and substitute those values into (5.10). Because the thermal management system has bilinear dynamics, the resulting system will be in the form

$$\begin{bmatrix} C_d \dot{x} \\ 0 \end{bmatrix} = A_{eq}\left(\rho\right)\tilde{x} + B_{eq}\left(\rho\right), \tag{5.29}$$

$$\text{where,}$$

$$\tilde{x} = \begin{bmatrix} x \\ a \end{bmatrix}.$$

Let $x_{eq} = \begin{bmatrix} \hat{x} \\ \hat{a} \end{bmatrix}$ be the equilibrium solution to (5.29) when $\dot{x} = 0$. The equilibrium is computed via a matrix inverse

$$x_{eq}\left(\rho\right) = -A_{eq}^{-1}\left(\rho\right) B_{eq}\left(\rho\right). \tag{5.30}$$

Therefore, the equilibrium point for any topology is $\left(\hat{x}, \hat{a}, \hat{u}, \hat{x}^e, \hat{P}^e, \hat{\lambda}_p\right)$.

The linearization methods introduced in Section 2.4 are applied to linearize the full nonlinear system dynamics (5.10) about the identified equilibrium condition, with the addition that the linear system dynamics are a function of the topology design variables ρ. The linear system dynamics are

$$\Delta\dot{x} = A\left(\rho\right)\Delta x + B\left(\rho\right)\Delta u, \tag{5.31}$$

$$\text{where,}$$

$$\Delta x = x - \hat{x},$$

$$\Delta u = u - \hat{u}.$$

This model assumes operation at the nominal disturbance and parameter values (i.e. $x^e - \hat{x}^e = 0$, $P^e - \hat{P}^e = 0$, and $\lambda - \hat{\lambda} = 0$).

An input constrained AALQR is synthesized for the nominal system. The controller synthesis is subjected to a flow conservation equality constraint $M\left(\rho\right)u = 0$ that guarantees, at any instant, the same amount of fluid both enters and leaves the component. This is suitable for the type of incompressible flow utilized in the class single-phase thermal management systems depicted in Figure 5.14. The flow conservation constraint for every flow junction is

$$M_j(\rho)\,u = 0 = \sum_{u_i \in U_j^{in}} \rho_i u_i - \sum_{u_i \in U_j^{out}} \rho_i u_i, \tag{5.32}$$

where U_j^{in} is the set of all mass flow rates directed into the junction j and U_j^{out} is the set of all mass flow rates directed out of the junction j. As presented in Section 5.3.4.3, the equality constraint is eliminated by finding the nullspace of the flow conservation constraint $F(\rho) = \text{null}(M(\rho))$. The nullspace can be computed using *null()* in MATLAB or by putting $M(\rho)$ in reduced row echelon form. For the flow conservation constraint, the nullspace has a physical meaning.

Every flow conservation constraint (5.32) may be dependent on up to N_m unique flow rates. Of the N_m flow rates, $N_m - 1$ flow rates may be freely selected while the last flow rate is entirely dependent on the other $N_m - 1$ flow rates. An example is provided for the case where $N_m = 3$. In (5.33a), the flow conservation constraint is written in the form (5.32), whereas (5.33b) illustrates the solved flow conservation constraint that is represented by the nullspace in which u_3 is dependent on the choice of u_1 and u_2. In this sense, some flow rates are independent while other flow rates are dependent. The independent flow rates are directly selected by the synthesized controller, and then the dependent flow rates are computed from the control solution. Note that there may be various representation for the nullspace and any will yield a valid solution.

$$0 = \rho_1 u_1 + \rho_2 u_2 - \rho_3 u_3 \tag{5.33a}$$

$$u_3 = \frac{1}{\rho_3}\left(\rho_1 u_1 + \rho_2 u_2\right) \tag{5.33b}$$

In addition to the equality constraints, the controller is subjected to lower and upper flow bounds imposed as inequality constraints. If the lower and upper flow limits for the nonlinear system are $\underline{u}$ and $\bar{u}$, then the lower and upper flow limits for the linearized system are $\Delta\underline{u} = \underline{u} - \hat{u}$ and $\Delta\bar{u} = \bar{u} - \hat{u}$, respectively. Consequently, for the system with eliminated flow constraints, $\underline{z}$ and $\bar{z}$ are the elements of $\Delta\underline{u}$ and $\Delta\bar{u}$, respectively, that correspond to the

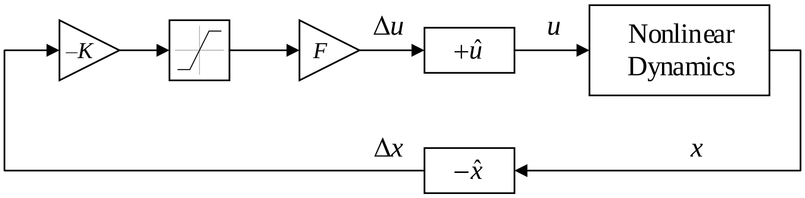


Figure 5.15: The block diagram for the input constrained AALQR for TMS design.

independent flow rates. The lower and upper bound flow constraints are imposed through the approximate saturation function (4.4).

The complete controller synthesis equations that are encoded into the co-design problem are summarized below, and the feedback control system is illustrated in Figure 5.15. The controller synthesis requires a candidate ρ, LQR weight matrices Q and R, and nominal input, disturbance and parameter information $\hat{u}$, $\hat{x}^e$, $\hat{P}^e$, and $\hat{\lambda}$.

$$\begin{bmatrix} \hat{x} \\ \hat{a} \end{bmatrix} = -A_{eq}^{-1}(\rho)\, B_{eq}(\rho)\,, \tag{5.34a}$$

$$F(\rho) = \text{null}\,(M(\rho))\,, \tag{5.34b}$$

$$0 = P(\rho)\, A(\rho) + A(\rho)^T P(\rho) + Q - P(\rho)\, \bar{B}(\rho)\, \bar{R}(\rho)^{-1} \bar{B}(\rho)^T P(\rho) \tag{5.34c}$$

$$K = \bar{R}(\rho)^{-1} \bar{B}(\rho)^T P(\rho)\,, \tag{5.34d}$$

$$u = F(\rho)\, \text{asat}\,(-K(\rho)\,(x - \hat{x}))_{\underline{z}}^{\bar{z}} + \hat{u}, \tag{5.34e}$$

where,

$$\bar{R}(\rho) = F(\rho)^T RF(\rho)\,,$$

$$\bar{B}(\rho) = B(\rho)\, F(\rho)$$

Recall that $A(\rho)$ and $B(\rho)$ are computed by linearizing the system dynamics about the equilibrium condition.

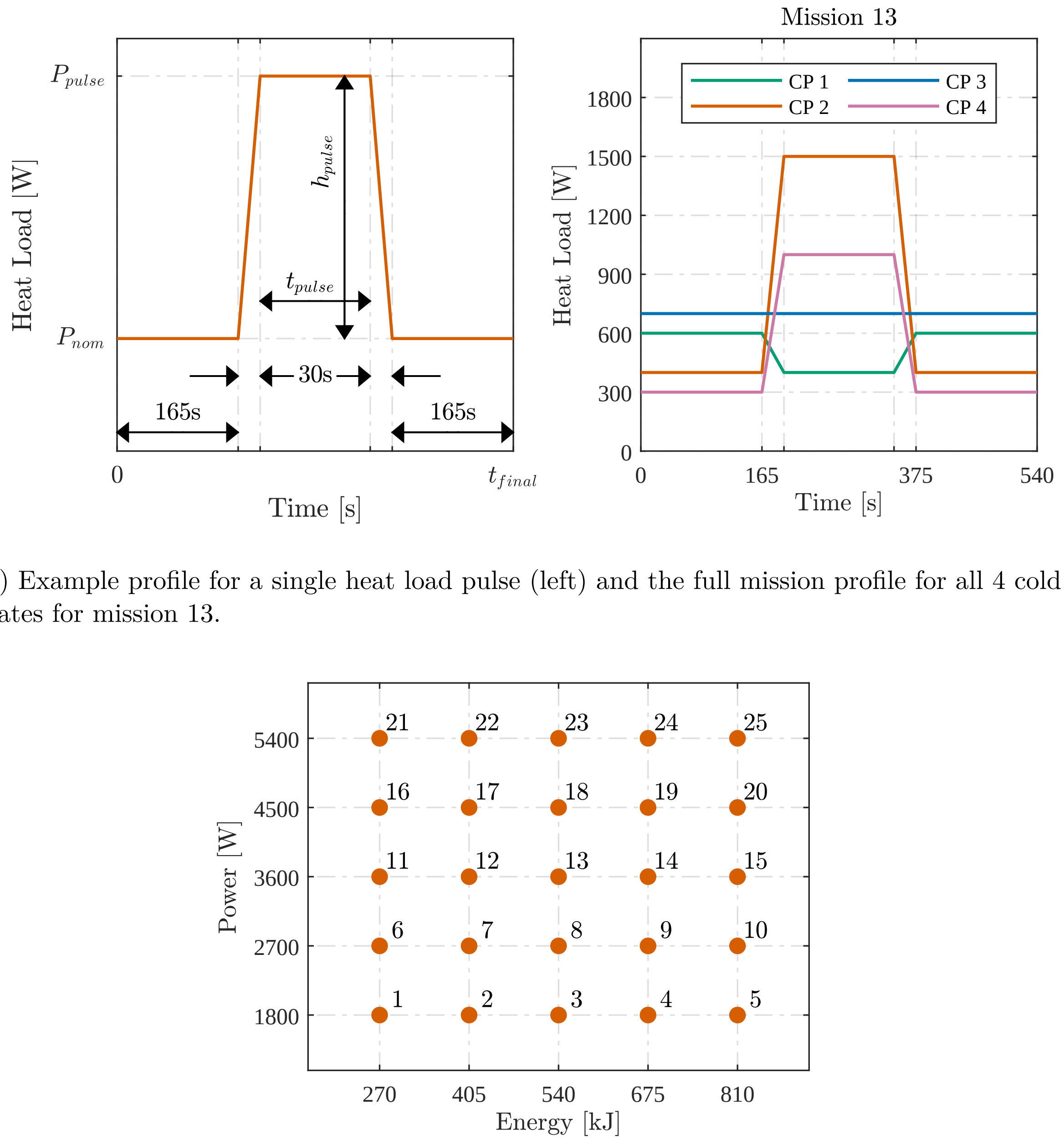
5.4.4 Co-Design Optimization Problem

The proposed methods are used to understand the impact different mission requirements have on the design and control of a thermal management system. In this study, each mission comprises four unique pulsed heat loads that are applied to each of the four TMS cold plates (Figure 5.16a). Each heat load has a nominal power level P_{nom} engaged for 165s, a pulsed power level P_{pulse} engaged for a variable amount of time t_{pulse}, and 30s transient ramp segments in between. To explore how the system architecture and control behavior changes as a function of mission, an optimal vehicle will be designed for 25 different power and energy requirements (Figure 5.16b). The total power P and total energy E for each mission is computed as

$$P = \sum_{i=1}^{4} P_{pulse,i}, \tag{5.35a}$$

$$E = t_{pulse} \sum_{i=1}^{4} P_{pulse,i}, \tag{5.35b}$$

where $P_{pulse,i}$ is the pulse power of heat load i. To vary the mission power and energy requirements, the magnitude and length of the pulse loads are changed. Table 5.1 provides the pulse power and length of each of the 25 missions illustrated in Figure 5.16b. Mission 13 is considered the nominal mission. Note that each mission has the same P_{nom}. Those values are $P_{nom,1} = 600W$, $P_{nom,2} = 400W$, $P_{nom,3} = 700W$, and $P_{nom,4} = 300W$.



(a) Example profile for a single heat load pulse (left) and the full mission profile for all 4 cold plates for mission 13.

(b) Grid of 25 thermal energy and power requirements

Figure 5.16: An illustration of (a) The cold plates' pulse heat loads. (b) The 25 different thermal power and energy mission requirements for the TMS design study.

Table 5.1: Pulse heat load power and energy information for each cold plate and mission as described in Figure 5.16b.

Mission	t_{pulse} [s]	CP 1 [W]	CP 2 [W]	CP 3 [W]	CP 4 [W]	Total Power [W]	Total Energy [kJ]
1	150	200	750	350	500	1800	270
2	225	200	750	350	500	1800	405
3	300	200	750	350	500	1800	540
4	375	200	750	350	500	1800	675
5	450	200	750	350	500	1800	810
6	100	300	1125	525	750	2700	270
7	150	300	1125	525	750	2700	405
8	200	300	1125	525	750	2700	540
9	250	300	1125	525	750	2700	675
10	300	300	1125	525	750	2700	810
11	75	400	1500	700	1000	3600	270
12	112.5	400	1500	700	1000	3600	405
13	150	400	1500	700	1000	3600	540
14	187.5	400	1500	700	1000	3600	675
15	225	400	1500	700	1000	3600	810
16	60	500	1875	875	1250	4500	270
17	90	500	1875	875	1250	4500	405
18	120	500	1875	875	1250	4500	540
19	150	500	1875	875	1250	4500	675
20	180	500	1875	875	1250	4500	810
21	50	600	2250	1050	1500	5400	270
22	75	600	2250	1050	1500	5400	405
23	100	600	2250	1050	1500	5400	540
24	125	600	2250	1050	1500	5400	675
25	150	600	2250	1050	1500	5400	810

For each mission, the optimal system design is characterized by the choice of topology ρ and LQR matrix weights Q and R. The optimal set of plant and control parameters are identified through the solution of the following plant architecture and closed-loop control co-design optimization problems. Problem (5.36) is the discrete variation, while (5.37) is the continuous variation.

$$\begin{aligned}
&\text{minimize} && \int_0^{t_f} \|u\|_\rho^2 \, dt && (5.36\text{a})\\
&\text{with respect to} && \lambda, Q, R &&\\
&\text{subject to} && &&\\
& && \left.\begin{aligned} &\rho = \mathrm{P}\lambda,\\ &\sum \lambda = 1,\\ &\lambda \in \{0,1\}, \end{aligned}\right\} \text{Discretely Feasible Topology} && (5.36\text{b})\\
& && \text{Equation (5.34)}\Big\} \text{Input Constrained AALQR Synthesis} && (5.36\text{c})\\
& && \text{Equation (5.10)}\Big\} \text{Superset Graph Model} && (5.36\text{d})\\
& && \left.\begin{aligned} &10^{-2} \leq Q \leq 10^{2}\\ &10^{2} \leq R \leq 10^{6} \end{aligned}\right\} \text{LQR Weight Limits} && (5.36\text{e})\\
& && \text{Upper and lower bound constraints from Tables B.9 and B.11.} && (5.36\text{f})
\end{aligned}$$

$$\begin{aligned}
&\text{minimize} && \int_0^{t_f} \|u\|_\rho^2 \, dt && (5.37\text{a})\\
&\text{with respect to} && \lambda, Q, R &&\\
&\text{subject to} && &&\\
& && \left.\begin{aligned} &\rho = \mathrm{P}\lambda,\\ &\sum \lambda = 1,\\ &\lambda \geq 0, \end{aligned}\right\} \text{Continuous Feasible Topology} && (5.37\text{b})
\end{aligned}$$

$$\text{Equation (5.34)}\Big\}\ \text{Input Constrained AALQR Synthesis} \tag{5.37c}$$

$$\text{Equation (5.10)}\Big\}\ \text{Superset Graph Model} \tag{5.37d}$$

$$\left.\begin{array}{l} 10^{-2} \leq Q \leq 10^{2} \\ 10^{2} \leq R \leq 10^{6} \end{array}\right\}\ \text{LQR Weight Limits} \tag{5.37e}$$

$$\text{Upper and lower bound constraints from Tables B.9 and B.11.} \tag{5.37f}$$

where $\|u\|_{\rho}^{2} = u^{T}\text{diag}\,(\rho)\,u$ and $\mathrm{P} = \begin{bmatrix} \mathrm{P}_1 & \mathrm{P}_2 & \cdots & \mathrm{P}_{682} \end{bmatrix}$ as described in Section 5.3.1.

As described in Section 5.3.3, the TMS design problem will be solved using two methods. First, the exhaustive search method is applied (Section 5.3.3.1) by optimizing (5.36) each of the 682 feasible TMS topologies. Second, the distance sorted search method is used (Section 5.3.3.2) by first solving (5.37) and then searching different architectures based on the optimal continuous results. By solving the problem 2 ways, the relative performance between the two optimization strategies can be compared. Note that the distance-sorted search will not repeat the optimizations completed for the exhaustive search. In total, 17,075 optimizations will be solved for this design study: 683 solutions (682 discrete and 1 continuous) for each of the 25 power and energy requirements.

These optimization problems are built in the python-based optimization packages OpenMDAO and Dymos [115, 116] and solved using the sequential least squares programming (SLSQP) gradient-based optimizer from SciPyOptimize [117]. It is solved on a workstation with an AMD Threadripper 3990X 2.90 GHz 64 Core CPU and 256 GB of RAM. Each design case is warm-started using the nominal Q and R gain values (Table B.12).

The following mission-specific design analysis will investigate the optimal design for each mission scenario, determine if a class of architectures is best suited for a class of mission, investigate the optimized closed-loop system trajectories, and evaluate the effectiveness of the distance-sorted search.

5.5 Results

The combined architecture and control optimization results are analyzed to answer the following questions:

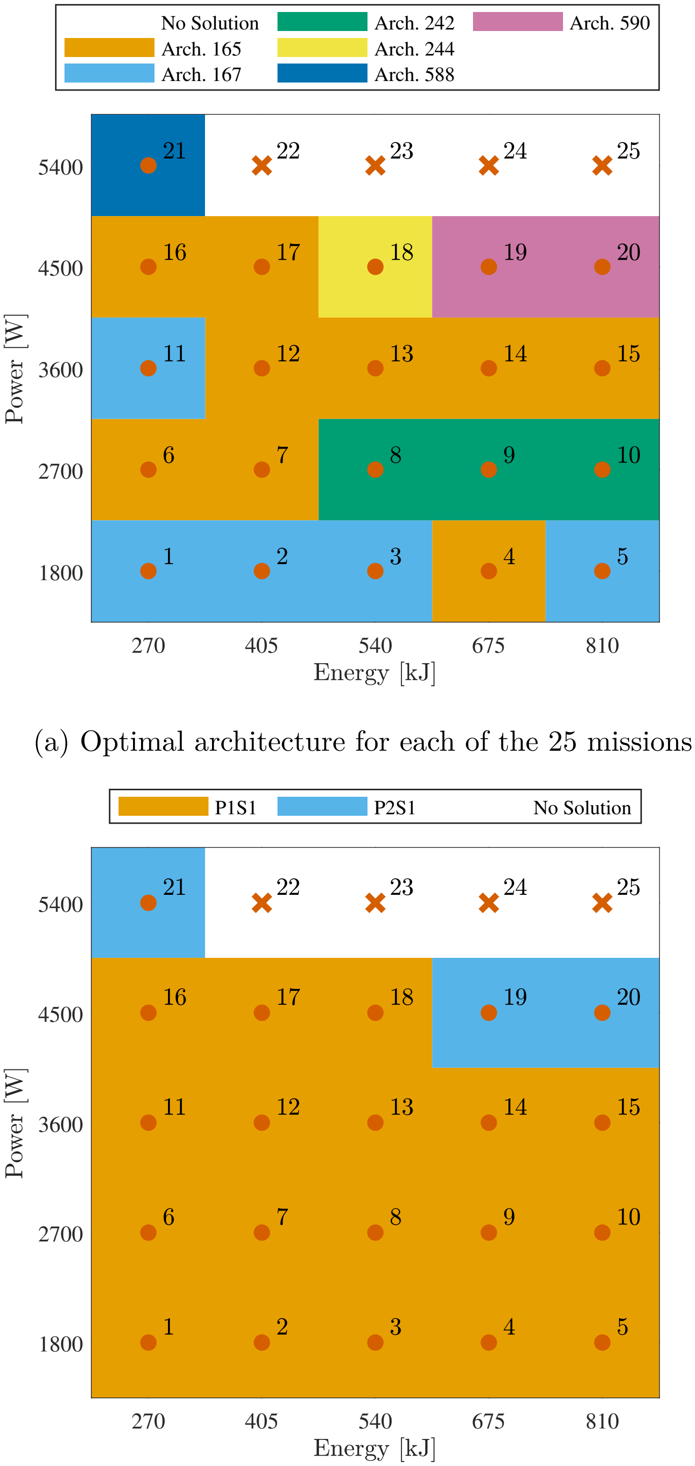
> *Which class of system architectures is best suited for a range of missions?*
>
> *How is a given topology and its controller optimized for any given mission?*
>
> *Does relaxing the traditional discrete topology selection problem improve the optimization process? Can it find a system that is suitable faster?*

These questions are answered through a mission-level, system-level, and optimization routine analyses.

5.5.1 Mission-Level Analysis

The optimal TMS system topology and controller is identified through both the exhaustive and distance-sorted search methods for each of the missions described by Figure 5.16b. Figure 5.17a illustrates the optimal architecture design for each of the 25 missions. Notably, missions 22 through 25, which had the highest power requirements, had no feasible solutions, thus the missions could not be completed while respecting constraints using this class of thermal management systems. Otherwise, architectures 165, 167, 242, 244, 588, and 590 were the optimal solutions for the other 21 feasible missions (Figure 5.18). There is sparsity in optimal design architectures in Figure 5.17a where architecture 165 is optimal for mission 4 but not for any of the similar missions 3, 5, 8, 9, or 10. Note that these are non-convex optimizations that may have local minima with many similarly performing designs.

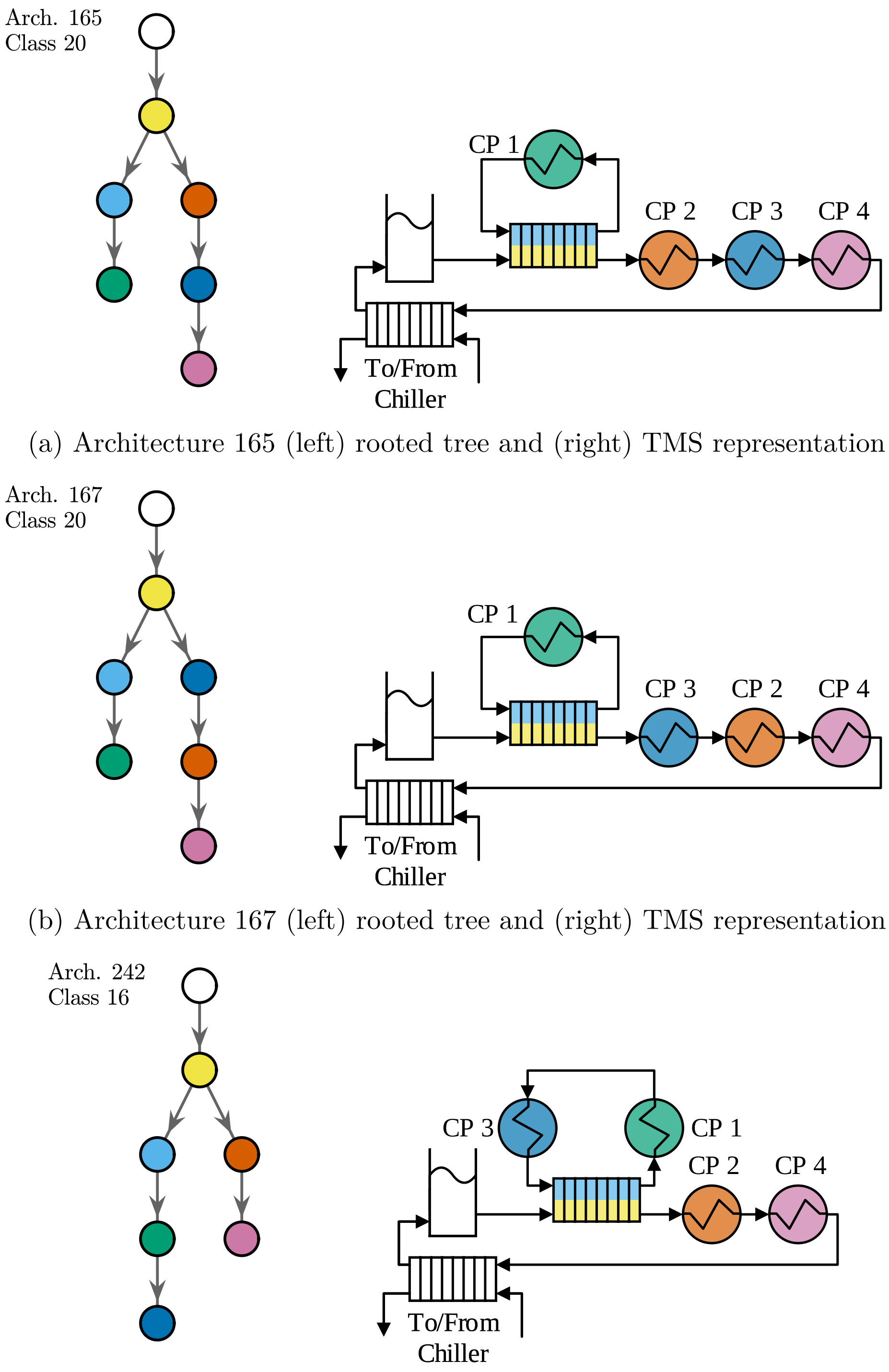
Therefore, Figure 5.17b re-illustrates the result from Figure 5.17a where the optimal topologies (Figure 5.18) are sorted by general structure. Figure 5.17b shows that all optimal topologies from missions 1-18 (Archs. 165, 167, 242, and 244) have two single flow paths; one that flows through the tank and another that flows through the secondary side of the heat exchanger. These designs will be referred to as *P1S1* for 1 *primary* flow path



(a) Optimal architecture for each of the 25 missions

(b) Optimal architecture for each of the 25 missions, categorized by the number of flow paths

Figure 5.17: (a) The optimal TMS architectures categorized by (b) number of flow paths.

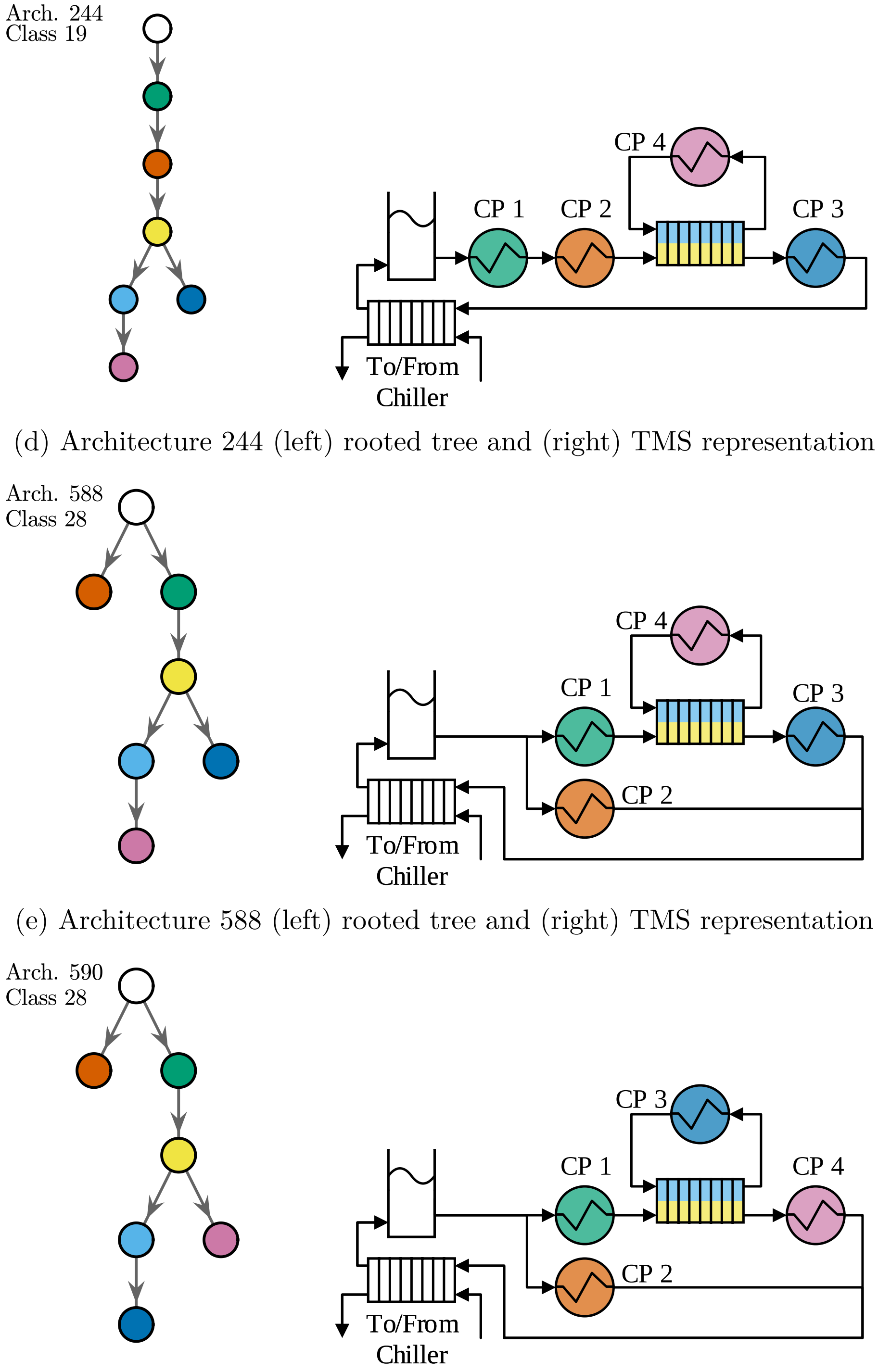


(a) Architecture 165 (left) rooted tree and (right) TMS representation

(b) Architecture 167 (left) rooted tree and (right) TMS representation

(c) Architecture 242 (left) rooted tree and (right) TMS representation

Figure 5.18: The rooted tree and TMS schematic representation for the optimal TMS designs from Figure 5.17a for Architectures (a) 165, (b) 167, (c) 242, (d) 244, (e) 588, and (f) 590.



(d) Architecture 244 (left) rooted tree and (right) TMS representation

(e) Architecture 588 (left) rooted tree and (right) TMS representation

(f) Architecture 590 (left) rooted tree and (right) TMS representation

Figure 5.18 (cont.): The rooted tree and TMS schematic representation for the optimal TMS designs from Figure 5.17a for architectures (a) 165, (b) 167, (c) 242, (d) 244, (e) 588, and (f) 590.

Table 5.2: The number of systems with 1 or 2 primary flow paths for the best 7 designs for each mission. P1S1-1 primary and 1 secondary flow path. P2S1-2 primary and 1 secondary flow path.

Missions	P1S1	P2S1
1-17	7	0
18	2	5
19-20	1	6
21	0	7

through the tank and 1 *secondary* flow path through the secondary side of the heat exchanger. The P1S1 class of system architectures contrasts the *P2S1* designs illustrated by architectures 588 and 590 where 2 flow paths are present in the primary fluid loop containing the tank. Figure 5.17b illustrates that as the thermal power and energy mission requirements increase, the number of primary flow paths increases so that fresh coolant can be supplied to the hottest cold plates.

Because these optimization problems are non-convex, the structure of the best 1% of designs are investigated for each mission (~ 7 architectures per mission). This set of designs is defined as *near optimal* and includes the single optimal design and the 6 next best designs. In total, there is a set of 21 unique, near optimal system architectures spanning the 21 feasible missions (Figure D.2). For each mission, Table 5.2 categorizes the best 1% of designs by flow structure (either P1S1 or P2S1). For the lower power and energy missions 1-17, all near optimal designs are single primary flow path configurations (P1S1). However, as the energy and power requirements increase, more dual primary flow path configurations (P2S1) are included in the set of near optimal designs because including more flow paths allows fresh coolant to be supplied to cold plates, thus allowing the system to continue operating within its thermal constraints.

The multi-flow path systems will require more pumping energy and therefore perform worse than their single flow path counterparts. However, the performance degradation is at the benefit of increased reliability because the multi-flow path systems are better-suited to respect constraints under high thermal load. Figure 5.19 shows the variation in pumping

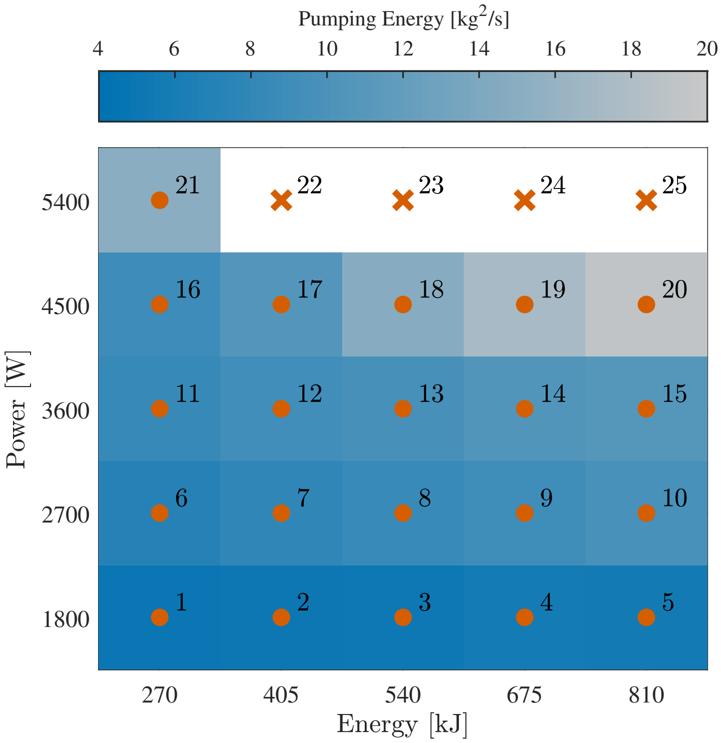


Figure 5.19: The pumping energy (optimization objective) requirement for the optimal system design for each mission.

energy requirements (5.37a) across the 25 missions for the optimal designs in Figure 5.17a. The low power and energy missions use less pumping energy than the more challenging missions because the low degree of thermal loading requires less fluid movement to reject heat from the system. Note that as the thermal energy requirements increase, so does the pumping energy requirements because the pumps have to operate for a longer time duration. The P2S1 designs (missions 19-21) use more pumping energy than the P1S1 designs (missions 1-18).

Notably, there are zero system designs that satisfy the requirements and constraints for missions 22-25. As Figure 5.20 illustrates, there also exist a range of infeasible designs for all 25 missions. Almost the entire design space is feasible for the missions with the lowest power and energy requirements (bottom left corner of Figure 5.20). However, as energy and power levels increase and the missions become more challenging, the number of system configurations that satisfy both the mission requirements and constraints decreases. In

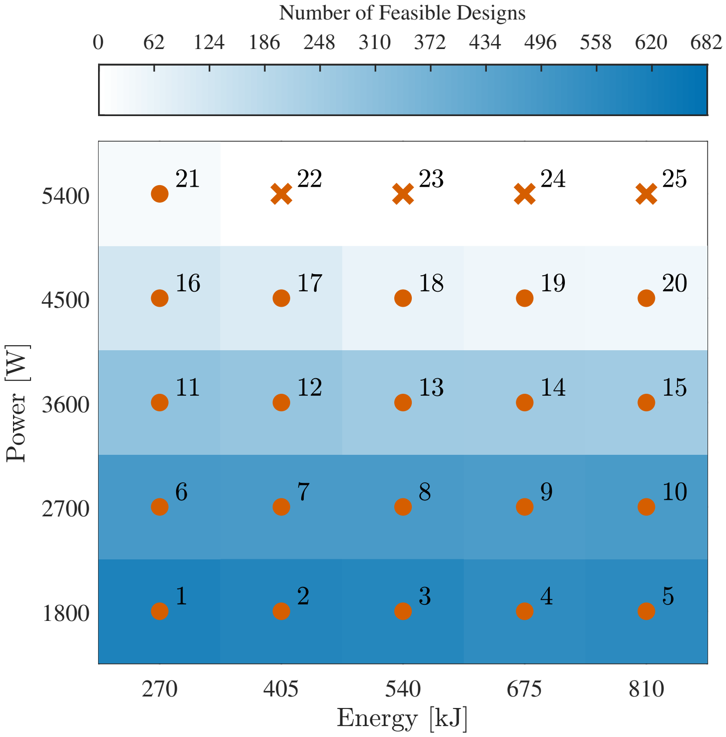


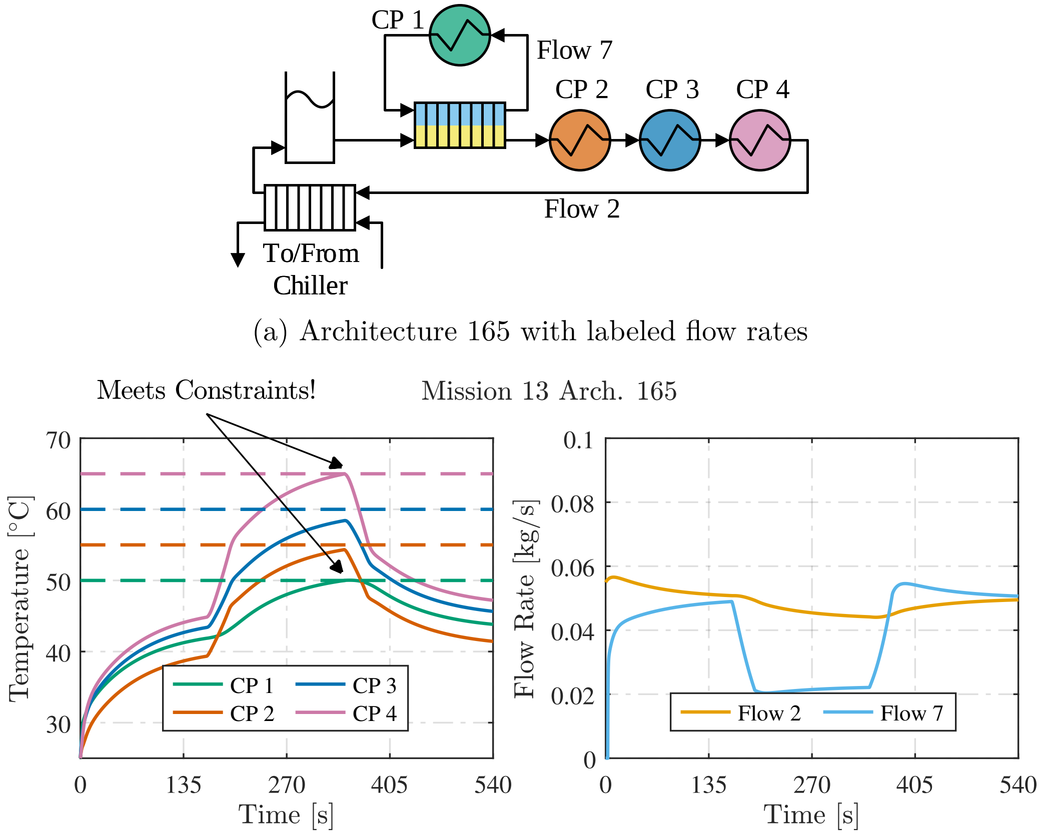
Figure 5.20: The number of dynamically feasible TMS architectures for each mission. There are at most 682 feasible topologies.

particular, the gradient shading in Figure 5.20 shows that the thermal power requirements have a greater impact on the feasible design space than the energy requirements.

While the mission-level analysis has shown which designs perform best for specific missions, it has also generated valuable insights for a mission-specific vehicle design. For comparatively low thermal power and energy mission requirements, P1S1 class topologies are relatively simple and perform well. For more challenging missions, multi-flow path P2S1 class architectures are required in order to respect thermal operating limits. For this class of systems and missions investigated in this work, the transition between P1S1 and P2S1 topologies occurs nears 4500W and over 540kJ.

5.5.2 System-Level Analysis

The control behavior is optimized to meet thermal operating limits for a given topology. As the previous section demonstrated, there are two general architecture classes: P1S1 and



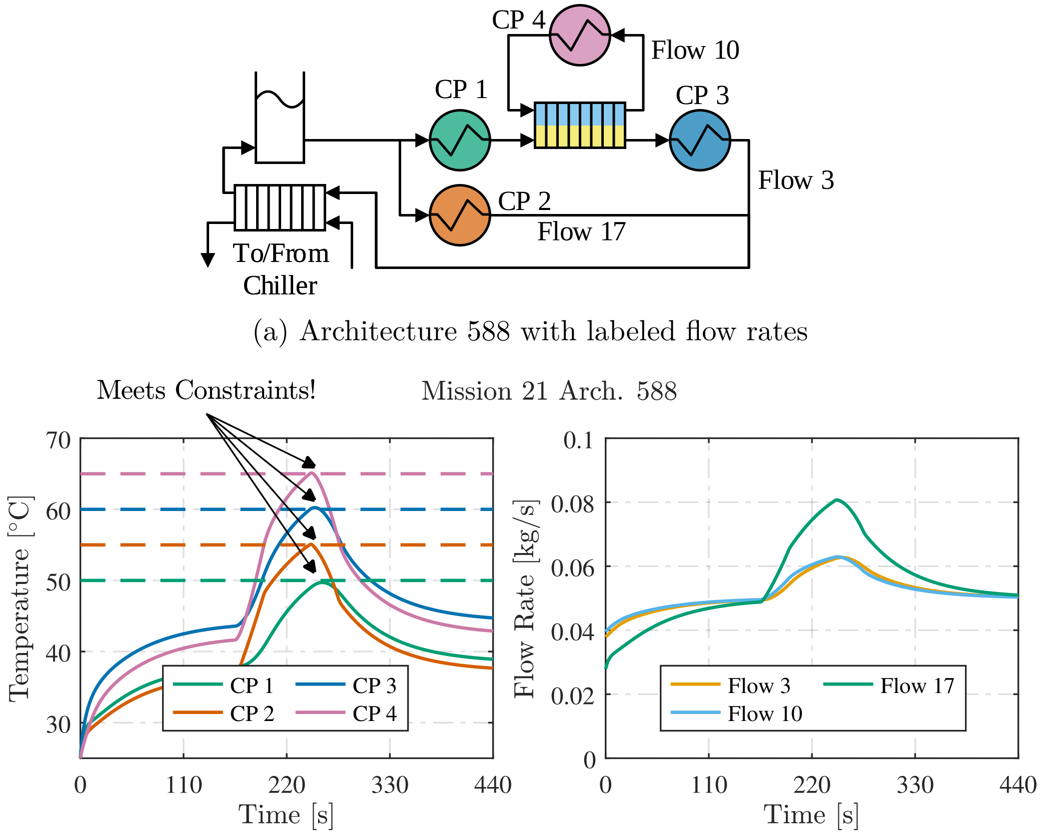
(a) Architecture 165 with labeled flow rates

(b) Optimal (left) temperature state and (right) flow rate input trajectories for mission 13 architecture 165.

Figure 5.21: The (a) TMS schematic representation and (b) optimal state and input trajectories for mission 13 architecture 165.

P2S1. Figure 5.21b illustrates the cold plate temperature (CP) state and mass flow rate input trajectories for mission 13 where the optimal design was P1S1 architecture 165. This architecture has two controllable flow rates as labeled in Figure 5.21a and 4 cold plate temperatures to maintain within thermal operating limits. In general, to be most energy efficient, it is desirable to operate at the system's thermal limit. However, the two flow rates only provide two degrees of freedom, so only two cold plate temperatures (CP 1 and 4) meet their thermal operating limit.

Conversely, a P2S1 architecture has additional flow paths and consequently more control authority. Figure 5.22 shows the cold plate temperature state and mass flow rate input trajectories for the optimal architecture 588 for mission 21. In this system, every cold plate temperature meets its thermal operating limit. Not only does this system benefit from the



(a) Architecture 588 with labeled flow rates

(b) Optimal (left) temperature state and (right) flow rate input trajectories for mission 21 architecture 588.

Figure 5.22: The (a) TMS schematic representation and (b) optimal state and input trajectories for mission 21 architecture 588.

additional degree of control freedom from the additional flow rate, but it can also configure the system so that two serially connected cold plates exactly meet their thermal limit using the same fluid. Notably, cold plate 2 is put on its own flow path because that component has the greatest thermal load and second-strictest thermal constraint, so it requires the largest flow rates for adequate heat rejection. The state and input trajectories for missions 13 and 21 were selected here as representative P1S1 and P2S1 architectures, respectively, for the entire mission space. All state and input trajectories for other missions are provided in the Appendix (Figure D.3).

To show that multiple flow rates are required for the more challenging missions, Figure 5.23 is included to compare the state trajectories for architectures 165 and 588 for the mission 21. Architecture 588 was the optimal design for mission 21 while architecture 165 was one

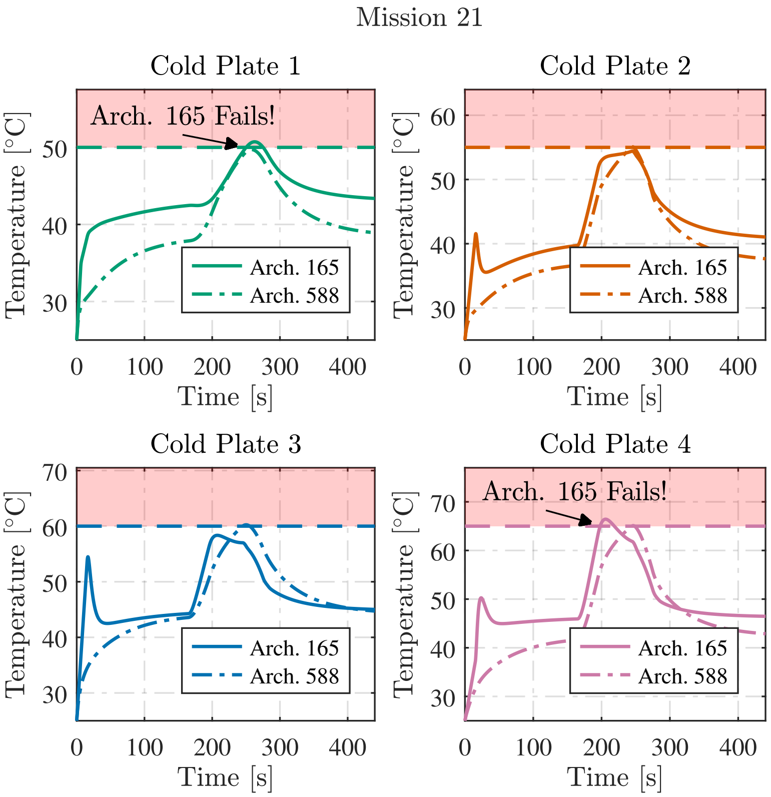


Figure 5.23: A comparison of the state trajectories for architectures 165 and 588 for mission 21. Architecture 165 cannot complete the mission because it cannot satisfy thermal constraints

of the highest performing designs for many of the other missions. Both architectures were previously illustrated in Figures 5.21a and 5.22a, respectively. From Figure 5.23, architecture 165 cannot maintain cold plates 1 or 4 within their thermal operating limits during the pulse heat load segment because the fluid is too warm by the time it reaches the loads.

The system-level analysis shows that both a suitable topology and controller must be generated for a jointly optimal solution. The results indicated that a controller is intuitively designed to meet the system's thermal operating limits. Also, the system architecture can be modified to provide additional control degrees of freedom that may be required to meet the design specification and objective.

5.5.3 Optimization Analysis

The topology optimization problems were solved using two methods: exhaustive search, which is commonplace in the literature, and distance-sorted search (Section 5.3.3). The distance-sorted search method leverages the solution to the novel relaxed topology optimization formulation introduced in this chapter. However, there is a trade-off between the two approaches. An exhaustive search evaluates the entire feasible design space to identify the globally optimal solution, which is ideal but does not scale well to large discrete design problems like system-level vehicle design. It may be preferred to search less of the design space to find a design that is near optimal. The following analysis shows how the distance-sorted search optimization approach can be used to identify adequate designs with fewer optimizations and understand the tradeoff in design performance.

Figure 5.24 describes the number of architecture evaluations in the distance-sorted search that are required for different design objectives. *Lost performance* is the performance of a given design relative to the optimal design. For example, the optimal design has 0% lost performance, while a design that performs 5% worse has 5% lost performance. While identifying the global optimal design may be ideal, it is likely that similar performing designs can be identified with fewer architecture evaluations. Figure 5.24 (left) shows that approximately

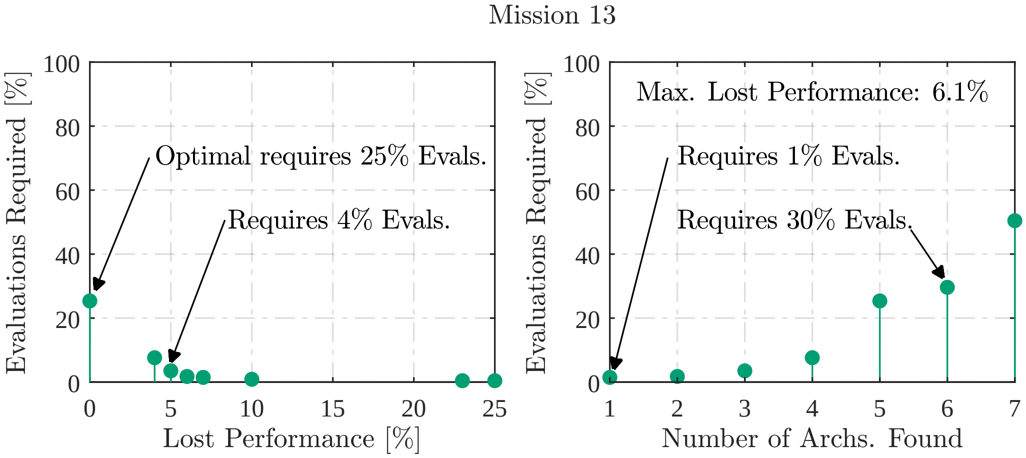


Figure 5.24: The number of architecture evaluations required versus performance trade-offs when using the distance-sorted search optimization method. The results are shown for mission 13.

25% of the discrete topology design space would need to be evaluated to identify the global optimal design. However, if willing to sacrifice 5% performance, a similar design could be identified by searching 4% of the entire design space.

Figure 5.24 (right) provides a similar metric for the distance-sorted search by investigating the best 7 architectures (i.e. best ∼1%), which have at most 6.1% lost performance for mission 13. This set of architectures near optimal because it includes the single optimal system and the 6 next best designs. The vertical axis shows the number of architecture evaluations required to identify any number of those designs. To find 1 of the near optimal designs, 1% of the design space would be searched. Conversely, 30% of the design space could be searched to identify 6 of the best 7 designs. Recall that all the designs will have at worst 6.1% lost performance in comparison to the global optimal solution. This result highlights that near optimal designs can be identified quickly if some performance can be sacrificed.

If using an exhaustive search approach, designs are randomly optimized until the entire design space is evaluated. Conversely, Figure 5.24 demonstrates that using the solution to the relaxed architecture optimization to guide a distance-sorted search process may help identify the optimal or near optimal design in far fewer system optimizations. For mission 13, the global optimal design could be identified 4 times faster (i.e. 25% evaluations required) than an exhaustive approach. However, it is not possible to determine if any given design is globally optimal without an exhaustive search, so stopping criterion are applied to distance-sorted searches to limit the number of total optimizations. As illustrated in Figure 5.24 (left), one of the near optimal designs was identified 100 times faster (i.e. 1% required evaluations) than an exhaustive search. Although Figure 5.24 only provides optimization results for mission 13, this trend holds across all missions. In almost all missions, a near optimal design would be identified after applying a distance sorted search to only 10% of the design space. This equates to identifying a high performance design an order of magnitude faster than through an exhaustive approach. Notably, the distance sorted search finds a near optimal design in the first system optimization for missions 16 and 18. The search results for all missions are provided in the Appendix (Figure D.4).

5.6 Conclusion

Electrified aircraft may have unique energy system architectures tailored to the need of their end application. But the combinatorial size of the discrete architecture design space makes it challenging for engineers to systematically evaluate the performance trade-offs between dissimilar system topologies. Additionally, each unique system architecture has unique control requirements, thus increasing the complexity of the overall design problem. Therefore, this chapter proposed a relaxed approach to the traditionally discrete topology co-design problem wherein a single model and controller is designed and adapted to a continuum of feasible system architectures, thus facilitating a continuous search of the discrete design space.

An architecture generation method from the literature is used to identify the set of all structurally feasible design architectures. Instead of representing each unique topology as an adjacency matrix, the novel approach converts each topology to a bitstring representation that represents connectivity between components. This bitstring representation converts the integer topology problem into a binary one that can be relaxed to represent the discrete topology design space as a convex and continuous space that can be more efficiently searched. An essential element of this approach is the development of a superset model; enabled by the modular graph-based modeling framework that represents connections between components as edges of a dynamic graph. A complementary architecture adaptive LQR controller was formulated, so a controller can be synthesized for a continuum of the system models, thus facilitating a search of the continuous design space. The methods are used to identify a mission-specific design of an aircraft thermal management system. The results demonstrated that a class of single flow path designs are optimal at low power and energy levels but are unable to respect constraints for more challenging missions. To obey thermal limits for more difficult missions, multiple flow paths are required to provide fresh coolant to each load independently, albeit at the cost of a more complex controller with increased control effort. By relaxing the topology design problem and using a distance-sorted search, a near optimal design solution could be identified an order of magnitude faster than the conventional exhaustive search approach.

The relaxed system representation approach was able to more efficiently search the discrete design space while providing design insights. However, other optimization approaches, such as a penalty method [119], may provide better performance. Alternatively, an energy-based distance metric [54] may be better suited for a distance-sorted search approach. Another natural extension should include plant sizing design into the optimization process or explore alternative control approaches. Lastly, Figure 5.20 illustrated that many designs may be structurally feasible but are unable to respect dynamic operating constraints. Therefore, methods should be developed that identify the subset of dynamically feasible system designs in order to reduce the total number of designs that would need to be optimized.

Chapter 6

Conclusion

6.1 Summary of Research Contributions

Electrified aircraft can provide a more sustainable future while offering a unique set of novel capabilities. However, electrified aircraft are complex; they are multi-functional and comprised of multi-domain and multi-timescale systems with strict operational limits. Addressing all of these challenges when designing a vehicle is difficult. Therefore, this dissertation proposes a physics-based and generalizable mission-specific approach to aircraft design that considers both the vehicle and its operation within the design process.

To meet this need, Chapter 2 introduces the graph-based modeling approach for conservation based systems. The graph-based modeling framework is energy domain agnostic, optimization-oriented, modular, scalable, and variable fidelity, which facilities the development of optimization-based decision-making tools presented in this thesis. In particular, modularity is a key requirement for the architecture optimization methods presented in Chapter 5. The graph-based modeling framework was introduced with model-order reduction, linearization, and discretization techniques that are useful when integrating these models into optimization algorithms.

Chapter 3 presents a hierarchical model predictive controller for an aircraft's integrated power, propulsion, and thermal management system. The hierarchical control framework facilities a computationally efficient application of model predictive control to the aircraft's multi-domain and multi-timescale energy management system. In comparison to a baseline control approach, the hierarchical control provides superior performance and efficiency while better respecting power and thermal constraints for safe operation. The validation of the hierarchical model predictive controller on an experimental platform that represents an aircraft IPPTS was a key aspect of this contribution.

Chapter 4 proposes an approach for combined plant sizing and CLCD. In contrast to conventional OLCD, the CLCD formulation treats the feedback control law as an additional constraint in the optimization, thus accounting for closed-loop dynamics when designing the plant. The advantage to the CLCD approach is a physically implementable system and jointly optimal plant and controller. Because system-level design can be computationally expensive, a convergence analysis is proposed to systematically evaluate the tradeoff between computation time and optimization accuracy. The proposed co-design method is compared against alternative design strategies through the development of a hybrid UAV energy storage system, propulsion system, and feedback controllers. The results demonstrated improvements in closed-loop system performance and provided valuable insights on battery and motor sizing and controller tuning.

Chapter 5 introduces a system-level co-design problem for combined architecture and closed-loop control. The novelty of the approach is the relaxation of the integer topology selection problem into a continuous form, thus facilitating a more efficient search of the discrete topology design space. A superset graph-based model and architecture adaptive LQR were formulated to represent a continuum of feasible topology and control designs. The methods were used to design an aircraft thermal management system and its feedback controller for 25 different missions characterized by unique thermal power and energy requirements. The results generated mission-specific design insights, illustrating that single flow path thermal management systems can provide superior performance at low power and energy levels, but require multi-flow path topologies for more challenging missions.

6.2 Future Work

This dissertation presents an initial approach into system-level co-design for aircraft energy management systems. Additional investigations into alternative applications and design disciples are relevant to consider.

6.2.1 Observer Design

In practical systems, full state feedback is either too expensive or not possible, and therefore the control design requires an observer. Like control design, tuning the observer may be a complex process that can limit closed-loop performance. Therefore, observer design should also be considered within the system-level co-design process.

6.2.2 Spatial Packaging of Interconnected Systems

While the topology optimization methods introduced in Chapter 4 develop physically meaningful topologies, it discounts the relative complexity between different system architectures. Some designs are easy to integrate into a physical system, while others may be more difficult. This relative difficulty of system integration should be considered when selecting an optimal design. Therefore, the proposed topology and control design techniques should be combined with a spatial packaging algorithm as designed in [124] to evaluate both operational and integration constraints and objectives.

6.2.3 Multi-Vehicle Design and Coordination

Unmanned air vehicles' low cost and diverse set of capabilities lend themselves to fleet applications where multiple coordinated vehicles operate cohesively to complete a greater control task. While research efforts have considered multi-agent system coordination, there exists a gap in multi-vehicle system co-design. Therefore, future work should consider both plant and control design of multi-agent systems to understand how a fleet of vehicles should be composed to most effectively complete a mutual task.

6.2.4 Mission Planning

The co-design methods presented in this dissertation primarily considered regulatory control while neglecting mission level coordination. Describing and creating the missions that vehicles are designed for is a key step in the mission-specific design process. Research should investigate mission planning and methods to encode the mission objectives and constraints into the mission-specific design framework.

6.2.5 Robust Design

This dissertation considered the design of systems for either a single or class of missions. However, uncertainty is present in physical systems due to environmental conditions, system degradation, changing objectives, and modeling error. To account for such uncertainty, robust co-design methods should be developed and evaluated. Importantly, these methods should be computationally efficient as to not increase the already significant computational requirements for system-level co-design problems.

6.2.6 Experimental Validation

While some studies have validated control systems on physical platforms, few have experimentally validated co-design methods. Because the co-design process makes various assumptions about system dynamics and operating requirements, the theoretical results never translate directly to a physical design implementation. Quantifying and accounting for that design error would result in more meaningful insights and system designs.

Bibliography

[1] D. Herron, “Hanks GM EV1, saving the world Dave,” 2018. [Online]. Available: https://www.wwwatts.net/videos/history/gm-ev1/ev1-tom-hanks.html

[2] L. D. Burns and C. Shulgan, *Autonomy - The Quest to Build the Driverless Car-And How It Will Reshape Our World.* Ecco, 2018.

[3] “Hybrid-Electric, Plug-in Hybrid-Electric and Electric Vehicle Sales,” 2022. [Online]. Available: https://www.bts.gov/content/gasoline-hybrid-and-electric-vehicle-sales

[4] R. E. Quigley, “More electric aircraft,” in *Conf. Procedings - IEEE Appl. Power Electron. Conf. Expo. - APEC.* IEEE, 1993, pp. 906–911.

[5] J. Mukhopadhaya and B. Graver, “Performance analysis of Regional Electric Aircraft,” The International Council on Clean Transportation, Tech. Rep., 2022.

[6] “Heart Aerospace unveils new airplane design, confirms Air Canada and Saab as new shareholders,” 2022. [Online]. Available: https://tinyurl.com/ycx7xc2h

[7] B. Graver, D. Rutherford, and S. Zheng, “CO2 Emissions from Commercial Aviation,” The International Council on Clean Transportation, Tech. Rep., 2020.

[8] L. A. Garrow, B. J. German, and C. E. Leonard, “Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research,” *Transp. Res. Part C*, vol. 132, 2021.

[9] J. W. Leverton, “Helicopter noise: What is the problem?” *Vertiflite*, vol. 60, no. 2, pp. 12–15, 2014.

[10] R. V. Simonsen, M. Hartung, K. C. Brejndal-Hansen, S. Yding Sørensen, K. O. Sylvester-Hvid, and D. Klein, “Global Trends of Unmanned Aerial Systems,” Danish Technological Institute, Tech. Rep., 2019.

[11] H. E. Perez, J. B. Siegel, X. Lin, and A. G. Stefanopoulou, “Parameterization and Validation of an Integrated Electro-Thermal Cylindrical LFP Battery Model,” in *Annu. Dyn. Syst. Control Conf.* ASME, 2012.

[12] B. J. M. Kolly, J. Panagiotou, and B. A. Czech, “The Investigation of a Lithium-Ion Battery Fire Onboard a Boeing 787 by the US National Transportation Safety Board,” National Transportation Safety Board, Tech. Rep., 2013.

[13] J. P. Koeln, H. C. Pangborn, M. A. Williams, M. L. Kawamura, and A. G. Alleyne, "Hierarchical Control of Aircraft Electro-Thermal Systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 4, pp. 1218–1232, 2020.

[14] H. K. Fathy, J. A. Reyer, P. Y. Papalambros, and A. G. Ulsoy, "On the coupling between the plant and controller optimization problems," in *Am. Control Conf.* IEEE, 2001.

[15] E. Silvas, T. Hofman, N. Murgovski, L. F. P. Etman, and M. Steinbuch, "Review of Optimization Strategies for System-Level Design in Hybrid Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 57–70, 2017.

[16] S. R. Peddada, D. R. Herber, H. C. Pangborn, A. G. Alleyne, and J. T. Allison, "Optimal flow control and single split architecture exploration for fluid-based thermal management," *J. Mech. Des. Trans. ASME*, vol. 141, no. 8, 2019.

[17] D. R. Herber, J. T. Allison, R. Buettner, P. Abolmoali, and S. S. Patnaik, "Architecture generation and performance evaluation of aircraft thermal management systems through graph-based techniques," *AIAA Scitech 2020 Forum*, pp. 1–18, 2020.

[18] D. J. Docimo, Z. Kang, K. A. James, and A. G. Alleyne, "A novel framework for simultaneous topology and sizing optimization of complex, multi-Domain systems-of-Systems," *J. Mech. Des.*, vol. 142, no. 9, pp. 1–16, 2020.

[19] T. Hofman, S. Ebbesen, and L. Guzzella, "Topology optimization for hybrid electric vehicles with automated transmissions," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2442–2451, 2012.

[20] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Design of hybrid-electric vehicle architectures using auto-generation of feasible driving modes," in *Proc. ASME Des. Eng. Tech. Conf.*, 2013.

[21] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Optimal Dual-Mode Hybrid Electric Vehicle Powertrain Architecture Design for a Variety of Loading Scenarios," in *Proc. ASME 2014 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.* ASME, 2014.

[22] G. P. Huang, D. B. Doman, M. W. Oppenheimer, A. Tipton, and D. O. Sigthorsson, "Topology optimization of a fuel thermal management system," in *AIAA Aviat. 2019 Forum.* AIAA, 2019.

[23] J. P. Jasa, C. A. Mader, and J. R. Martins, "Trajectory optimization of a supersonic aircraft with a thermal fuel management system," *Multidiscip. Anal. Optim. Conf.*, pp. 1–16, 2018.

[24] E. S. Hendricks, E. D. Aretskin-Hariton, J. W. Chapman, J. S. Gray, R. D. Falck, and E. H. Gov, "Propulsion System Optimization for a Turboelectric Tiltwing Urban Air Mobility Aircraft," *Int. Soc. Air Breath. Engines Conf.*, 2019.

[25] C. A. Lupp, D. L. Clark, C. T. Aksland, and A. G. Alleyne, "Mission and Shape Optimization of a HALE Aircraft including Transient Power and Thermal Constraints," in *AIAA Aviat. 2022 Forum.* AIAA, 2022.

[26] O. Borsboom, C. A. Fahdzyana, T. Hofman, and M. Salazar, "A Convex Optimization Framework for Minimum Lap Time Design and Control of Electric Race Cars," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8478–8489, 2021.

[27] G. Riva, S. Radrizzani, G. Panzani, M. Corno, and S. M. Savaresi, "An Optimal Battery Sizing Co-Design Approach for Electric Racing Cars," *IEEE Control Syst. Lett.*, vol. 6, pp. 3074–3079, 2022.

[28] E. J. Adler, B. J. Brelje, and J. R. Martins, "Thermal Management System Optimization for a Parallel Hybrid Aircraft Considering Mission Fuel Burn," *Aerospace*, vol. 9, no. 5, 2022.

[29] D. L. Clark and P. C. Abolmoali, "Gradient-Based Optimization of Time-Dependent Aircraft Subsystems under Uncertainty," in *AIAA Aviat. Aeronaut. Forum Expo.* AIAA, 2021.

[30] Q. Xun, N. Murgovski, and Y. Liu, "Chance-constrained robust co-design optimization for fuel cell hybrid electric trucks," *Appl. Energy*, vol. 320, no. December 2021, 2022.

[31] S. Azad, M. Behtash, A. Houshmand, and M. J. Alexander-Ramos, "PHEV powertrain co-design with vehicle performance considerations using MDSDO," *Struct. Multidiscip. Optim.*, vol. 60, pp. 1155–1169, 2019.

[32] S. Li, C. Gu, M. Xu, J. Li, P. Zhao, and S. Cheng, "Optimal power system design and energy management for more electric aircrafts," *J. Power Sources*, vol. 512, no. July, 2021.

[33] P. Renkert, "Component-Based Design Optimization of Multirotor Aircraft," MS Thesis, University of Illinois Urbana Champaign-Urbana, IL, 2022.

[34] C. Laird, Z. Kang, K. James, and A. Alleyne, "Framework for integrated plant and control optimization of electro-thermal systems: an energy storage system case study," *Energy*, vol. 258, 2022.

[35] C. A. Fahdzyana, M. Salazar, and T. Hofman, "Integrated Plant and Control Design of a Continuously Variable Transmission," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4212–4224, 2021.

[36] J. A. Reyer and P. Y. Papalambros, "Combined optimal design and control with application to an electric DC motor," *J. Mech. Des.*, vol. 124, no. 2, pp. 183–191, 2002.

[37] S. F. Alyaqout, P. Y. Papalambros, and A. G. Ulsoy, "Combined robust design and robust control of an electric DC motor," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 3, pp. 574–582, 2011.

[38] A. L. Nash and N. Jain, "Combined Plant and Control Co-Design for Robust Disturbance Rejection in Thermal-Fluid Systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2532–2539, 2020.

[39] M. J. Kim and H. Peng, "Power management and design optimization of fuel cell/battery hybrid vehicles," *J. Power Sources*, vol. 165, no. 2, pp. 819–832, 2007.

[40] A. L. Nash, H. C. Pangborn, and N. Jain, "Robust Control Co-Design with Receding-Horizon MPC," in *Proc. Am. Control Conf.* IEEE, 2021.

[41] C. C. Lin, H. Peng, J. W. Grizzle, and J. M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 6, pp. 839–849, nov 2003.

[42] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-based energy management of a power-split hybrid electric vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 3, pp. 593–603, 2012.

[43] W. Dunham, B. Hencey, I. Kolmanovsky, and A. Girard, "Scenario based stochastic MPC for more electric aircraft coordinated engine and power management," in *Proc. Am. Control Conf.* IEEE, 2019.

[44] J. Hou, J. Sun, and H. Hofmann, "Integrated control of power generation, electric motor and hybrid energy storage for all-electric ships," in *Proc. Am. Control Conf.* IEEE, 2016.

[45] J. B. Rawlings, N. R. Patel, M. J. Risbeck, C. T. Maravelias, M. J. Wenzel, and R. D. Turney, "Economic MPC and real-time decision making with application to large-scale HVAC energy systems," *Comput. Chem. Eng.*, vol. 114, pp. 89–98, 2018.

[46] W. Dunham, B. Hencey, A. R. Girard, and I. Kolmanovsky, "Distributed Model Predictive Control for More Electric Aircraft Subsystems Operating at Multiple Time Scales," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2177–2190, 2020.

[47] W. Dunham, B. Hencey, A. Girard, and I. Kolmanovsky, "Distributed MPC via ADMM for coordination and control of more electric aircraft subsystems," in *ASME Dyn. Syst. Control Conf.* ASME, 2017.

[48] M. A. Velasquez, J. Barreiro-Gomez, N. Quijano, A. I. Cadena, and M. Shahidehpour, "Distributed model predictive control for economic dispatch of power systems with high penetration of renewable energy resources," *Int. J. Electr. Power Energy Syst.*, vol. 113, pp. 607–617, 2019.

[49] R. Halvgaard, L. Vandenberghe, N. K. Poulsen, H. Madsen, and J. B. Jørgensen, "Distributed Model Predictive Control for Smart Energy Systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1675–1682, 2016.

[50] F. Kennel, D. Gorges, and S. Liu, "Energy management for smart grids with electric vehicles based on hierarchical MPC," *IEEE Trans. Ind. Informatics*, vol. 9, no. 3, pp. 1528–1537, 2013.

[51] M. R. Amini, I. Kolmanovsky, and J. Sun, "Hierarchical MPC for Robust Eco-Cooling of Connected and Automated Vehicles and Its Application to Electric Vehicle Battery Thermal Management," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 316–328, 2021.

[52] H. C. Pangborn, C. E. Laird, and A. G. Alleyne, "Hierarchical Hybrid MPC for Management of Distributed Phase Change Thermal Energy Storage," in *Proc. Am. Control Conf.* IEEE, 2020.

[53] H. C. Pangborn, J. P. Koeln, M. A. Williams, and A. G. Alleyne, "Experimental validation of graph-based hierarchical control for thermal management," *J. Dyn. Syst. Meas. Control*, vol. 140, no. 10, 2018.

[54] D. Docimo and A. G. Alleyne, "Electro-Thermal Graph-Based Modeling for Hierarchical Control with Application to an Electric Vehicle," in *IEEE Conf. Control Technol. Appl.* IEEE, 2018.

[55] G. Seenumani, J. Sun, and H. Peng, "A hierarchical optimal control strategy for power management of hybrid power systems in all electric ships applications," *Proc. IEEE Conf. Decis. Control*, pp. 3972–3977, 2010.

[56] Q. Hu, M. R. Amini, H. Wang, I. Kolmanovsky, and J. Sun, "Integrated Power and Thermal Management of Connected HEVs via Multi-Horizon MPC," in *Proc. Am. Control Conf.* IEEE, 2020.

[57] C. T. Aksland and A. G. Alleyne, "Hierarchical model-based predictive controller for a hybrid UAV powertrain," *Control Eng. Pract.*, vol. 115, 2021.

[58] G. S. Hornby, H. Lipson, and J. B. Pollack, "Generative representations for the automated design of modular physical robots," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 703–719, 2003.

[59] L. C. Schmidt and J. Cagan, "GGREADA: A Graph Grammar-Based Machine Design Algorithm," *Res. Eng. Des.*, vol. 9, pp. 195–213, 1997.

[60] D. R. Herber, T. Guo, and J. T. Allison, "Enumeration of Architectures with Perfect Matchings," *J. Mech. Des.*, vol. 139, no. 5, pp. 1–13, 2017.

[61] D. R. Herber, "Enhancements to the perfect matching approach for graph enumeration-based engineering challenges," in *Proc. ASME Des. Eng. Tech. Conf.* ASME, 2020.

[62] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Topology Generation for Hybrid Electric Vehicle Architecture Design," *J. Mech. Des.*, vol. 138, no. 8, 2016.

[63] D. R. Herber and J. T. Allison, “A Problem Class with Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions,” *J. Mech. Des. Trans. ASME*, vol. 141, no. 10, 2018.

[64] L. Whittle and M. Sagliano, “Stochastic optimal trajectory generation via multivariate polynomial chaos,” *AIAA Guid. Navig. Control Conf.*, 2018.

[65] A. P. Deshmukh, D. R. Herber, and J. T. Allison, “Bridging the gap between open-loop and closed-loop control in co-design: A framework for complete optimal plant and control architecture design,” in *Proc. Am. Control Conf.* IEEE, 2015.

[66] A. K. Sundarrajan and D. R. Herber, “Towards a Fair Comparison between the Nested and Simultaneous Control Co-Design Methods using an Active Suspension Case Study,” in *Proc. Am. Control Conf.* IEEE, 2021.

[67] H. S. Yan and G. J. Yan, “Integrated control and mechanism design for the variable input-speed servo four-bar linkages,” *Mechatronics*, vol. 19, no. 2, pp. 274–285, 2009.

[68] F. Eastep, N. Khot, and R. Grandhi, “Improving the active vibrational control of large space structures through structural modifications,” *Acta Astronaut.*, vol. 15, no. 6-7, pp. 383–389, 1987.

[69] W. K. Belvin and K. C. Park, “Structural tailoring and feedback control synthesis - an interdisciplinary approach,” *J. Guid. Control. Dyn.*, vol. 13, no. 3, pp. 424–429, 1990.

[70] D. R. Herber and J. T. Allison, “Nested and simultaneous solution strategies for general combined plant and control design problems,” *J. Mech. Des.*, vol. 141, no. 1, pp. 1–11, 2019.

[71] S. S. Rao, “Combined structural and control optimization of flexible structures,” *Eng. Optim.*, vol. 13, no. 1, pp. 1–16, 1988.

[72] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[73] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Control - A review,” *J. Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[74] N. A. Diangelakis, B. Burnak, J. Katz, and E. N. Pistikopoulos, “Process design and control optimization: A simultaneous approach by multi-parametric programming,” *AIChE J.*, vol. 63, no. 11, pp. 4827–4846, 2017.

[75] P. T. McCarthy, K. McCarthy, M. Hasan, M. Boyd, M. Chang, E. Walters, and N. Niedbalski, “A Multi-Domain Component Based Modeling Toolset for Dynamic Integrated Power and Thermal System Modeling,” in *SAE Tech. Pap.* SAE International, 2019.

[76] B. P. Rasmussen and A. G. Alleyne, “Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems,” Ph.D. dissertation, University of Illinois Urbana-Champaign, Urbana, IL, 2005.

[77] M. Williams, S. Sridharan, S. Banerjee, C. Mak, C. Pauga, P. Krein, A. Alleyne, A. Jacobi, and S. D. Urso, “PowerFlow: A Toolbox for Modeling and Simulation of Aircraft Systems,” *SAE Tech. Pap.*, 2015.

[78] F. Milano, “An open source power system analysis toolbox,” *IEEE Power Eng. Soc. Gen. Meet.*, vol. 20, no. 3, pp. 1199–1206, 2006.

[79] J. L. Lin and T. J. Yeh, “Modeling, identification and control of air-conditioning systems,” *Int. J. Refrig.*, vol. 30, no. 2, pp. 209–220, 2007.

[80] T. Ersal, H. K. Fathy, and J. L. Stein, “Structural Simplification of Modular Bond-Graph Models Based on Junction Inactivity,” *Simul. Model. Pract. Theory*, vol. 17, pp. 175–196, 2009.

[81] H. C. Pangborn, “Hierarchical Control for Multi-Domain Coordination of Vehicle Energy Systems with Switched Dynamics,” PhD Dissertation, University of Illinois Urbana-Champaign, Urbana, IL, 2019.

[82] H. Behjati, A. Davoudi, and F. Lewis, “Modular DC-DC converters on graphs: Cooperative control,” *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6725–6741, 2014.

[83] K. L. Moore, T. L. Vincent, F. Lashhab, and C. Liu, “Dynamic consensus networks with application to the analysis of building thermal processes,” *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 3078–3083, 2011.

[84] H. A. Preisig, “A graph-theory-based approach to the analysis of large-scale plants,” *Comput. Chem. Eng.*, vol. 33, no. 3, pp. 598–604, 2009.

[85] P. J. Tannous and A. G. Alleyne, “Multilevel hierarchical estimation for thermal management systems of electrified vehicles with experimental validation,” *J. Dyn. Syst. Meas. Control*, vol. 142, no. 11, 2020.

[86] K. Deng, S. Goyal, P. Barooah, and P. G. Mehta, “Structure-preserving model reduction of nonlinear building thermal models,” *Automatica*, vol. 50, no. 4, pp. 1188–1195, 2014.

[87] K. M. Russell, C. T. Aksland, and A. G. Alleyne, “Graph-Based Dynamic Modeling of Two-Phase Heat Exchangers in Vapor Compression Systems,” *Int. J. Refrig.*, vol. 137, pp. 244–256, 2022.

[88] S. Lall, P. Krysl, and J. E. Marsden, “Structure-preserving model reduction for mechanical systems,” *Phys. D*, vol. 184, no. 1-4, pp. 304–318, 2003.

[89] P. V. Kokotovic, R. E. O'Malley, and P. Sannuti, "Singular perturbations and order reduction in control theory - An overview," *Automatica*, vol. 12, no. 2, pp. 123–132, 1976.

[90] Safiullah, Y. K. Gupta, and B. S. Prajapati, "Reduced Order Modeling Using Singular Perturbation Method," in *Int. Conf. Power Energy, Environ. Intell. Control.* IEEE, 2007.

[91] C. T. Aksland, "Modular Modeling and Control of a Hybrid Unmanned Aerial Vehicle's Powertrain," MS Thesis, University of Illinois Urbana Champaign-Urbana, IL, 2019.

[92] W. Wang and J. P. Koeln, "Hierarchical multi-timescale energy management for hybrid-electric aircraft," in *Procedings ASME Dyn. Syst. Control Conf.* ASME, 2020.

[93] R. H. Jansen, C. Bowman, A. Jankovsky, R. Dyson, and J. Felder, "Overview of NASA electrified aircraft propulsion research for large subsonic transports," NASA, Cleveland, Tech. Rep., 2017.

[94] D. B. Doman, "Fuel flow control for extending aircraft thermal endurance part I: Underlying principles," *AIAA Guid. Navig. Control Conf.*, 2016.

[95] R. A. Roberts and D. D. Decker, "Control architecture study focused on energy savings of an aircraft thermal management system," *J. Dyn. Syst. Meas. Control*, vol. 136, no. 4, pp. 1–11, 2014.

[96] W. Dunham, B. Hencey, I. Kolmanovsky, and A. Girard, "Predictive propulsion and power control for large transient power loads in a More Electric Aircraft," in *Proc. Am. Control Conf.* IEEE, 2017.

[97] N. J. Butt, R. A. Roberts, and S. S. Patnaik, "Laser diode optical output dependence on junction temperature for high-power laser systems," *Opt. Laser Technol.*, vol. 125, 2020.

[98] L. C. Herrera and B. H. Tsao, "Analysis and Control of Energy Storage in Aircraft Power Systems with Pulsed Power Loads," *SAE Int. J. Aerosp.*, vol. 9, no. 1, pp. 8–13, 2016.

[99] P. Koeln, Justin, "Hierarchical Power Management in Vehicle Systems," PhD Dissertation, University of Illinois Urbana Champaign-Urbana, IL, 2016.

[100] M. A. Williams, "A Framework for the Control of Electro-Thermal Aircraft Power Systems," Ph.D. dissertation, University of Illinois Urbana-Champaign, Urbana, IL, 2017.

[101] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*, 2nd ed. Nob Hill Publishing, 2016.

[102] J. P. Koeln and A. G. Alleyne, "Two-Level Hierarchical Mission-Based Model Predictive Control," in *Proc. Am. Control Conf.* IEEE, 2018.

[103] D. B. Doman, “Fuel flow topology and control for extending aircraft thermal endurance,” *J. Thermophys. Heat Transf.*, vol. 32, no. 1, pp. 35–50, 2018.

[104] W. Shabbir, “Control Strategies for Series Hybrid Electric Vehicles,” PhD Dissertation, Imperial College London, 2015.

[105] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, “Multi-Parametric Toolbox 3.0,” in *Proc. Eur. Control Conf.* IEEE, 2013.

[106] G. L. Plett, “Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 1: Introduction and state estimation,” *J. Power Sources*, vol. 161, no. 2, pp. 1356–1368, 2006.

[107] E. A. Wan and R. van der Merwe, “Unscented Kalman filter for non-linear estimation,” *Geomatics Inf. Sci. Wuhan Univ.*, vol. 31, no. 2, pp. 180–183, 2001.

[108] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in *Proc. IEEE Int. Symp. Comput. Control Syst. Des.* IEEE, 2004.

[109] I. I. Cplex, “V12.1: User’s Manural for CPLEX,” 2009.

[110] Y. Jiang, Y. Wang, S. A. Bortoff, and Z. P. Jiang, “An iterative approach to the optimal co-design of linear control systems,” *Int. J. Control*, vol. 89, no. 4, pp. 680–690, 2016.

[111] L. Serrao, S. Onori, and G. Rizzoni, “ECMS as a realization of pontryagin’s minimum principle for HEV control,” in *Proc. Am. Control Conf.* IEEE, 2009.

[112] R. D. Falck and J. S. Gray, “Optimal control within the context of multidisciplinary design, analysis, and optimization,” *AIAA Scitech 2019 Forum*, no. January, pp. 1–18, 2019.

[113] A. E. Bryson, M. N. Desai, and W. C. Hoffman, “Energy-state approximation in performance optimization of supersonic aircraft,” *J. Aircr.*, vol. 6, no. 6, pp. 481–488, 1969.

[114] D. Raymer, *Aircraft Design: A Conceptual Approach*, 6th ed. AIAA, 2018.

[115] J. S. Gray, J. T. Hwang, J. R. Martins, K. T. Moore, and B. A. Naylor, “OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization,” *Struct. Multidiscip. Optim.*, vol. 59, no. 4, pp. 1075–1104, 2019.

[116] R. Falck, J. Gray, K. Ponnapalli, and T. Wright, “dymos: A Python package for optimal control of multidisciplinary systems,” *J. Open Source Softw.*, vol. 6, no. 59, p. 2809, 2021.

[117] P. Virtanen and Etal., “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nat. Methods*, vol. 17, no. 3, pp. 261–272, 2020.

[118] J. R. Martins and A. Ning, *Engineering Design Optimization*, 1st ed. Cambridge University Press, 2021.

[119] J. D. Deaton and R. V. Grandhi, “A survey of structural and multidisciplinary continuum topology optimization: Post 2000,” *Struct. Multidiscip. Optim.*, vol. 49, no. 1, pp. 1–38, 2014.

[120] S. Roy, W. A. Crossley, B. K. Stanford, K. T. Moore, and J. S. Gray, “A mixed integer efficient global optimization algorithm with multiple infill strategy-applied to a wing topology optimization problem,” *AIAA Scitech 2019 Forum*, 2019.

[121] H.-T. C. Pedro and M. H. Kobayashi, “On a cellular division method for topology optimization,” *Int. J. Numer. Methods Eng.*, vol. 88, pp. 1175–1197, 2011.

[122] S. Boyd and L. Vandenberghe, *Convex Optimization*, 7th ed. Cambridge University Press, 2009.

[123] E. A. Walters, S. Iden, K. McCarthy, M. Amrhein, T. O’Connell, B. Raczkowski, J. Wells, P. Lamm, M. Wolff, K. Yerkes, W. Borger, and B. Wampler, “INVENT modeling, simulation, analysis and optimization,” in *AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.* AIAA, 2010.

[124] S. R. Peddada, L. E. Zeidner, H. T. Ilies, K. A. James, and J. T. Allison, “Toward Holistic Design of Spatial Packaging of Interconnected Systems with Physical Interactions (SPI2),” *J. Mech. Des. Trans. ASME*, vol. 144, no. 12, 2022.

[125] J. R. Magnus, “On the concept of matrix derivative,” *J. Multivar. Anal.*, vol. 101, no. 9, pp. 2200–2206, 2010.

Appendix A

Experimental Testbed Components

The powertrain and thermal management system testbeds' (Figure 3.1) parts lists are provided in Tables A.1 and A.2. Additional testbed details can be found in [81,91].

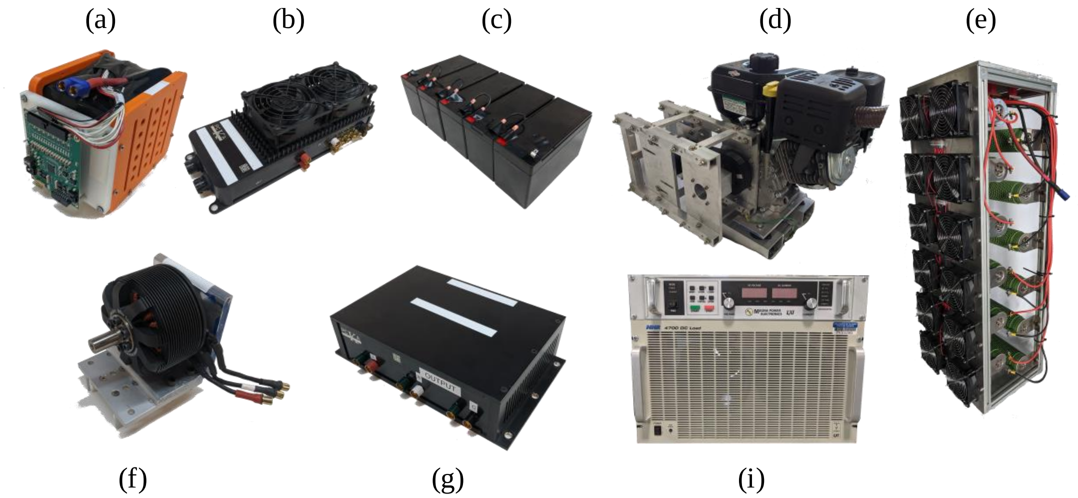


Figure A.1: Powertrain testbed (Figure 3.1a) parts list. (a) battery pack, (b) electronic speed controller, (c) low power battery pack, (d) genset, (e) resistor bank, (f) propulsion and dynamometer motor, (g) filter box, and (h) power supply and electronic load. Image credit [91].

Table A.1: The powertrain testbed parts list.

Component	Manufacturer	Details
Battery Pack	Custom	16S7P Samsung 18650 Cells
Low Power Battery Pack	Expert Power	5S1P EXP1270 Lead Acid Cells
Electronic Speed Controller (x3)	PCKA	-
Filter Box (x2)	PCKA	-

Continued on next page ↪

Table A.1 – continued from previous page

Component	Manufacturer	Details
Propulsion Motor	NeuMotors	Series 8038-105
Dynamometer Motor	NeuMotors	Series 8038-105
Starter/Generator	NeuMotors	Series 8038-140
Power Resistors (x12)	TE Connectivity	1Ω 2S6P
Engine	Briggs and Stratton	19N1 Series
Power Supply	Magna-Power	4kW Rating (XR400-10.0 Series Supply)
Electronic Load	NH Research	3kW Rating (4700-3-TP Series)
Battery Management System	Texas Instruments	bq76PL455A-Q1 Evaluation Board
CompactDAQ	National Instruments	4 slots (cDAQ-9174)
Voltage Input Module	National Instruments	Reads shunt resistors (NI-9205)
Digital I/O Module	National Instruments	Controls relays (NI-9403)
High Current Shunt (x2)	Rideon	200A rating (RSB-200-100)
Low Current Shunt (x4)	Rideon	100A rating (RSB-100-100)
USB to CAN Adapter	Intrepid Control Sys.	Facilitates CAN communication
Control Computer	Dell	7th Gen Intel Core i7 with 8GB RAM

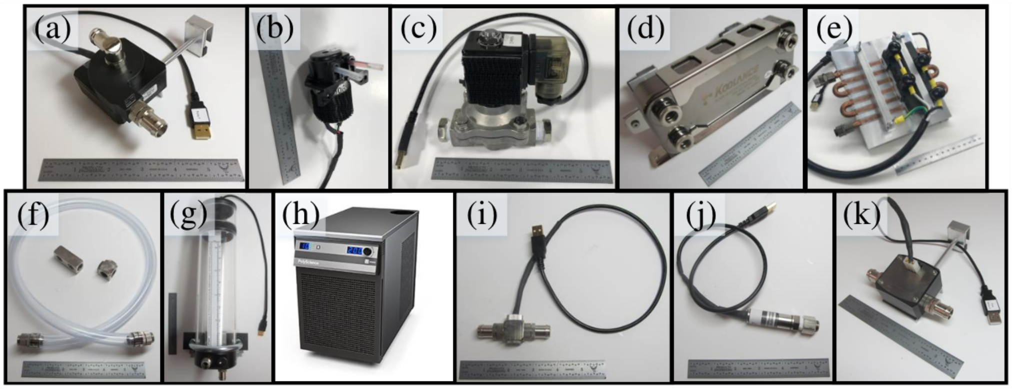


Figure A.2: Thermal management system testbed (Figure 3.1b) parts list. (a) centrifugal pump, (b) positive displacement pump, (c) solenoid valve, (d) heat exchanger, (e) cold plate, (f) pipe, (g) reservoir, (h) chiller, (i) temperature sensor, (j) pressure sensor, and (k) mass flow rate sensor. Image credit [81].

Table A.2: The thermal management system testbed parts list.

Component	Manufacturer	Details
Centrifugal Pump	Swiftech	MCP35X 17.5 LPM
Positive Disp. Pump	Gikfun	Peristaltic pump 0.1LPM
Solenoid Valve	WIC Valve	2SCW 150PSI
Heat Exchanger	Koolance	HXP-193 4.0kW
Cold Plate	Wakefield-Vette	6-pass 6in. cold plate
Pipe	Koolance	HOS-13CL
Reservior	Koolance	80x240mm
Chiller	Polyscience	6000 Series 5.0kW
Temperature Sensor	Koolance	10kΩ thermistor SEN-AP008B(fluid) & SEN-AP007P(surface)
Pressure Sensor	Measurement Specialties	US300 310kPa guage
Flow Rate Sensor	Aqua Computer	High Flow 0.5 to 25LPM
Current Sensor	Sparkfun	ACS712 low current
Resistive Heaters	Vishay	LPS1100H47R0JB resistors, 47 ?, 1.1kW
Current Sensor	Echun Electronic Co.	ECS1030-L72 non-invasive current sensor

Appendix B

Graph-Based Models

B.1 Aircraft Integrated Power, Propulsion, and Thermal System

The aircraft IPPTS graph-based model introduced in Chapter 3 is shown below (Figure B.1) with vertex, edge, input, and parameter tables (Tables B.1-B.4).

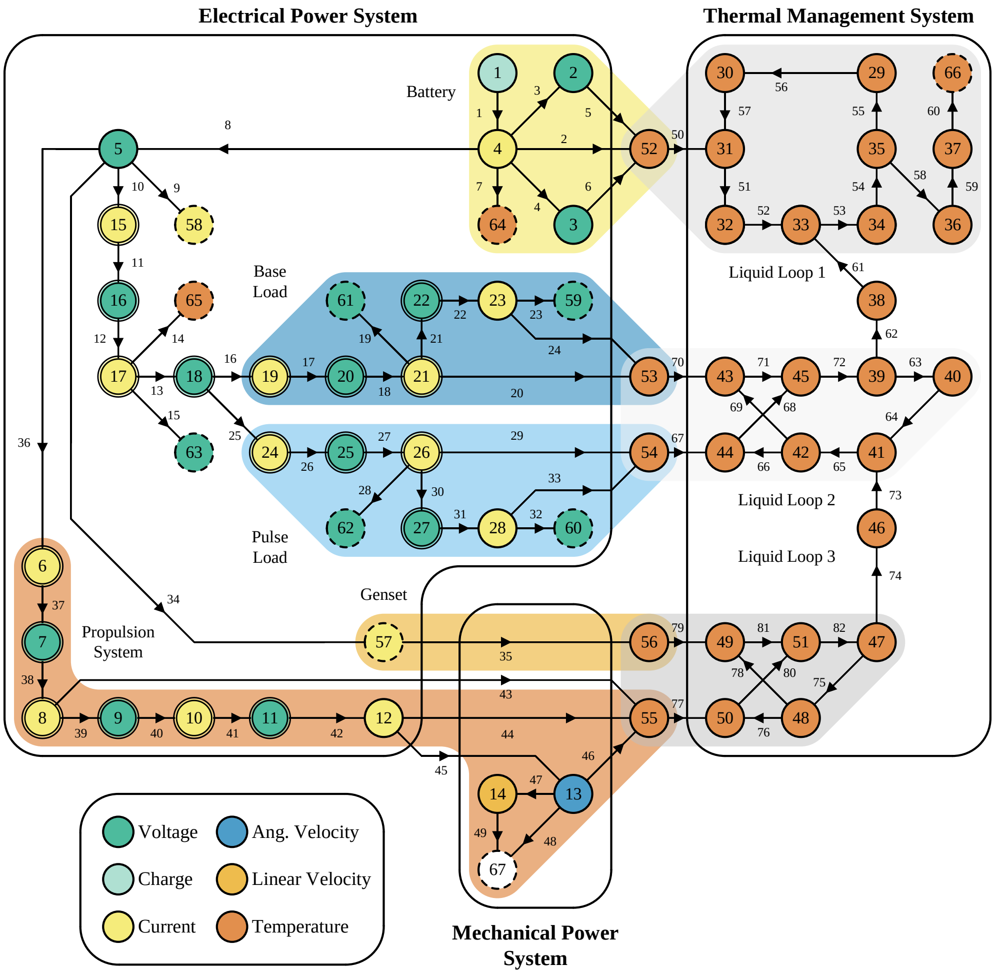


Figure B.1: Graph-model of the aircraft IPPTS illustrated in Figure 3.1.

Table B.1: Vertex information for the IPPTS graph model.

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
1	x_1	Battery State of Charge	-	$QV_{ocv}\left(x_1\right)$	0.3	0.9
2	x_2	RC Pair Voltage 1	V	C_1x_2	-	-
3	x_3	RC Pair Voltage 2	V	C_2x_3	-	-
4	x_4	Battery Current	A	L_bx_4	-10.5	105
5	x_5	Bus Voltage	V	C_bx_5	-	-
6	a_6	Inverter DC Current	A	0	-	-
7	a_7	Inverter DC Link Voltage	V	0	-	-
8	a_8	Inverter Y q-axis Current	A	0	20	100
9	a_9	Inverter Y q-axis Voltage	V	0	-	-
10	a_{10}	Inverter Δ Current	A	0	-	-
11	a_{11}	Inverter Δ Voltage	V	0	-	-
12	x_{12}	Prop. Current	A	L_px_{12}	20	100
13	x_{13}	Prop Speed	rad/s	J_px_{13}	50π	167π
14	x_{14}	Vehicle Speed	m/s	m_vx_{14}	15	22.5
15	a_{15}	LV Converter Current	A	0	-	-
16	a_{16}	LV Converter DC Link Voltage	V	0	-	-
17	a_{17}	Net Avionic Load Current	A	0	-	-
18	a_{18}	Avionics Voltage	V	0	39	41
19	a_{19}	Base Converter Current	A	0	-	-
20	a_{20}	LV Converter Voltage	V	0	-	-
21	a_{21}	Base Converter Load	A	0	-	-
22	a_{22}	Base Converter Voltage	V	0	-	-
23	x_{23}	Base Load Current	A	$L_{base}x_{23}$	5	12
24	a_{24}	Pulse Converter Current	A	0	-	-
25	a_{25}	LV Converter Voltage	V	0	-	-
26	a_{26}	Pulse Converter Load	A	0	-	-
27	a_{27}	Pulse Converter Voltage	V	0	-	-
28	x_{28}	Pulse Load Current	A	$L_{pulse}x_{28}$	0	25
29	x_{29}	Loop 1 Tank Temp.	°C	$m_{29}c_{p,f}$	5	40
30	x_{30}	Pipe Temp.	°C	$m_{30}c_{p,f}$	5	40

Continued on next page ↪

Table B.1 – continued from previous page

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
31	x_{31}	Battery CP Fluid Temp.	°C	$m_{31}c_{p,f}$	5	40
32	x_{32}	Pipe Temp.	°C	$m_{32}c_{p,f}$	5	40
33	x_{33}	Loop 1-2 HX Side 1 Temp.	°C	$m_{33}c_{p,f}$	5	40
34	x_{34}	Return Temp.	°C	$m_{34}c_{p,f}$	5	40
35	x_{35}	Loop C-1 HX Side 2 Temp.	°C	$m_{35}c_{p,f}$	5	40
36	x_{36}	Loop C-1 HX Wall Temp.	°C	C_{36}	5	40
37	x_{37}	Loop C-1 HX Side 1 Temp.	°C	$m_{37}c_{p,f}$	5	40
38	x_{38}	Loop 1-2 HX Wall Temp.	°C	C_{38}	5	40
39	x_{39}	Loop 1-2 HX Side 2 Temp.	°C	$m_{29}c_{p,f}$	5	45
40	x_{40}	Loop 2 Tank Temp.	°C	$m_{40}c_{p,f}$	5	45
41	x_{41}	Loop 2-3 HX Side 1 Temp.	°C	$m_{41}c_{p,f}$	5	45
42	x_{42}	Split Temp.	°C	$m_{42}c_{p,f}$	5	45
43	x_{43}	Base CP Fluid Temp.	°C	$m_{43}c_{p,f}$	5	45
44	x_{44}	Pulse CP Fluid Temp.	°C	$m_{44}c_{p,f}$	5	45
45	x_{45}	Junction Temp.	°C	$m_{45}c_{p,f}$	5	45
46	x_{46}	Loop 2-3 HX Wall Temp.	°C	C_{46}	5	45
47	x_{47}	Loop 2-3 HX Side 2 Temp.	°C	$m_{47}c_{p,f}$	5	50
48	x_{48}	Split Temp.	°C	$m_{48}c_{p,f}$	5	50
49	x_{49}	Prop. CP Fluid Temp.	°C	$m_{49}c_{p,f}$	5	50
50	x_{50}	Genset CP Fluid Temp.	°C	$m_{50}c_{p,f}$	5	50
51	x_{51}	Junction Temp	°C	$m_{51}c_{p,f}$	5	50
52	x_{52}	Battery Temp.	°C	C_{52}	5	50
53	x_{53}	Base Temp.	°C	C_{53}	5	50
54	x_{54}	Pulse Temp.	°C	C_{54}	5	50
55	x_{55}	Prop. Temp.	°C	C_{55}	5	50
56	x_{56}	Genset Temp.	°C	C_{56}	5	50
57	x^e_{57}	Genset Current	A	-	-	-
58	x^e_{58}	ESC Current	A	-	-	-
59	x^e_{59}	Base Load Voltage	V	-	-	-
60	x^e_{60}	Pulse Load Voltage	V	-	-	-
61	x^e_{61}	Base Converter Diode Voltage	V	-	-	-

Continued on next page ↪

Table B.1 – continued from previous page

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
62	x^e_{62}	Pulse Converter Diode Voltage	V	-	-	-
63	x^e_{63}	LV Converter Diode Voltage	V	-	-	-
64	x^e_{64}	Bleed Resistor Sink	°C	-	-	-
65	x^e_{65}	LV Converter Resistor Sink	°C	-	-	-
66	x^e_{66}	Chiller Temp.	°C	-	-	-
67	x^e_{67}	Misc.	-	-	-	-

Table B.2: Edge information for the IPPTS graph model.

Edge	Description	Power flow
1	Battery Power	$V_{ocv}(x_1)x_4$
2	Resistive Loss/Heat Load	$R_s x_4^2$
3	Electrical Power	$x_4 x_2$
4	Electrical Power	$x_4 x_3$
5	Resistive Loss/Heat Load	$\frac{1}{R_1}x_2^2$
6	Resistive Loss/Heat Load	$\frac{1}{R_2}x_3^2$
7	Resistive Loss/Heat Load	$R_b x_4^2$
8	Electrical Power	$x_4 x_5$
9	Electrical Power	$x_5 x^e_{58}$
10	Electrical Power	$x_5 a_{15}$
11	Electrical Power	$a_{15} a_{16}$
12	Controlled Electrical Power	$u_1 a_{16} a_{17}$
13	Electrical Power	$a_{17} a_{18}$
14	Switching & Conduction Loss	$(R_{s,LV} u_1 + R_{c,LV}) a_{17}^2$
15	Diode Loss	$(1 - u_1) a_{17} x^e_{63}$
16	Electrical Power	$a_{18} a_{19}$
17	Electrical Power	$a_{19} a_{20}$
18	Controlled Electrical Power	$u_2 a_{20} a_{21}$

Continued on next page ↪

Table B.2 – continued from previous page

Edge	Description	Power flow
19	Diode Loss	$(1-u_2)a_{21}x^e_{61}$
20	Switching & Conduction Loss	$(R_{s,base}u_2+R_{c,base})a^2_{21}$
21	Electrical Power	$a_{21}a_{22}$
22	Electrical Power	$a_{22}x_{23}$
23	Electrical Power	$x_{23}x^e_{59}$
24	Temp. Dependent Resistive Loss/Heat Load	$R_{base}(x_{53})x^2_{23}$
25	Electrical Power	$a_{18}a_{24}$
26	Electrical Power	$a_{24}a_{25}$
27	Controlled Electrical Power	$u_3a_{25}a_{26}$
28	Diode Loss	$(1-u_3)a_{26}x^e_{62}$
29	Switching & Conduction Loss	$(R_{s,pulse}u_3+R_{c,pulse})a^2_{26}$
30	Electrical Power	$a_{26}a_{27}$
31	Electrical Power	$a_{27}x_{28}$
32	Electrical Power	$x_{28}x^e_{60}$
33	Temp. Dependent Resistive Loss/Heat Load	$R_{pulse}(x_{54})x^2_{28}$
34	Electrical Power	$x_5x^e_{57}$
35	Resistive Loss/Heat Load	$R_g{x^e_{57}}^2$
36	Electrical Power	x_5a_6
37	Electrical Power	a_6a_7
38	Controlled Electrical Power	$u_4a_7a_8$
39	Electrical Power	a_8a_9
40	Electrical Power	a_9a_{10}
41	Y to Δ Conversion	$\sqrt{\frac{1}{3}}a_{10}a_{11}$
42	Electrical Power	$a_{11}x_{12}$
43	Resistive Loss/Heat Load	$u_5R_ia^2_8$
44	Resistive Loss/Heat Load	$R_px^2_{12}$
45	Electro-magnetics	$k_vx_{12}x_{13}$
46	Friction Loss	$b_px^2_{13}+c_px_{13}$
47	Propulsion Power	$\rho\frac{D^4}{4\pi^2}C_Tx^2_{13}x_{14}$
48	Drag	$(1-\eta_p)\rho\frac{D^5}{4\pi^2}C_\tau x^3_{13}$
49	Aircraft Drag and Weight	$m_vg\sin\theta x_{14}+\frac{1}{2}\rho AC_Dx^3_{14}$
50	Convection	$U_{batt}(x_{52}-x_{31})$

Continued on next page ↪

Table B.2 – continued from previous page

Edge	Description	Power flow
51	Advection	$u_6 c_{p,f} x_{31}$
52	Advection	$u_6 c_{p,f} x_{32}$
53	Advection	$u_6 c_{p,f} x_{33}$
54	Advection	$u_6 c_{p,f} x_{34}$
55	Advection	$u_6 c_{p,f} x_{35}$
56	Advection	$u_6 c_{p,f} x_{29}$
57	Advection	$u_6 c_{p,f} x_{30}$
58	Convection	$U_1(x_{35} - x_{36})$
59	Convection	$U_1(x_{36} - x_{37})$
60	Bi-directional Advection	$u_5 c_{p,f}(x_{37} - x_{66}^e)$
61	Convection	$U_2(x_{38} - x_{33})$
62	Convection	$U_2(x_{39} - x_{38})$
63	Advection	$u_9 c_{p,f} x_{39}$
64	Advection	$u_9 c_{p,f} x_{40}$
65	Advection	$u_9 c_{p,f} x_{41}$
66	Advection	$u_8 c_{p,f} x_{42}$
67	Convection	$U_{pulse}(x_{54} - x_{44})$
68	Advection	$u_8 c_{p,f} x_{44}$
69	Advection	$u_7 c_{p,f} x_{42}$
70	Convection	$U_{base}(x_{53} - x_{43})$
71	Advection	$u_7 c_{p,f} x_{43}$
72	Advection	$u_9 c_{p,f} x_{45}$
73	Convection	$U_3(x_{46} - x_{41})$
74	Convection	$U_3(x_{47} - x_{46})$
75	Advection	$u_{12} c_{p,f} x_{47}$
76	Advection	$u_{11} c_{p,f} x_{48}$
77	Convection	$U_{genset}(x_{55} - x_{50})$
78	Advection	$u_{10} c_{p,f} x_{48}$
79	Convection	$U_{prop}(x_{56} - x_{49})$
80	Advection	$u_{11} c_{p,f} x_{50}$
81	Advection	$u_{10} c_{p,f} x_{49}$
82	Advection	$u_{12} c_{p,f} x_{51}$

Table B.3: Input information for the IPPTS model.

Input	Description	Units	Lower Bound	Upper Bound
u_1	LV Converter Duty Cycle	-	0.01	0.99
u_2	Base Converter Duty Cycle	-	0.01	0.99
u_3	Pulse Converter Duty Cycle	-	0.01	0.99
u_4	Inverter Duty Cycle	-	0.01	0.99
u_5	Chiller Flow Rate	kg/s	0.3	0.3
u_6	Loop 1 Flow Rate	kg/s	0.04	0.13
u_7	Loop 2 Through Line Flow Rate	kg/s	Figure 3.5	Figure 3.5
u_8	Loop 2 Valve Line Flow Rate	kg/s	Figure 3.5	Figure 3.5
$u_9 = u_7 + u_8$	Loop 2 Net Flow Rate	kg/s	Figure 3.5	Figure 3.5
u_{10}	Loop 3 Through Line Flow Rate	kg/s	Figure 3.5	Figure 3.5
u_{11}	Loop 3 Valve Line Flow Rate	kg/s	Figure 3.5	Figure 3.5
$u_{12} = u_{10} + u_{11}$	Loop 3 Net Flow Rate	kg/s	Figure 3.5	Figure 3.5
u_{13}	Genset Input	-	0.0	3.3

Table B.4: Parameter information for the IPPTS model.

Parameter	Description	Units	Value
ρ	Air Density	kg/m^3	1.225
θ	Angle of Attack	°	3.6
$R_{c,base}$	Base Converter Conduction Resistance	mΩ	23.9
$R_{s,base}$	Base Converter Switching Resistance	mΩ	24.2
L_{base}	Base Load Inductance	μH	100
$r_{1,base}$	Base Load Resistance Coefficient	Ω/°C	1/150
$r_{0,base}$	Base Load Resistance Coefficient	Ω	0.1
R_{base}	Base Resistance	Ω	(3.1)
Q	Battery Capacity	Ah	21.33
C_1	Battery RC Capacitance 1	F	862

Continued on next page ↪

Table B.4 – continued from previous page

Parameter	Description	Units	Value
C_2	Battery RC Capacitance 2	F	2.31×10^4
R_1	Battery RC Resistance 1	mΩ	63.8
R_2	Battery RC Resistance 2	mΩ	78.1
R_s	Battery Series Resistance	mΩ	47.6
C_b	Bus Capacitance	F	0.1
R_b	Bus Resistance	mΩ	0.5
C_D	Drag Constant	-	0.0167
m_{29}	Fluid Mass	kg	1.9
m_{30}	Fluid Mass	kg	0.014
m_{31}	Fluid Mass	kg	1.1
m_{32}	Fluid Mass	kg	0.014
m_{33}	Fluid Mass	kg	1.1
m_{34}	Fluid Mass	kg	0.014
m_{35}	Fluid Mass	kg	0.57
m_{37}	Fluid Mass	kg	0.57
m_{39}	Fluid Mass	kg	0.71
m_{40}	Fluid Mass	kg	0.86
m_{41}	Fluid Mass	kg	0.28
m_{42}	Fluid Mass	kg	0.014
m_{43}	Fluid Mass	kg	1.1
m_{44}	Fluid Mass	kg	1.1
m_{45}	Fluid Mass	kg	0.014
m_{47}	Fluid Mass	kg	0.28
m_{48}	Fluid Mass	kg	0.014
m_{49}	Fluid Mass	kg	1.1
m_{50}	Fluid Mass	kg	1.1
m_{51}	Fluid Mass	kg	0.014
$c_{p,f}$	Fluid Specific Heat Capacity	J/kg-K	3498
a_5	Genset sfc Coefficient	g/kW-A^{2}h	0.22
a_4	Genset sfc Coefficient	g/kW-V^{2}h	0.089
a_3	Genset sfc Coefficient	g/kW-VAh	0.26
a_2	Genset sfc Coefficient	g/kW-Ah	-33.2

Continued on next page ↪

Table B.4 – continued from previous page

Parameter	Description	Units	Value
a_1	Genset sfc Coefficient	g/kW-Vh	-21.0
a_0	Genset sfc Coefficient	g/kW-h	1.50
K	Genset Input Gain	A	63.3
b_1	Genset Limit Coefficient	1/V	-0.015
b_0	Genset Limit Coefficient	-	1.71
R_g	Genset Resistance	mΩ	425
tau	Genset Time Constant	s	0.5
g	Gravitational Acceleration	m/s^2	9.81
C_{36}	Heat Capacity	J/K	45
C_{38}	Heat Capacity	J/K	45
C_{46}	Heat Capacity	J/K	27.5
C_{52}	Heat Capacity	J/K	1866
C_{53}	Heat Capacity	J/K	1800
C_{54}	Heat Capacity	J/K	1800
C_{55}	Heat Capacity	J/K	1800
C_{56}	Heat Capacity	J/K	1800
R_i	Inverter Resistance	mΩ	80.8
$R_{c,LV}$	LV Converter Conduction Resistance	mΩ	23.9
$R_{s,LV}$	LV Converter Switching Resistance	mΩ	24.2
U_{batt}	Overall Heat Transfer Coefficient	W/K	219
U_{base}	Overall Heat Transfer Coefficient	W/K	484
U_{pulse}	Overall Heat Transfer Coefficient	W/K	200
U_{prop}	Overall Heat Transfer Coefficient	W/K	777
U_{genset}	Overall Heat Transfer Coefficient	W/K	536
U_1	Overall Heat Transfer Coefficient	W/K	100×10^3
U_2	Overall Heat Transfer Coefficient	W/K	898
U_3	Overall Heat Transfer Coefficient	W/K	2.36×10^3
L_p	Prop. Coil Inductance	μH	48.7×10^{-6}
R_p	Prop. Coil Resistance	mΩ	29.2
J_p	Prop. Inertia	kg/m^2	2.73×10^{-2}
k_v	Prop. Motor Constant	mNm/A	0.1238
D	Propeller Diameter	m	0.625

Continued on next page ↪

Table B.4 – continued from previous page

Parameter	Description	Units	Value
η_p	Propeller Efficiency	-	0.95
$R_{c,pulse}$	Pulse Converter Conduction Resistance	mΩ	23.9
$R_{s,pulse}$	Pulse Converter Switching Resistance	mΩ	24.2
L_{pulse}	Pulse Load Inductance	μH	100
$r_{1,pulse}$	Pulse Load Resistance Coefficient	Ω/°C	1/200
$r_{0,pulse}$	Pulse Load Resistance Coefficient	Ω	0.05
R_{pulse}	Pulse Resistance	Ω	(3.1)
c_p	Static Friction	mNm	228
C_T	Thrust Constant	-	0.07
C_τ	Torque Constant	-	0.01
m_v	Vehicle Mass	kg	85
L_b	Virtual Inductance	H	4×10^{-8}
b_p	Viscous Friction	mNms/rad	0.36

B.2 Series Hybrid UAV Powertrain

The series hybrid UAV powertrain graph-based model introduced in Chapter 4 is shown below (Figure B.2) with vertex, edge, input, and parameter tables (Tables B.5-B.8).

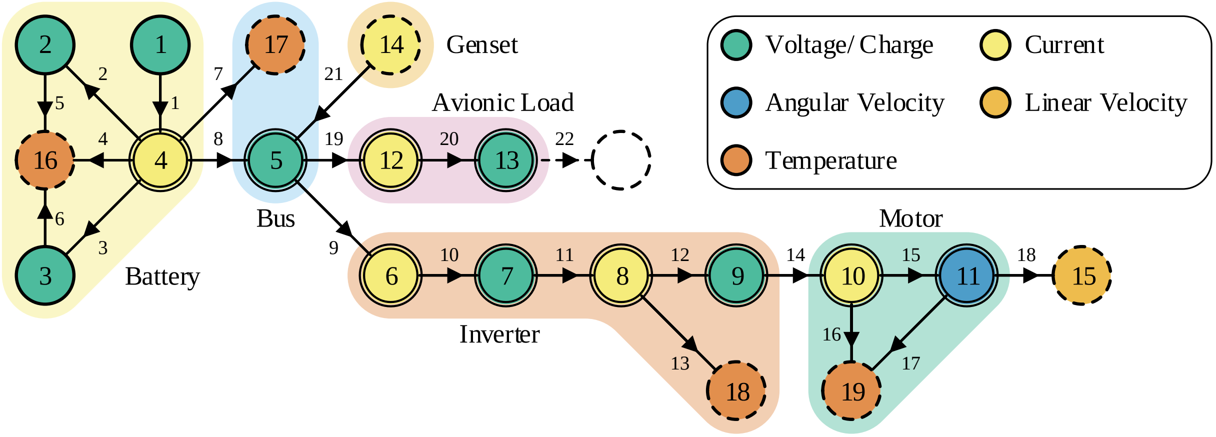


Figure B.2: The graph model for the series hybrid UAV powertrain.

Table B.5: Vertex information for the Hybrid UAV PT graph model.

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
1	x_1	Battery State of Charge	-	$N_sN_pQV_{ocv}(x_1)$	0.3	0.9
2	x_2	RC Pair Voltage 1	V	$\frac{N_p}{N_s}C_1x_2$	-10	10
3	x_3	RC Pair Voltage 2	V	$\frac{N_p}{N_s}C_2x_3$	-10	10
4	a_4	Battery Current	A	0	$-1.5N_s$	$15N_s$
5	a_5	Bus Voltage	V	0	-	-
6	a_6	Inverter DC Current	A	0	-	-
7	a_7	Inverter DC Link Voltage	V	0	-	-
8	a_8	Inverter q-axis Current	A	0	-	-
9	a_9	Inverter q-axis Voltage	V	0	-	-
10	a_{10}	Prop. Current	A	0	0	(4.25f)
11	a_{11}	Prop Speed	rad/s	0	-	-
12	a_{12}	Avionic Load Current	A	0	-	-
13	a_{13}	Avionic Load Voltage	V	0	-	-

Continued on next page ↪

Table B.5 – continued from previous page

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
14	x^e_{14}	Genset Current	A	-	0	50
15	x^e_{15}	Vehicle Velocity	m/s	-	25	50
16	x^e_{16}	Thermal Sink	-	-	-	-
17	x^e_{17}	Thermal Sink	-	-	-	-
18	x^e_{18}	Thermal Sink	-	-	-	-
19	x^e_{19}	Thermal Sink	-	-	-	-

Table B.6: Edge information for the Hybrid UAV PT graph model.

Edge	Description	Power flow
1	Battery Power	$N_s V_{ocv}(x_1) a_4$
2	Electrical Power	$a_4 x_2$
3	Electrical Power	$a_4 x_3$
4	Resistive Loss/Heat Load	$R_s \frac{N_s}{N_p} a_4^2$
5	Resistive Loss/Heat Load	$\frac{1}{R_1} \frac{N_p}{N_s} x_2^2$
6	Resistive Loss/Heat Load	$\frac{1}{R_2} \frac{N_p}{N_s} x_3^2$
7	Resistive Loss/Heat Load	$R_b a_4^2$
8	Electrical Power	$a_4 a_5$
9	Electrical Power	$a_5 a_6$
10	Electrical Power	$a_6 a_7$
11	Controlled Electrical Power	$u_1 a_7 a_8$
12	Electrical Power	$a_8 a_9$
13	Switching & Conduction Loss	$R_i u_1 a_8^2$
14	Electrical Power	$a_9 a_{10}$
15	Electro-magnetics	$k_v a_{10} a_{11}$
16	Resistive Loss/Heat Load	$R_m a_{10}^2$
17	Friction Loss	$b a_{11}^2$
18	Propulsion Power	$\frac{\rho d^4}{\eta} C_T a_{11}^2 x^e_{15}$

Continued on next page ↪

Table B.6 – continued from previous page

Edge	Description	Power flow
19	Electrical Power	$a_5 a_{12}$
20	Electrical Power	$a_{12} a_{13}$
21	Electrical Power	$x_{14}^e a_5$
22	Avionic Load Power	P_{22}^e

Table B.7: Input information for the Hybrid UAV model.

Input	Description	Units	Lower Bound	Upper Bound
u_1	Inverter Duty Cycle	-	0.01	0.99
u_2	Genset Command	-	0.01	0.99
γ	Flight Path Angle	rad	$-\frac{\pi}{6}$	$\frac{\pi}{6}$

Table B.8: Parameter information for the Hybrid UAV model.

Parameter	Description	Units	Default Value	Lower Bound	Upper Bound
$m_{airframe}$	Airframe mass	kg	18	N/A	N/A
R_b	Bus resistance	mΩ	0.5	N/A	N/A
C_1	Cell RC capacitance 1	kF	1.97	N/A	N/A
C_2	Cell RC capacitance 2	kF	52.9	N/A	N/A
R_1	Cell RC resistance 1	mΩ	27.9	N/A	N/A
R_2	Cell RC resistance 2	mΩ	34.2	N/A	N/A
Q	Cell capacity	Ah	3.05	N/A	N/A
m_{cell}	Cell mass	g	45.9	N/A	N/A
R_s	Cell series resistance	mΩ	20.8	N/A	N/A
i_0	Current limit coefficient	A	160	N/A	N/A
i_1	Current limit coefficient	A-rad/V-s	601	N/A	N/A
i_2	Current limit coefficient	A/Ω	−3440	N/A	N/A

Continued on next page ↪

Table B.8 – continued from previous page

Parameter	Description	Units	Default Value	Lower Bound	Upper Bound
K_L	Drag-due-to-lift factor	-	0.05	N/A	N/A
τ	Engine time constant	s	20	N/A	N/A
ε	Equivalence factor	-	18.0	9	30
c_0	Fuel consumption coefficients	kg	7.13×10^{-5}	N/A	N/A
c_1	Fuel consumption coefficients	kg/W	2.79×10^{-7}	N/A	N/A
Q_{LHV}	Fuel lower heating value	MJ/kg	43.4	N/A	N/A
K	Genset input gain	A	63.3	N/A	N/A
m_{genset}	Genset mass	kg	12	N/A	N/A
P_gen	Genset power	kW	–	N/A	N/A
g	Gravitational constant	m/s^2	9.81	N/A	N/A
I	Integral gain	-	0.0020	0.0	0.01
$m_{inv.}$	Inverter mass	kg	1.0	N/A	N/A
R_i	Inverter resistance	mΩ	80.8	N/A	N/A
m_{load}	Load mass	kg	7.1	N/A	N/A
α_{max}	Max AoA	°	8	N/A	N/A
$C_{L,max}$	Max lift	-	1.23	N/A	N/A
R_{mot}	Motor friction	mNm-s/rad	0.36	N/A	N/A
m_0	Motor mass coefficients	kg	0.81	N/A	N/A
m_1	Motor mass coefficients	kg-rad/V-s	21.3	N/A	N/A
m_2	Motor mass coefficients	kg/Ω	-39.5	N/A	N/A
N_p	Number of parallel cells	-	7	1	20
N_s	Number of series cells	-	16	1	20
v_5	OCV coefficients	V	13.75	N/A	N/A
v_4	OCV coefficients	V	-40.12	N/A	N/A
v_3	OCV coefficients	V	44.14	N/A	N/A
v_2	OCV coefficients	V	-22.98	N/A	N/A
v_1	OCV coefficients	V	6.73	N/A	N/A
v_0	OCV coefficients	V	2.66	N/A	N/A
R_m	Prop. coil resistance	mΩ	29.2	Fig. 4.11	Fig. 4.11
k_v	Prop. motor constant	V-s/rad	0.1238	Fig. 4.11	Fig. 4.11
d	Propeller diameter	m	0.7	N/A	N/A

Continued on next page ↪

Table B.8 – continued from previous page

Parameter	Description	Units	Default Value	Lower Bound	Upper Bound
η	Propeller efficiency	-	0.8	N/A	N/A
P	Proportional gain	-	0.0371	0.001	0.1
C_T	Thrust coefficient	-	2.1×10^{-3}	N/A	N/A
$x^e_{15,ref}$	Velocity reference	m/s	–	25	50
b	Viscous friction	mNm-s/rad	0.36	N/A	N/A
A	Wing area	MJ/kg	3.4	N/A	N/A
$C_{L,0}$	Zero lift coefficient	-	0.49	N/A	N/A

B.3 Thermal Management System

The thermal management system superset graph-based model introduced in Chapter 5 is shown below (Figure B.3) with vertex, edge, input, and parameter tables (Tables B.9-B.12).

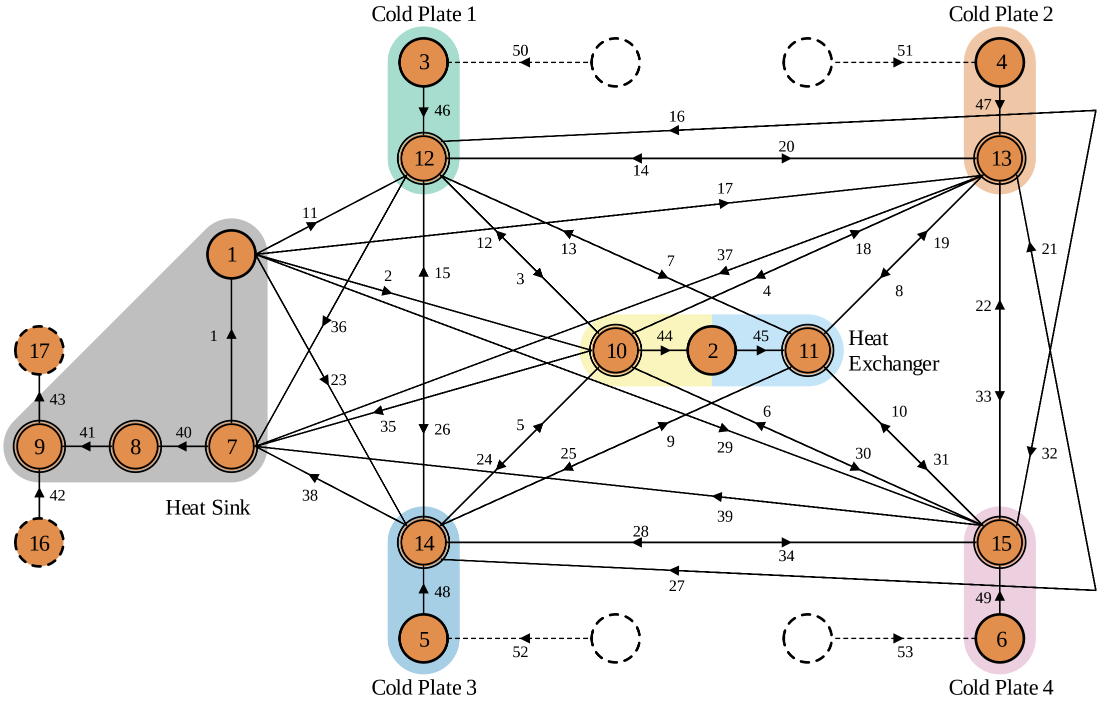


Figure B.3: The superset TMS graph-based model.

Table B.9: Vertex information for the TMS graph model. HX-Heat Exchanger

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
1	x_1	Tank Temp.	°C	$c_{p,f}m_{tank}$	20	70
2	x_2	HX Temp.	°C	C_{hx}	20	70
3	x_3	Load 1 Temp.	°C	C_{cp1}	20	50
4	x_4	Load 2 Temp.	°C	C_{cp2}	20	55
5	x_5	Load 3 Temp.	°C	C_{cp3}	20	60
6	x_6	Load 4 Temp.	°C	C_{cp4}	20	65

Continued on next page ↪

Table B.9 – continued from previous page

Vertex	State Variable	Description	Units	Capacitance	Lower Bound	Upper Bound
7	a_7	Recirculation Temp.	°C	0	-	-
8	a_8	Heat Sink Wall Temp.	°C	0	-	-
9	a_9	Heat Sink Temp.	°C	0	-	-
10	a_{10}	HX Primary Side Temp.	°C	0	-	-
11	a_{11}	HX Secondary Side Temp.	°C	0	-	-
12	a_{12}	Cold Plate 1 Temp.	°C	0	-	-
13	a_{13}	Cold Plate 2 Temp.	°C	0	-	-
14	a_{14}	Cold Plate 3 Temp.	°C	0	-	-
15	a_{15}	Cold Plate 4 Temp.	°C	0	-	-
16	x^e_{16}	Chiller Temp.	°C	-	-	-
17	x^e_{17}	-	°C	-	-	-

Table B.10: Edge information for the TMS graph model.

Edge	Description	Power flow
1	Advection	$\rho_1 u_1 c_{p,f} a_7$
2	Advection	$\rho_2 u_2 c_{p,f} x_1$
3	Advection	$\rho_3 u_3 c_{p,f} a_{12}$
4	Advection	$\rho_4 u_4 c_{p,f} a_{13}$
5	Advection	$\rho_5 u_5 c_{p,f} a_{14}$
6	Advection	$\rho_6 u_6 c_{p,f} a_{15}$
7	Advection	$\rho_7 u_7 c_{p,f} a_{12}$
8	Advection	$\rho_8 u_8 c_{p,f} a_{13}$
9	Advection	$\rho_9 u_9 c_{p,f} a_{14}$
10	Advection	$\rho_{10} u_{10} c_{p,f} a_{15}$
11	Advection	$\rho_{11} u_{11} c_{p,f} x_1$
12	Advection	$\rho_{12} u_{12} c_{p,f} a_{10}$
13	Advection	$\rho_{13} u_{13} c_{p,f} a_{11}$

Continued on next page ↪

Table B.10 – continued from previous page

Edge	Description	Power flow
14	Advection	$\rho_{14}u_{14}c_{p,f}a_{13}$
15	Advection	$\rho_{15}u_{15}c_{p,f}a_{14}$
16	Advection	$\rho_{16}u_{16}c_{p,f}a_{15}$
17	Advection	$\rho_{17}u_{17}c_{p,f}x_{1}$
18	Advection	$\rho_{18}u_{18}c_{p,f}a_{10}$
19	Advection	$\rho_{19}u_{19}c_{p,f}a_{11}$
20	Advection	$\rho_{20}u_{20}c_{p,f}a_{13}$
21	Advection	$\rho_{21}u_{21}c_{p,f}a_{14}$
22	Advection	$\rho_{22}u_{22}c_{p,f}a_{15}$
23	Advection	$\rho_{23}u_{23}c_{p,f}x_{1}$
24	Advection	$\rho_{24}u_{24}c_{p,f}a_{10}$
25	Advection	$\rho_{25}u_{25}c_{p,f}a_{11}$
26	Advection	$\rho_{26}u_{26}c_{p,f}a_{12}$
27	Advection	$\rho_{27}u_{27}c_{p,f}a_{13}$
28	Advection	$\rho_{28}u_{28}c_{p,f}a_{15}$
29	Advection	$\rho_{29}u_{29}c_{p,f}x_{1}$
30	Advection	$\rho_{30}u_{30}c_{p,f}a_{10}$
31	Advection	$\rho_{31}u_{31}c_{p,f}a_{11}$
32	Advection	$\rho_{32}u_{32}c_{p,f}a_{12}$
33	Advection	$\rho_{33}u_{33}c_{p,f}a_{13}$
34	Advection	$\rho_{34}u_{34}c_{p,f}a_{14}$
35	Advection	$\rho_{35}u_{35}c_{p,f}a_{10}$
36	Advection	$\rho_{36}u_{36}c_{p,f}a_{12}$
37	Advection	$\rho_{37}u_{37}c_{p,f}a_{13}$
38	Advection	$\rho_{38}u_{38}c_{p,f}a_{14}$
39	Advection	$\rho_{39}u_{39}c_{p,f}a_{15}$
40	Convection	$U_{hs}(a_7 - a_8)$
41	Convection	$U_{hs}(a_8 - a_9)$
42	Advection	$\dot{m}_{chiller}c_{p,f}x^{e}_{16}$
43	Advection	$\dot{m}_{chiller}c_{p,f}a_{9}$
44	Convection	$U_{hx}(a_{10} - x_2)$
45	Convection	$U_{hx}(x_2 - a_{11})$

Continued on next page ↪

Table B.10 – continued from previous page

Edge	Description	Power flow
46	Convection	$U_{cp}(x_3 - a_{12})$
47	Convection	$U_{cp}(x_4 - a_{13})$
48	Convection	$U_{cp}(x_5 - a_{14})$
49	Convection	$U_{cp}(x_6 - a_{15})$
50	Heat Load 1	P^e_{50}
51	Heat Load 2	P^e_{51}
52	Heat Load 3	P^e_{52}
53	Heat Load 4	P^e_{53}

Table B.11: Input information for the TMS model.

Input	Description	Units	Lower Bound	Upper Bound
u_1	Flow Rate 1	kg/s	0	0.4
u_2	Flow Rate 2	kg/s	0	0.1
u_3	Flow Rate 3	kg/s	0	0.1
u_4	Flow Rate 4	kg/s	0	0.1
u_5	Flow Rate 5	kg/s	0	0.1
u_6	Flow Rate 6	kg/s	0	0.1
u_7	Flow Rate 7	kg/s	0	0.1
u_8	Flow Rate 8	kg/s	0	0.1
u_9	Flow Rate 9	kg/s	0	0.1
u_{10}	Flow Rate 10	kg/s	0	0.1
u_{11}	Flow Rate 11	kg/s	0	0.1
u_{12}	Flow Rate 12	kg/s	0	0.1
u_{13}	Flow Rate 13	kg/s	0	0.1
u_{14}	Flow Rate 14	kg/s	0	0.1
u_{15}	Flow Rate 15	kg/s	0	0.1
u_{16}	Flow Rate 16	kg/s	0	0.1
u_{17}	Flow Rate 17	kg/s	0	0.1

Continued on next page ↪

Table B.11 – continued from previous page

Input	Description	Units	Lower Bound	Upper Bound
u_{18}	Flow Rate 18	kg/s	0	0.1
u_{19}	Flow Rate 19	kg/s	0	0.1
u_{20}	Flow Rate 20	kg/s	0	0.1
u_{21}	Flow Rate 21	kg/s	0	0.1
u_{22}	Flow Rate 22	kg/s	0	0.1
u_{23}	Flow Rate 23	kg/s	0	0.1
u_{24}	Flow Rate 24	kg/s	0	0.1
u_{25}	Flow Rate 25	kg/s	0	0.1
u_{26}	Flow Rate 26	kg/s	0	0.1
u_{27}	Flow Rate 27	kg/s	0	0.1
u_{28}	Flow Rate 28	kg/s	0	0.1
u_{29}	Flow Rate 29	kg/s	0	0.1
u_{30}	Flow Rate 30	kg/s	0	0.1
u_{31}	Flow Rate 31	kg/s	0	0.1
u_{32}	Flow Rate 32	kg/s	0	0.1
u_{33}	Flow Rate 33	kg/s	0	0.1
u_{34}	Flow Rate 34	kg/s	0	0.1
u_{35}	Flow Rate 35	kg/s	0	0.1
u_{36}	Flow Rate 36	kg/s	0	0.1
u_{37}	Flow Rate 37	kg/s	0	0.1
u_{38}	Flow Rate 38	kg/s	0	0.1
u_{39}	Flow Rate 39	kg/s	0	0.1

Table B.12: Parameter information for the TMS model.

Parameter	Description	Units	Value
U_{cp}	CP Overall HTC	W/K	2000
$\dot{m}_{chiller}$	Chiller Flow Rate	kg/s	0.1
$c_{p,f}$	Fluid Specific Heat	J/kg-K	3500

Continued on next page ↪

Table B.12 – continued from previous page

Parameter	Description	Units	Value
U_{hx}	HX Overall HTC	W/K	1000
C_{hx}	HX Thermal Capacity	J/K	937.5
U_{hs}	Heat Sink Overall HTC	W/K	1000
R	Input Weight Matrix	-	10^4
C_{cp1}	Load 1 Thermal Capacity	J/K	375
C_{cp2}	Load 2 Thermal Capacity	J/K	375
C_{cp3}	Load 3 Thermal Capacity	J/K	375
C_{cp4}	Load 4 Thermal Capacity	J/K	375
Q	State Weight Matrix	-	1
m_{tank}	Tank Mass	kg	1.6
ρ	Topology Variables	-	–

Appendix C

Architecture Adaptive LQR Sensitivity Computations

C.1 Background on Matrix Derivatives

An infinite horizon linear quadratic regulator in synthesized through the solution of the Algebraic Riccati Equation (ARE). To facilitate gradient-based design of an LQR class controller, it is necessary to evaluate the sensitivity of the ARE output to its inputs. This evaluation requires knowledge of matrix calculus, since the ARE is a matrix function. Although it might seem reasonable to vectorize the matrices and apply standard vector calculus techniques, the problem is not that trivial. As [125] makes clear, there are multiple ways to compute matrix derivatives that yield different sensitivity results. If interested, it is recommended to read [125]. The following sections introduce basic matrix derivative computation rules that are then applied to the architecture adaptive LQR synthesis process.

Note, the following computations are valid for the OpenMDAO implementation of matrix calculus. As [125] points out, there is not a universal approach for computing matrix derivatives, so special care should be taken when using this approach.

C.2 Matrix Derivatives

C.2.1 Identity, Transpose, Product, and Chain Rule

Define the following matrices:

$A \in \mathbb{R}^{m \times n}$

$B \in \mathbb{R}^{q \times r}$

$X \in \mathbb{R}^{n\times n}$

$Y \in \mathbb{R}^{n\times r}$

$Z \in \mathbb{R}^{q\times n}$

Let $K_{i,j} \in \mathbb{R}^{ij\times ij}$ be the commutation matrix and $I_k \in \mathbb{R}^{k\times k}$ be the identity matrix. The operation $\otimes$ is the Kronecker product. The following identities are useful

$$B \otimes A = K_{q,m}\left(A \otimes B\right)K_{n,r}, \tag{C.1a}$$

$$\left(M \otimes N\right)\left(O \otimes P\right) = \left(MO\right) \otimes \left(NP\right) \text{ for compatible matrices } M, N, O \text{ ,and } P \tag{C.1b}$$

The following matrix operations and their sensitivity functions are provided

Self Differentiation:

$$F\left(A\right) = A, \tag{C.2a}$$

$$\frac{\partial F}{\partial A} = I_{mn} \tag{C.2b}$$

Transposed Self Differentiation:

$$F\left(A\right) = A^T, \tag{C.3a}$$

$$\frac{\partial F}{\partial A} = K_{n,m} \tag{C.3b}$$

Product Rule: for linear matrix functions

$$F\left(A, Y\right) = AY, \tag{C.4a}$$

$$\frac{\partial F}{\partial A} = K_{m,r}\left(Y^T \otimes I_m\right)K_{n,m}, \tag{C.4b}$$

$$= I_m \otimes Y^T, \tag{C.4c}$$

$$\frac{\partial F}{\partial Y} = A \otimes I_r \tag{C.4d}$$

Chain Rule:

$$F\left(X\right) = G\left(H\left(X\right)\right), \tag{C.5a}$$

$$\frac{\partial F}{\partial X} = \frac{\partial G}{\partial H}\frac{\partial H}{\partial X} \tag{C.5b}$$

Product Rule: for general matrix functions. Let $A\left(X\right) \in \mathbb{R}^{m\times n}$ and $Y\left(X\right) \in \mathbb{R}^{n\times r}$

$$F\left(X\right) = AY, \tag{C.6a}$$

$$\frac{\partial F}{\partial X} = K_{m,r}\left(Y^T \otimes I_m\right)K_{n,m}\frac{\partial A}{\partial X} + K_{m,r}\left(I_r \otimes A\right)K_{r,n}\frac{\partial Y}{\partial X}, \tag{C.6b}$$

$$= \left(I_m \otimes Y^T\right)\frac{\partial A}{\partial X} + \left(A \otimes I_r\right)\frac{\partial Y}{\partial X} \tag{C.6c}$$

C.2.2 Key Examples

From the product and chain rule, the partial derivatives of many other matrix functions can be computed. Consider the following notable derivations

Product Rule with Transpose: Make the substitution $C = Z^T$, apply product and chain rules knowing that $\frac{\partial C}{\partial Z} = K_{n,q}$.

$$F\left(B, Z\right) = Z^T B = C\left(Z\right)B, \tag{C.7a}$$

$$\frac{\partial F}{\partial Z} = \left(I_n \otimes B^T\right)\frac{\partial C}{\partial Z} + \left(C \otimes I_r\right)\frac{\partial B}{\partial Z}, \tag{C.7b}$$

$$= \left(I_n \otimes B^T\right)K_{n,q}, \tag{C.7c}$$

$$\frac{\partial F}{\partial B} = Z^T \otimes I_r \tag{C.7d}$$

Matrix Inverse: To derive the partial derivative of the matrix inverse, use the definition of the inverse function $I_n = X^{-1}X$, take the derivative of both sides using the product rule, then solve for $\frac{\partial\left(X^{-1}\right)}{\partial X}$.

$$
\begin{aligned}
F(X) &= X^{-1}, && \text{(C.8a)}\\
\frac{\partial}{\partial X}&(I_n = X^{-1}X), && \text{(C.8b)}\\
0 &= \left(I_n \otimes X^T\right)\frac{\partial\left(X^{-1}\right)}{\partial X} + \left(X^{-1}\otimes I_n\right)\frac{\partial X}{\partial X}, && \text{(C.8c)}\\
0 &= \left(I_n \otimes X^T\right)\frac{\partial\left(X^{-1}\right)}{\partial X} + \left(X^{-1}\otimes I_n\right), && \text{(C.8d)}\\
\frac{\partial\left(X^{-1}\right)}{\partial X} &= -\left(I_n \otimes X^T\right)^{-1}\left(X^{-1}\otimes I_n\right) && \text{(C.8e)}
\end{aligned}
$$

Quadratic Functions: To find $\frac{\partial F}{\partial A}$, use the substitution $C = AX$, apply product and chain rule, and then plug in partials computed in earlier examples. To find $\frac{\partial F}{\partial X}$, do the same thing, except differentiate with respect to X. Use (C.1b) to simplify.

$$
\begin{aligned}
F(A,X) &= AXA^T = C(A,X)A^T, && \text{(C.9a)}\\
\frac{\partial F}{\partial A} &= (I_m \otimes A)\frac{\partial C}{\partial A} + (C \otimes I_m)\frac{\partial\left(A^T\right)}{\partial A}, && \text{(C.9b)}\\
&= (I_m \otimes A)\left(I_m \otimes X^T\right) + (AX \otimes I_m)K_{n,m}, && \text{(C.9c)}\\
&= \left(I_m \otimes AX^T\right) + (AX \otimes I_m)K_{n,m}, && \text{(C.9d)}\\
\frac{\partial F}{\partial X} &= (I_m \otimes A)\frac{\partial C}{\partial X} + (C \otimes I_m)\frac{\partial\left(A^T\right)}{\partial X}, && \text{(C.9e)}\\
&= (I_m \otimes A)(A \otimes I_n), && \text{(C.9f)}\\
&= (A \otimes A) && \text{(C.9g)}
\end{aligned}
$$

C.3 Partial Derivatives for the AALQR

The partial derivative computations for the input constrained variation of the AALQR are provided in this section. The input constrained AALQR synthesis requires the state space matrices $A(\rho) \in \mathbb{R}^{x\times x}$ and $B(\rho) \in \mathbb{R}^{x\times u}$, the input constraints null space $F(\rho) \in \mathbb{R}^{u\times f}$, the state weight matrix $Q \in \mathbb{R}^{x\times x}$, and the input weight matrix $R \in \mathbb{R}^{u\times u}$. The control synthesis outputs a feedback control gain $K(\rho)$ that is a used to control the system. Because the optimization evaluates designs for different values of ρ, Q, and R, it is necessary to compute the sensitivities $\frac{\partial K}{\rho}$, $\frac{\partial K}{Q}$, and $\frac{\partial K}{R}$. However, directly evaluating those sensitivities can be challenging because the ARE is an implicit function, which is why OpenMDAO is used. OpenMDAO provides the option to define the model equations as modular components, from which OpenMDAO can efficiently compute the total derivatives from partial derivatives [115]. Therefore, OpenMDAO can automatically compute $\frac{\partial K}{\rho}$, $\frac{\partial K}{Q}$, and $\frac{\partial K}{R}$ from easier to derive partial derivatives.

The input constrained AALQR synthesis (Section 5.3.4.3) is decomposed into 3 components in OpenMDAO

$$\left.\begin{aligned}\bar{R} &= F^T R F,\\ \bar{B} &= BF\end{aligned}\right\}\text{Component 1,} \tag{C.10a}$$

$$0 = PA + A^T P + Q - P\bar{B}\bar{R}^{-1}\bar{B}^T P\Big\}\text{Component 2,} \tag{C.10b}$$

$$K = \bar{R}^{-1}\bar{B}^T P\Big\}\text{Component 3.} \tag{C.10c}$$

Note that the functional dependency on ρ is omitted for brevity. The following partial derivatives are derived using the methods from Section C.2, so the full derivations are omitted.

Component 1 (C.10a)**:** The outputs are $\bar{R} \in \mathbb{R}^{f\times f}$ and $\bar{B} \in \mathbb{R}^{x\times f}$ with inputs $B \in \mathbb{R}^{x\times u}$, $R \in \mathbb{R}^{u\times u}$, and $F \in \mathbb{R}^{u\times f}$.

$$\frac{\partial \bar{R}}{\partial R} = \left(F^T \otimes F^T\right), \tag{C.11a}$$

$$\frac{\partial \bar{R}}{\partial F} = \left(I_f \otimes (RF)^T\right) K_{f,u} + \left(F^T R \otimes I_f\right), \tag{C.11b}$$

$$\frac{\partial \bar{B}}{\partial B} = I_x \otimes F^T, \tag{C.11c}$$

$$\frac{\partial \bar{B}}{\partial F} = B \otimes I_f \tag{C.11d}$$

Component 2 (C.10b)**:** An implicit equation. The output is $\bar{P} \in \mathbb{R}^{x \times x}$ with inputs $A \in \mathbb{R}^{x \times x}$, $\bar{B} \in \mathbb{R}^{x \times f}$, $\bar{R} \in \mathbb{R}^{f \times f}$, and $Q \in \mathbb{R}^{x \times x}$.

$$\frac{\partial P}{\partial P} = \left(I_x \otimes A^T\right) + \left(A^T \otimes I_x\right) - \left(I_x \otimes \left(\bar{B}\bar{R}^{-1}\bar{B}^T P\right)^T\right) - \left(P\bar{B}\bar{R}^{-1}\bar{B}^T \otimes I_x\right), \tag{C.12a}$$

$$\frac{\partial P}{\partial A} = \left(P \otimes I_x\right) + \left(I_x \otimes P^T\right) K_{x,x}, \tag{C.12b}$$

$$\frac{\partial P}{\partial \bar{B}} = -\left(P \otimes P\right)\left(\left(I_x \otimes \bar{B}\bar{R}^{-1}\right) + \left(\bar{B}\bar{R}^{-1} \otimes I_x\right) K_{f,x}\right), \tag{C.12c}$$

$$\frac{\partial P}{\partial Q} = I_{xx}, \tag{C.12d}$$

$$\frac{\partial P}{\partial \bar{R}} = \left(P\bar{B} \otimes P\bar{B}\right)\left(I_f \otimes \bar{R}^T\right)^{-1}\left(\bar{R}^{-1} \otimes I_f\right) \tag{C.12e}$$

Component 3 (C.10c)**:** The output is $K \in \mathbb{R}^{f \times x}$ with inputs $\bar{B} \in \mathbb{R}^{x \times f}$, $\bar{R} \in \mathbb{R}^{f \times f}$, and $P \in \mathbb{R}^{x \times x}$.

$$\frac{\partial K}{\partial \bar{B}} = \left(\bar{R}^{-1} \otimes P^T\right) K_{f,x}, \tag{C.13a}$$

$$\frac{\partial K}{\partial \bar{R}} = -\left(I_f \otimes \left(\bar{B}^T P\right)^T\right)\left(I_f \otimes R^T\right)^{-1}\left(R^{-1} \otimes I_f\right), \tag{C.13b}$$

$$\frac{\partial K}{\partial P} = \bar{R}^{-1}\bar{B}^T \otimes I_x \tag{C.13c}$$

Appendix D

Supplemental Topology Optimization Figures

This section includes supplemental figures from Chapter 5 and includes all topologies classes (Figure D.1, the best 21 TMS designs (Figure D.2), all state in input trajectories for each mission (Figure D.3), and the distance-sorted search evaluation metrics for each mission (Figure D.4).

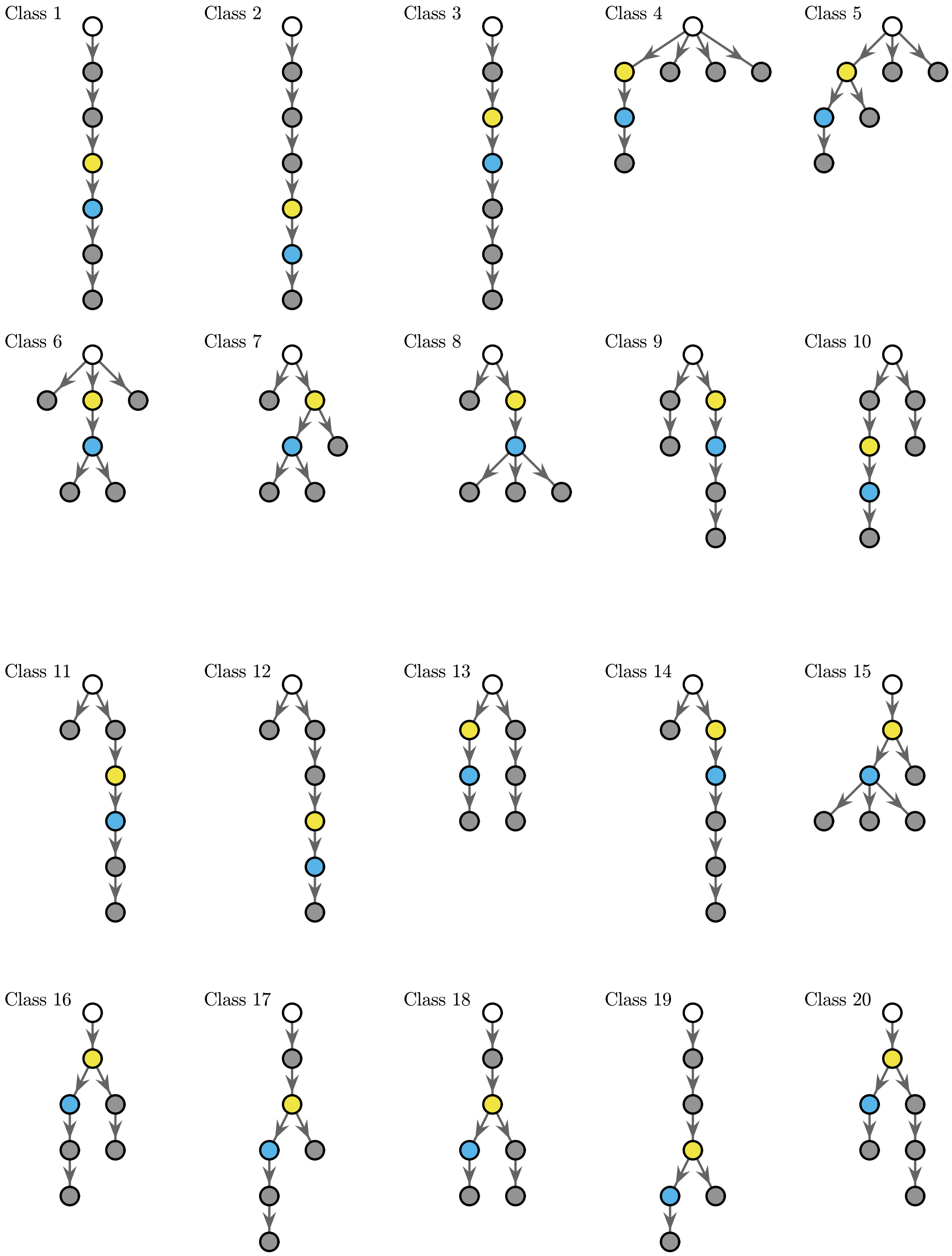


Figure D.1: All 37 unique classes of thermal management system architectures represented by rooted trees.

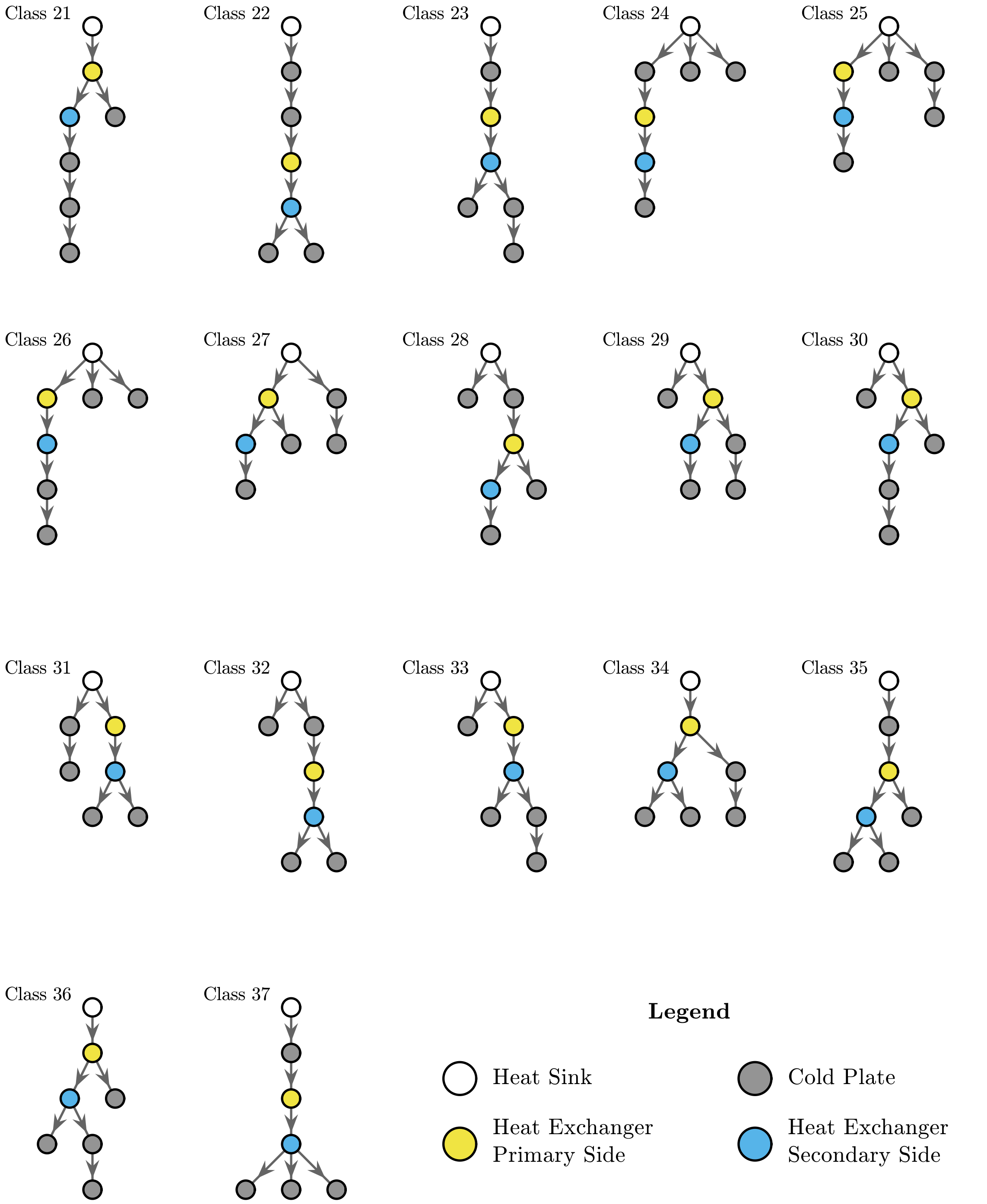


Figure D.1 (cont.): All 37 unique classes of thermal management system architectures represented by rooted trees.

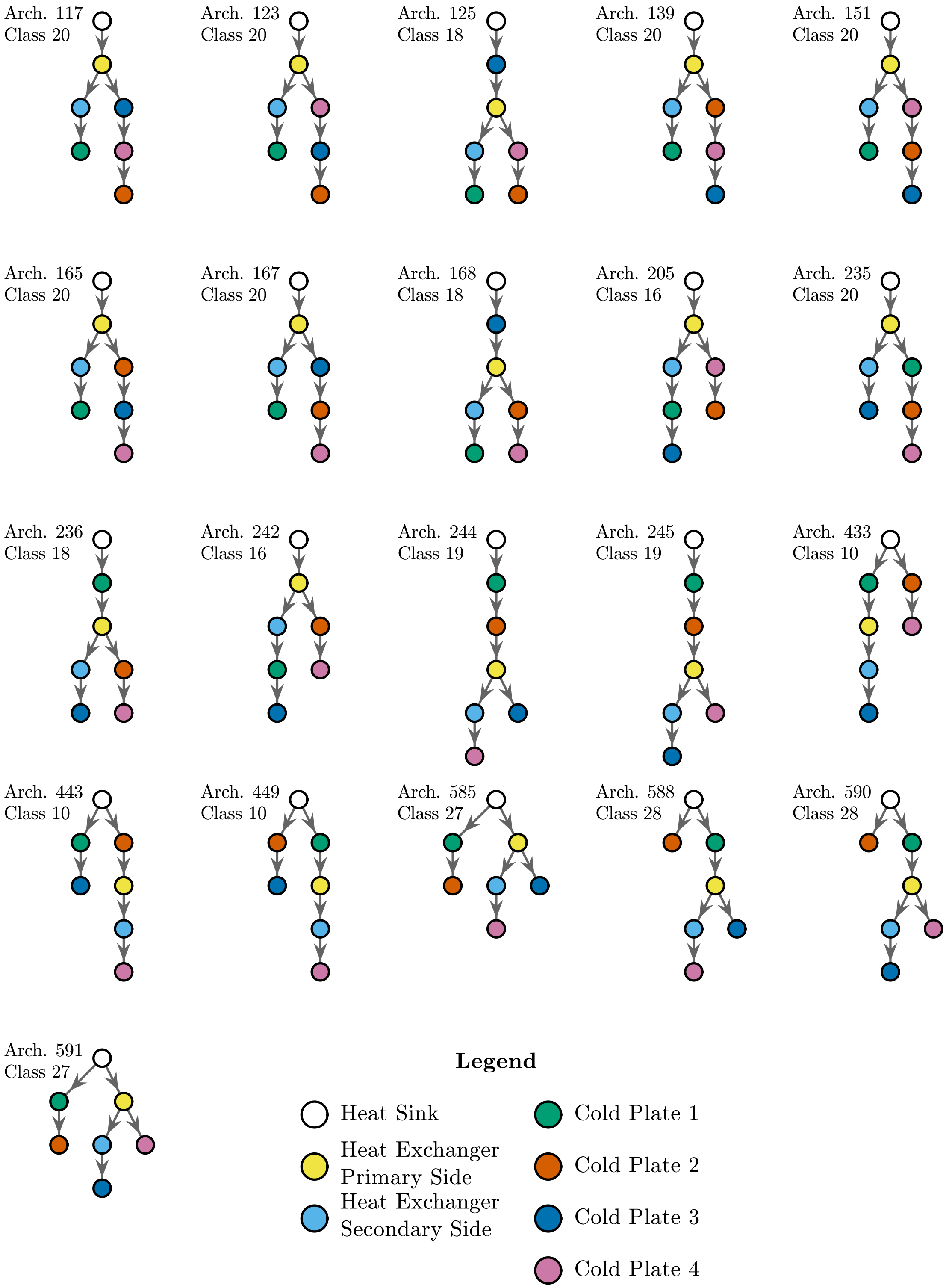
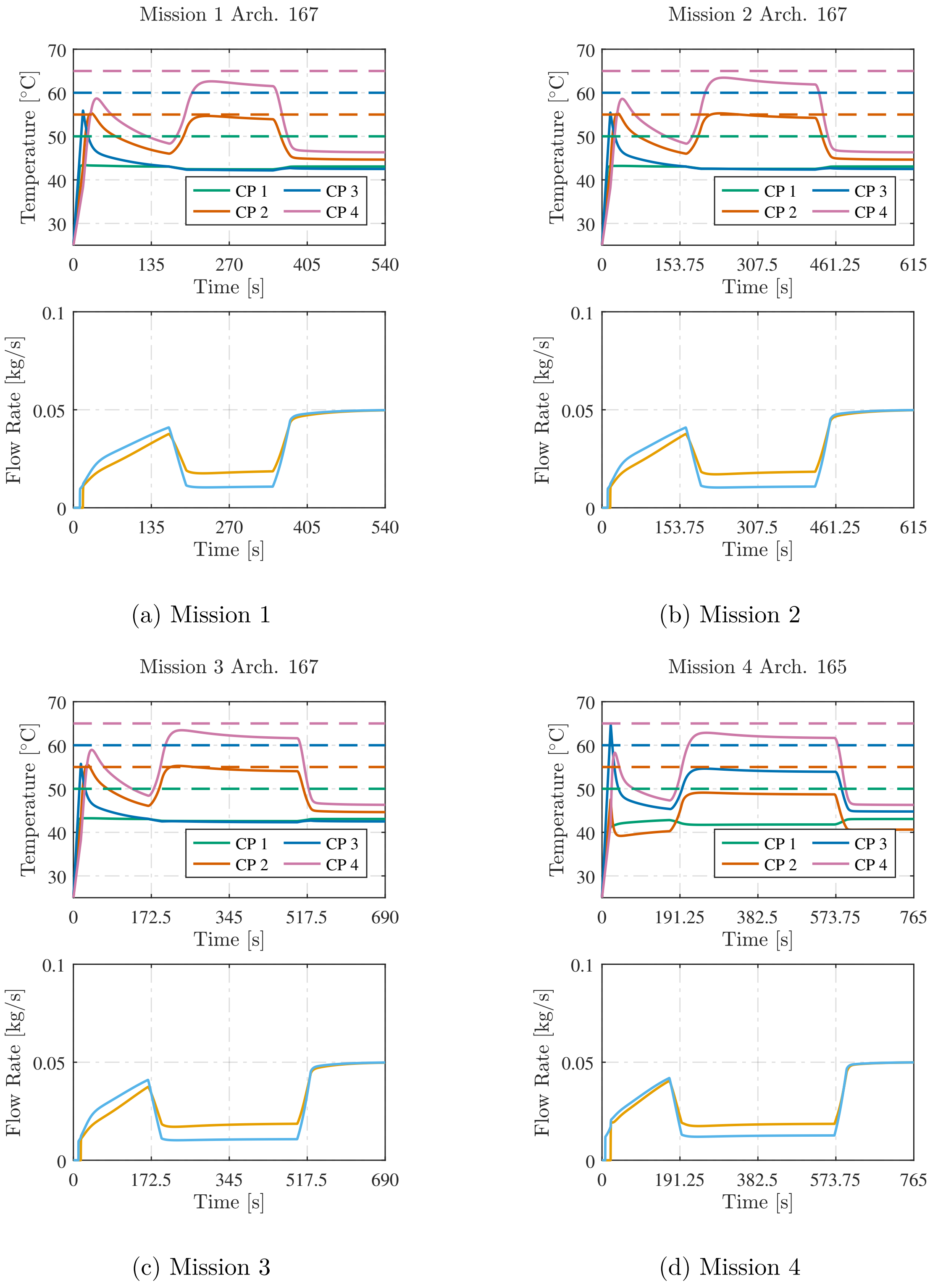


Figure D.2: The 21 highest performing TMS architectures represent by rooted trees.



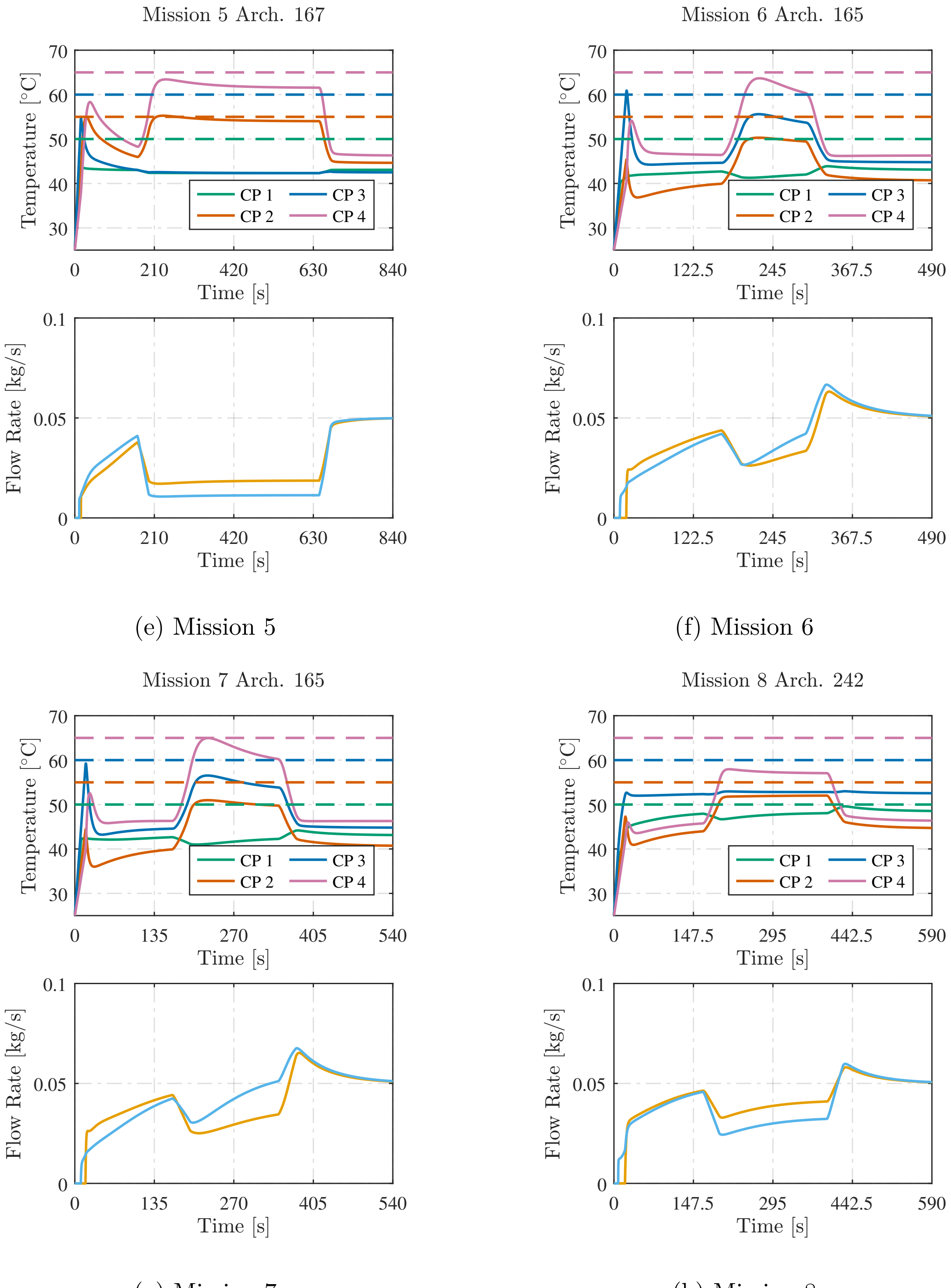
(a) Mission 1

(b) Mission 2

(c) Mission 3

(d) Mission 4

Figure D.3: The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



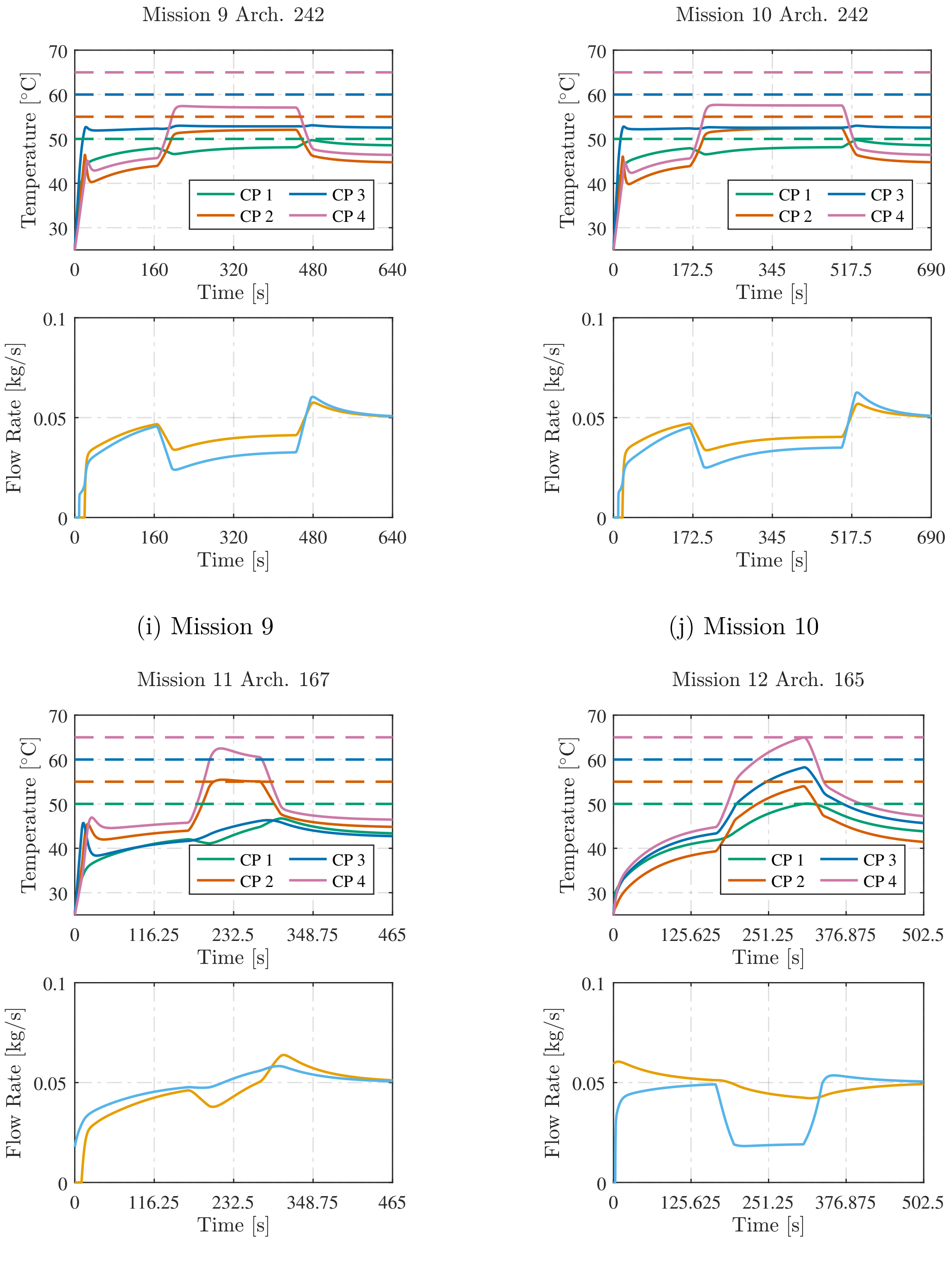
(e) Mission 5

(f) Mission 6

(g) Mission 7

(h) Mission 8

Figure D.3 (cont.): The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



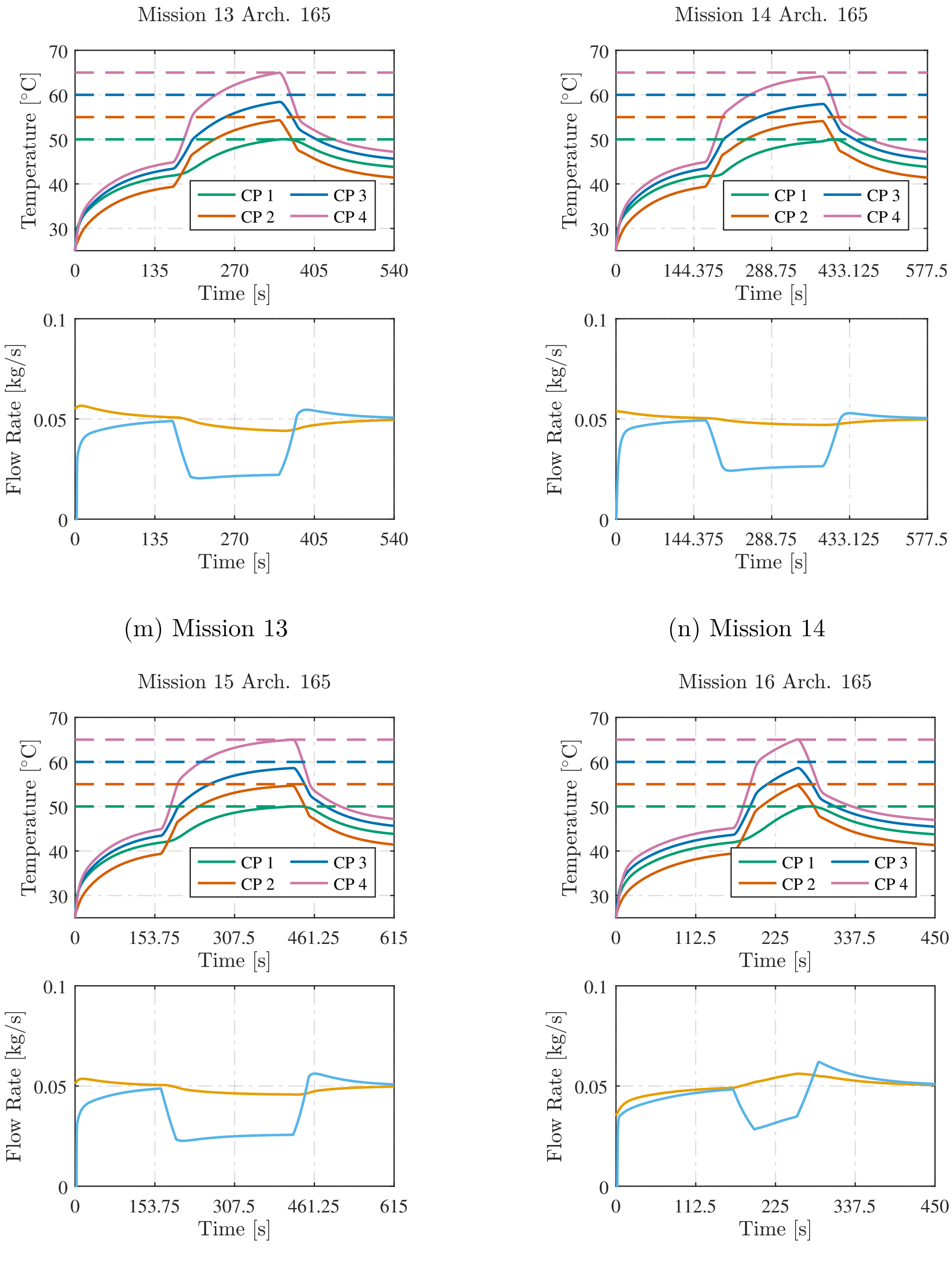
(i) Mission 9

(j) Mission 10

(k) Mission 11

(l) Mission 12

Figure D.3 (cont.): The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



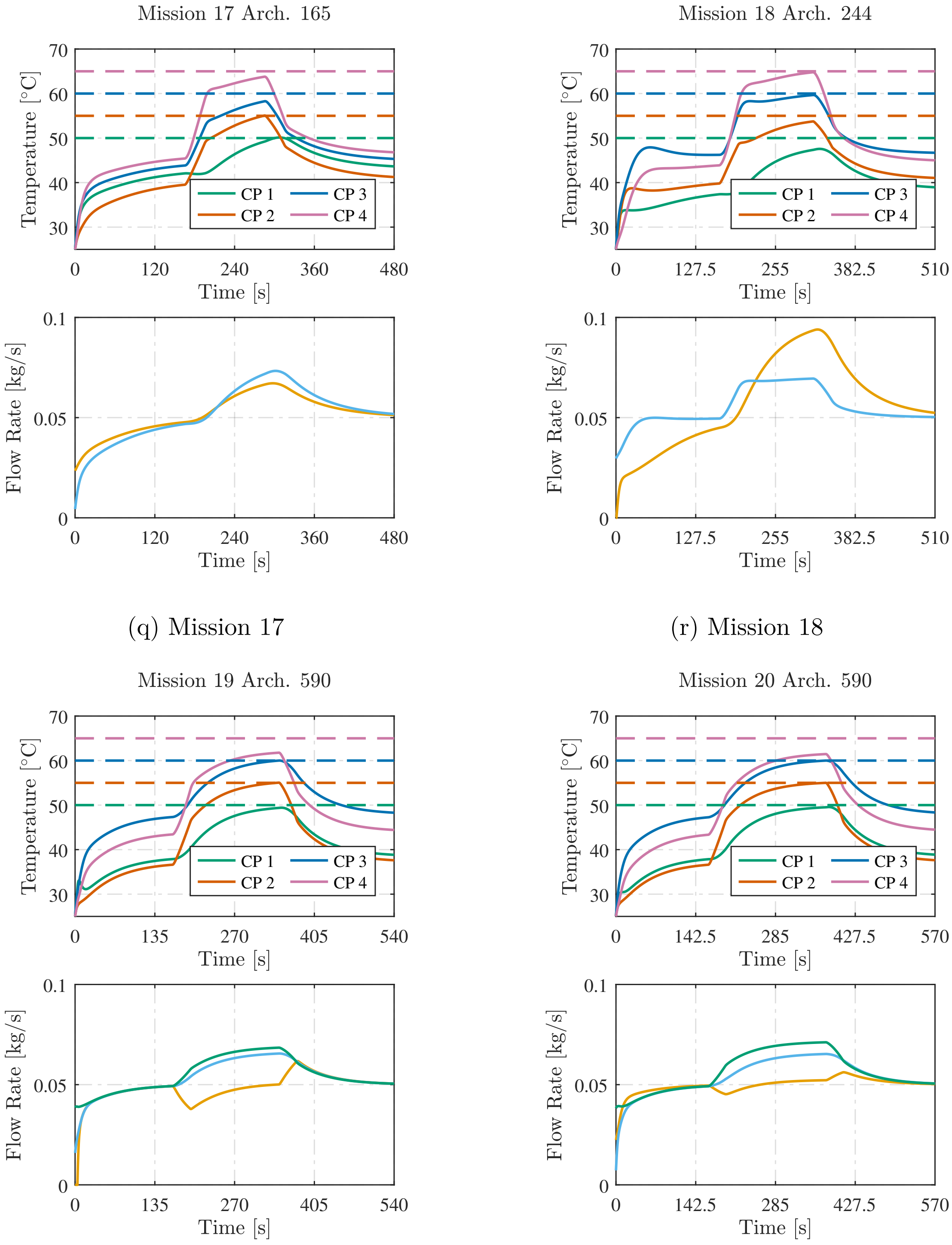
(m) Mission 13

(n) Mission 14

(o) Mission 15

(p) Mission 16

Figure D.3 (cont.): The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



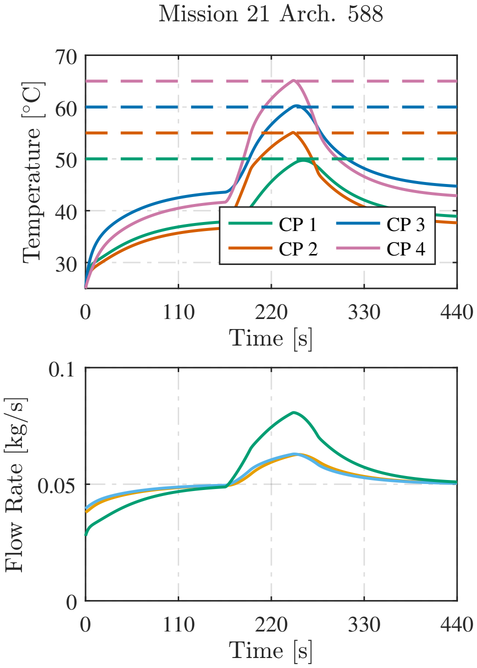
(q) Mission 17

(r) Mission 18

(s) Mission 19

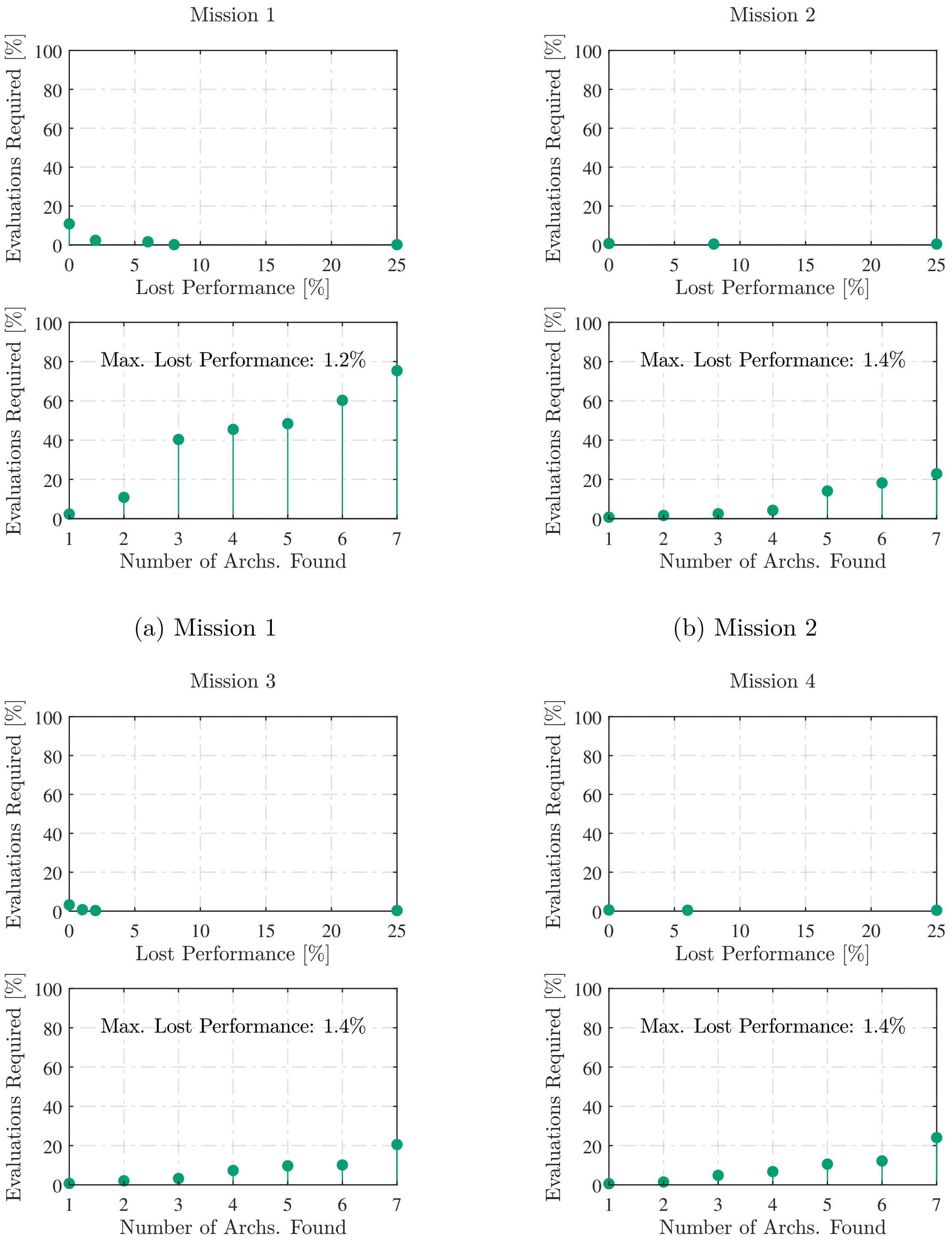
(t) Mission 20

Figure D.3 (cont.): The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



(u) Mission 21

Figure D.3 (cont.): The optimal TMS temperature state and flow rate input trajectories for each of the 21 feasible missions.



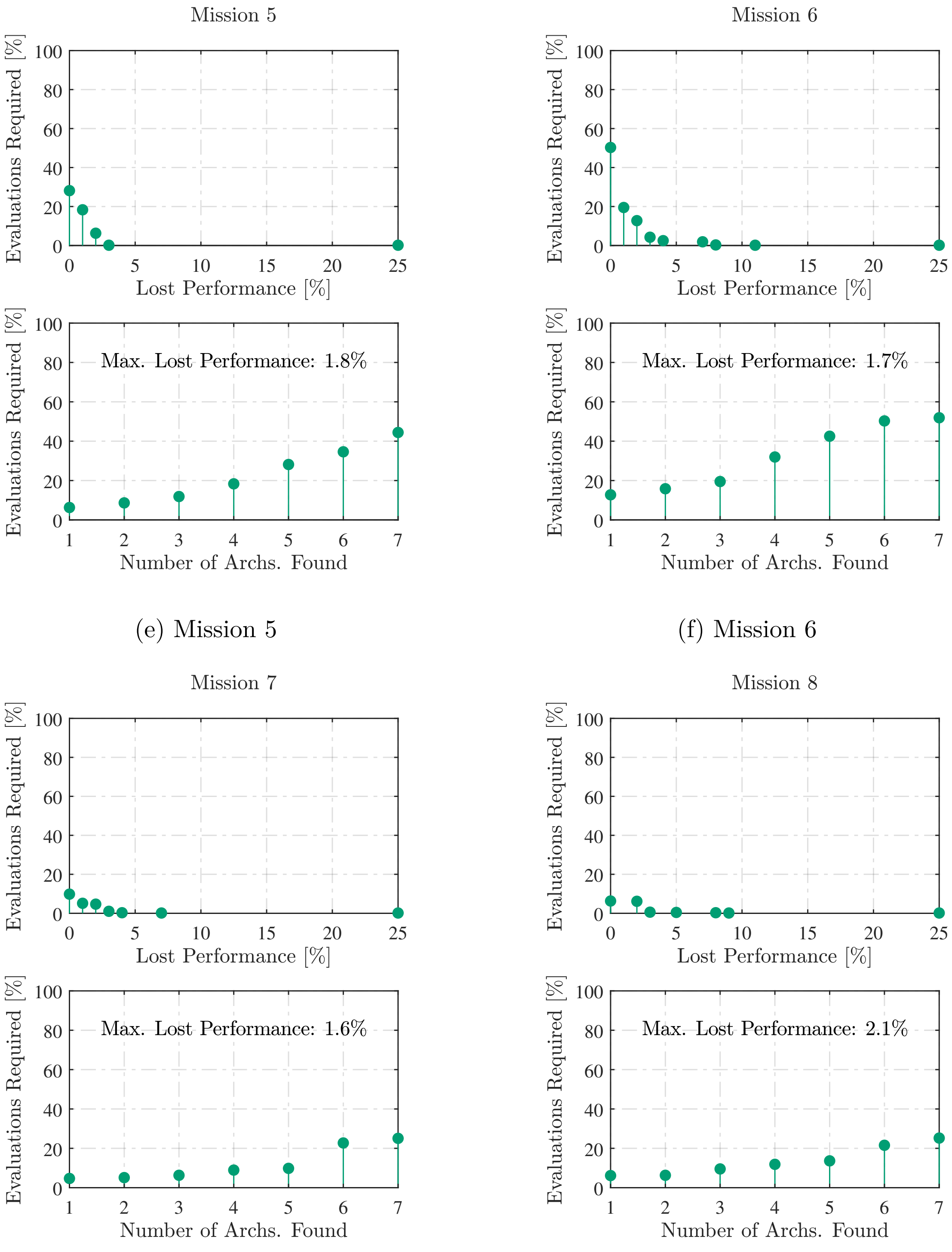
(a) Mission 1

(b) Mission 2

(c) Mission 3

(d) Mission 4

Figure D.4: The distance-sorted search optimization metrics for each mission.



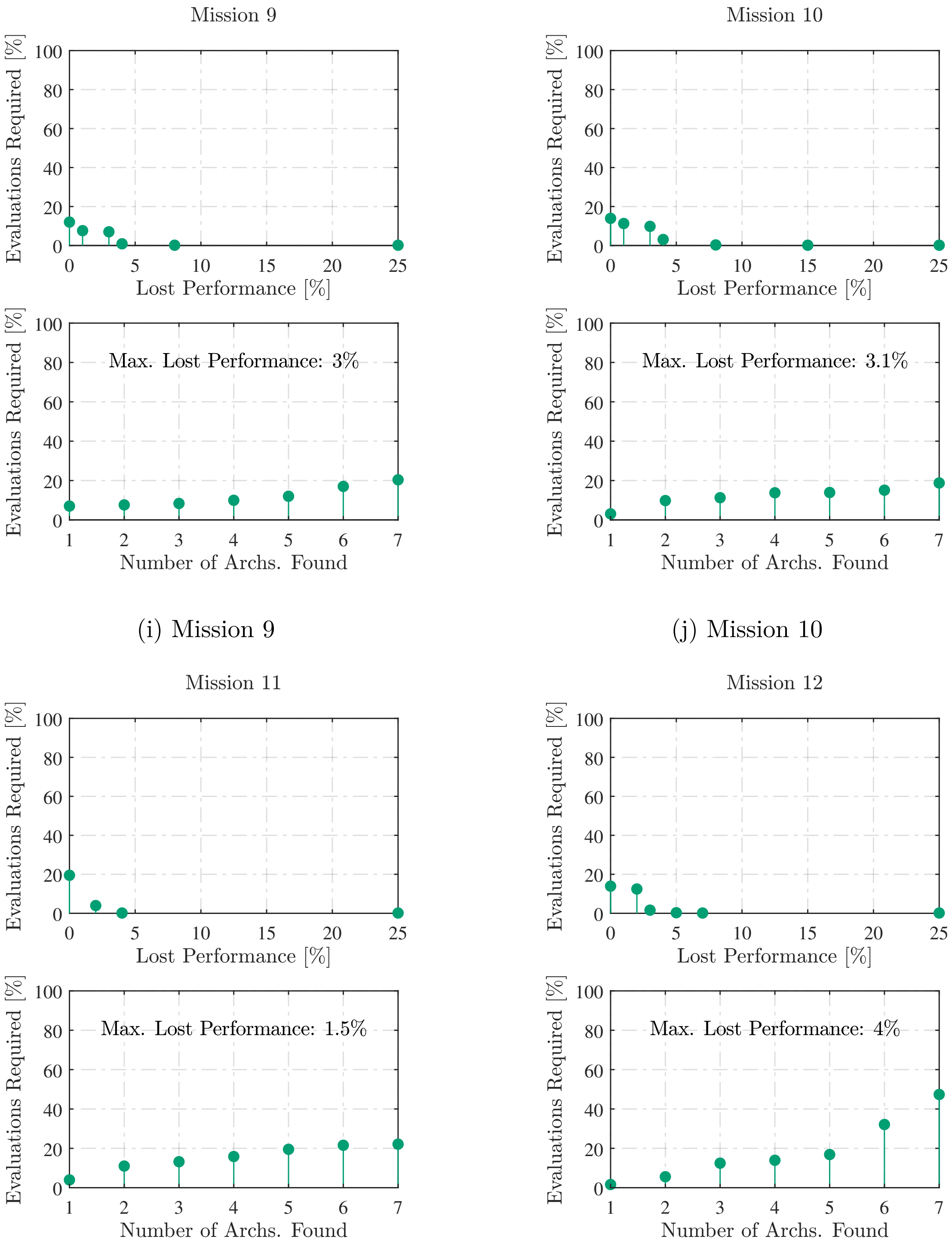
(e) Mission 5

(f) Mission 6

(g) Mission 7

(h) Mission 8

Figure D.4 (cont.): The distance-sorted search optimization metrics for each mission.



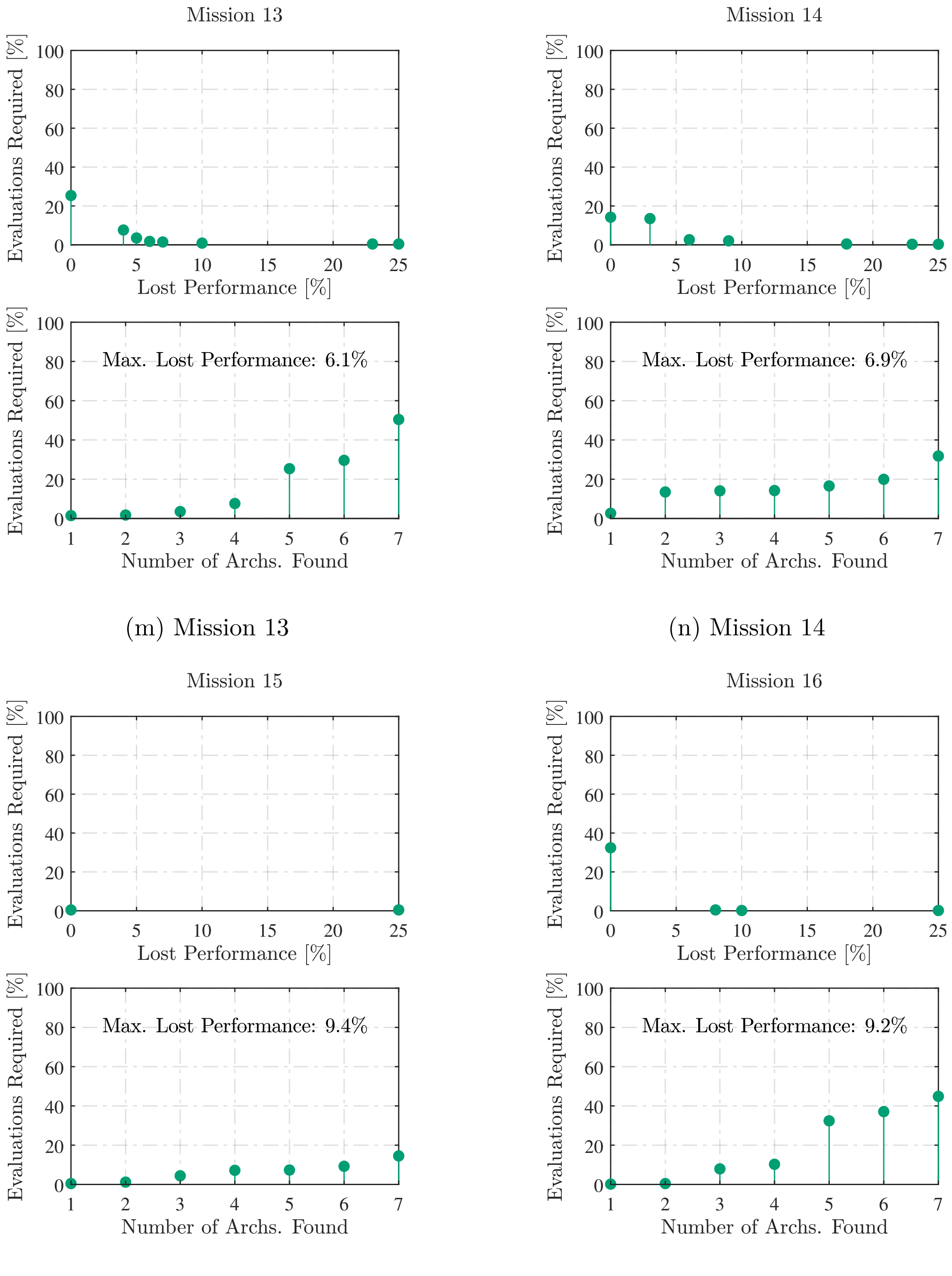
(i) Mission 9

(j) Mission 10

(k) Mission 11

(l) Mission 12

Figure D.4 (cont.): The distance-sorted search optimization metrics for each mission.



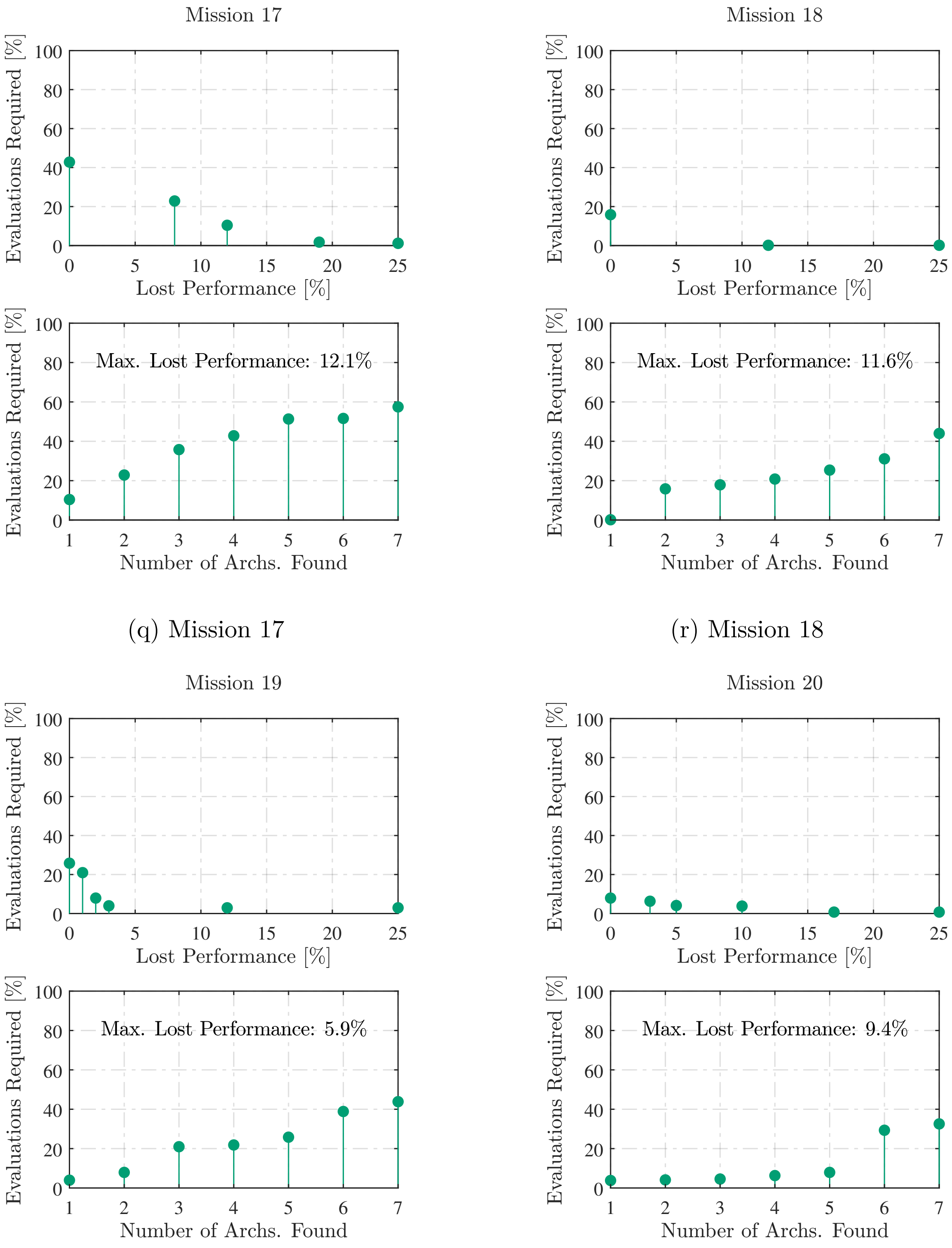
(m) Mission 13

(n) Mission 14

(o) Mission 15

(p) Mission 16

Figure D.4 (cont.): The distance-sorted search optimization metrics for each mission.



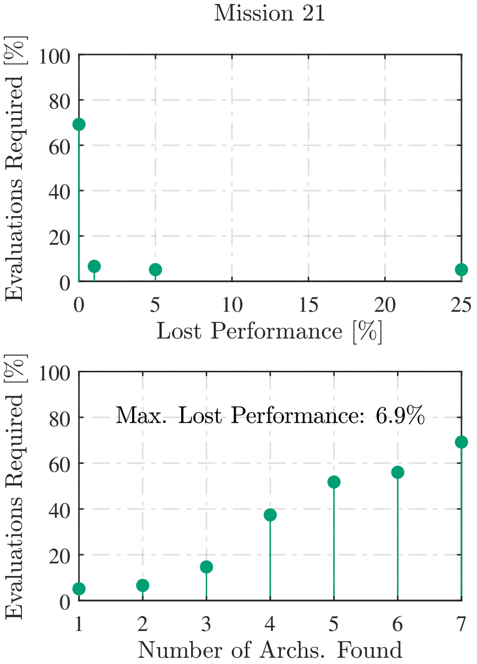
(q) Mission 17

(r) Mission 18

(s) Mission 19

(t) Mission 20

Figure D.4 (cont.): The distance-sorted search optimization metrics for each mission.



(u) Mission 21

Figure D.4 (cont.): The distance-sorted search optimization metrics for each mission.