RAPID, ASSURED PLANNING FOR SAFE OPERATION OF INTEGRATED POWER, PROPULSION, AND THERMAL SYSTEMS

BY

CARY LAIRD BUTLER

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, 2024

Urbana, Illinois

Doctoral Committee:

Professor Andrew Alleyne, Chair and Director of Research
Professor Carolyn Beck
Professor Kiruba Haran
Professor Nenad Miljkovic
Dr. Adam Parry, Air Force Research Laboratory

ABSTRACT

Use of electrified vehicles is on the rise in aerospace and marine applications, in part due to their provision of increased capability and efficiency compared to traditional combustion-powered vehicles. Leveraging their increased capabilities, electrified vehicles can perform missions with more complex and energy-intensive constraints such as operation with reduced emissions or noise during segments of the mission. At the same time, electrified vehicles' integrated power, propulsion, and thermal (IPPT) systems exhibit interactions between multiple energy domains (including electrical, thermal, and mechanical) that can lead to unsafe conditions if inadequately managed. To guarantee safe operation for these systems, energy and thermal management methods for these systems must address challenges including multi-domain, multi-timescale dynamics, high-dimensional, nonconvex planning problems, and mission specifications driven by uncertain, changing external conditions.

This dissertation addresses these challenges by introducing a two-stage approach to manage the multi-timescale dynamics. The first stage of this approach uses sampling-based planning methods in a novel application for energy and thermal management of IPPT systems. The second stage of this approach uses tracking control methods to robustly track reference trajectories from the first stage. Both of these stages leverage a graph-based modeling approach which is briefly surveyed.

The sampling-based methods manage slower dynamics and generate long-term mission plans. Notably, these methods can rapidly generate feasible mission plans in nonconvex feasible regions. A rapidly-exploring random trees (RRT) algorithm is presented which generates long-term mission plans using a reduced order model and a finite set of energy primitives, or predefined behaviors. Methods for re-planning online are presented to enable consideration of changing mission specifications. The presented planning algorithms ensure robustness to bounded error in the reduced order model due to imperfect tracking control.

Robust model predictive control (RMPC) is used to manage faster dynamics over a short time horizon and track the reference trajectory. The RMPC formulation used in this work considers a linearized, full order model to predict future system dynamics and ensures robustness to linearization error using error reachable sets. RMPC solves an optimization problem formulated to minimize tracking error in the reduced order model while satisfying the planner's error bound. Constraints applied to the optimization problem, including the error bound constraint, are tightened to account for the linearization error. The computation of reachable sets and numerical optimization are both performed online to determine optimal inputs to apply to the plant.

Three case studies are presented to demonstrate the application of this two-stage method to IPPT systems in marine and aerospace applications. These case studies include simulation results for a shipboard power system (SPS) and hybrid unmanned aerial vehicle (UAV) power, propulsion, and thermal system (PPTS), as well as experimental results for a hybrid UAV powertrain. In all three case studies, the two-stage approach yields rapid planning and assures constraint satisfaction even in the face of uncertain, time-varying constraints. In the SPS case study, scalability of the approach is demonstrated as the RRT algorithm reliably solves 20-dimensional, nonconvex planning problems in well under 1/1000th of the time taken to perform the mission. The hybrid UAV PPTS case study demonstrates the flexibility of this method to perform simultaneous energy and thermal management. Finally, the hybrid UAV powertrain case study demonstrates the real-time applicability of this approach to consider uncertain, time-varying constraints by validating the methods through implementation on an experimental testbed.

ACKNOWLEDGMENTS

Many individuals have contributed to my educational, personal, and research development and I would like to begin by extending my gratitude to these wonderful people.

To my advisor, Professor Andrew Alleyne, I am immensely grateful for your mentorship, guidance, and support. Thank you for the experiences and opportunities you have provided to boost my professional and career development. Thank you for teaching me how to tackle challenging problems in research and in life. Your dedication to students' success is unrivaled. I am grateful that your support and mentorship has remained strong through all the changes of the past five years. I look forward to continued collaboration in the years to come.

To the members of my doctoral committee, Professors Carolyn Beck, Kiruba Haran, Nenad Miljkovic, and Dr. Adam Parry, I am very grateful for your valuable suggestions and feedback.

To my family and friends, thank you for providing me with the foundation and fellowship to succeed. Your encouragement means the world to me. To my parents especially, thank you for teaching me the importance and rewards of hard work and for supporting my educational development.

To the students of the Alleyne Research Group (ARG), thank you for establishing a culture of collaboration, encouragement, support, and friendship. Thank you to Chris Aksland, Herschel Pangborn, Donald Docimo, and Spencer Igram for sharing your wisdom and support at various stages of my graduate school career. Thanks also to Pamela Tannous, Ashley Armstrong, and Oyuna Angatkina for making me feel welcome and valued from the very beginning of my time as an ARG student. Thank you to Mindy Wagenmaker, Reid Smith, Kayla Russell, Chris Urbanski, Frank Andujar Lugo, Phil Renkert, Kurt Kuipers, Dylan Charter, Enrico Sisti, Zahra Marvi, Mahsa Hemmat, and Christine Ohenzuwa for upholding the standard of success and friendship. I am grateful that my membership in ARG has provided me not only with an outstanding group of encouraging colleagues but with lifelong friendships.

I am grateful to the students, faculty, and staff of the Center of Power Optimization of Electro-Thermal Systems (POETS) for providing me with unparalleled opportunities for collaboration and professional development. I am especially grateful to Jodi Gritten for making my life easier with her friendship and administrative support. I am grateful to Profs. Andrew Alleyne, Alan Mantooth, Nenad Miljkovic, and Kiruba Haran, Dr. Jessica Perez, Owen Doyle, John Wierschem, Joe Muskin, and so many other current and former POETS members for making my experience as a student in POETS such a positive one. I would also like to extend my gratitude to Chris Aksland, Reid Smith, Sherry Yu, Kevin Colravy and others who made the experimental results of this dissertation possible.

To my collaborators at the Air Force Research Laboratory, I am incredibly grateful for your mentorship. Your feedback and suggestions have been instrumental to the development and fulfillment of the objectives of this dissertation and influential to my development as a researcher.

To my undergraduate advisors, thank you for encouraging me to pursue graduate school.

To Austin Butler, thank you for going on the graduate school journey with me and for reminding me at times to pause and enjoy the sights. You encourage me, reassure me, and always make me smile with your selfless joy. Your patience, support, and cooking skills have sustained me during many long and challenging days. I couldn’t have done this without you by my side. I am so grateful for your companionship in this journey, and I look forward to the journeys to come.

This research was supported by the National Science Foundation Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) with cooperative agreement EEC-1449548.

To my family, friends, and teachers

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Electrification is a major trend affecting all forms of mobility. Electrochemical sources are increasingly incorporated in powertrains that were historically powered by engines. This is apparently evidenced by the increasing numbers of pure electric and hybrid electric passenger vehicles on the road, as electrified vehicle sales have reached 16% of light-duty vehicle sales in the United States [1]. In another sense, electrification is evidenced by the inclusion of more electronic equipment onboard a vehicle to perform additional functions. This is an apparent trend in passenger cars, where electronics such as sensors, power converters, and electronic control units are increasingly common [2].

Multiple factors are driving this electrification trend. At the forefront, environmental concerns and associated regulatory pressures are forcing vehicle manufacturers to focus on hybrid or full electric options to reduce emissions [3], [4]. At the same time, these options provide multiple benefits to the driver, helping to increase adoption of electrified vehicles in the broader population. Benefits of electrified powertrains include improved performance, reduced maintenance, and reduced fuel expenses [5].

The electrification trend and its driving factors are not limited to ground vehicles like cars and trucks. Aircraft, ships, construction vehicles, and others are experiencing a similar electrification trend. Figure 1.1, based on data from [6]–[8], shows that this trend is expected to continue growing over the next decade. In the aerospace realm, electrified propulsion is garnering interest to meet emissions requirements during short, regional flights. Aircraft actuation and environmental control subsystems, which are traditionally hydraulically and pneumatically powered, are now increasingly driven by electricity [9]. Marine vehicles are following a similar path to electrification. Integrated power systems, including electrical and combustion-powered sources, are being considered to enable reduced emissions in ports. Over the past century, hotel electrical loads, including fuel pumps, cooling systems, and cargo handling, have increased to account for upwards of 20% of the installed power [10].

To enable the shift towards electrified transportation, engineers have made substantial progress over the past few decades to enable these vehicles to operate safely and efficiently. Energy management is a key consideration in the operation of an electrified vehicle. Depending on the particular vehicle platform, the energy management method may be subject to a variety of challenges.

1.1. ENERGY MANAGEMENT CHALLENGES FOR ELECTRIFIED VEHICLES

Complex physical interactions linking multiple energy domains, such as electrical, thermal, and mechanical, provide challenges for energy management of any electrified vehicle. While electro-mechanical interactions such as electromotive force make electrified mobility possible, tight integration of electrical, mechanical, and thermal systems leads to other multi-domain

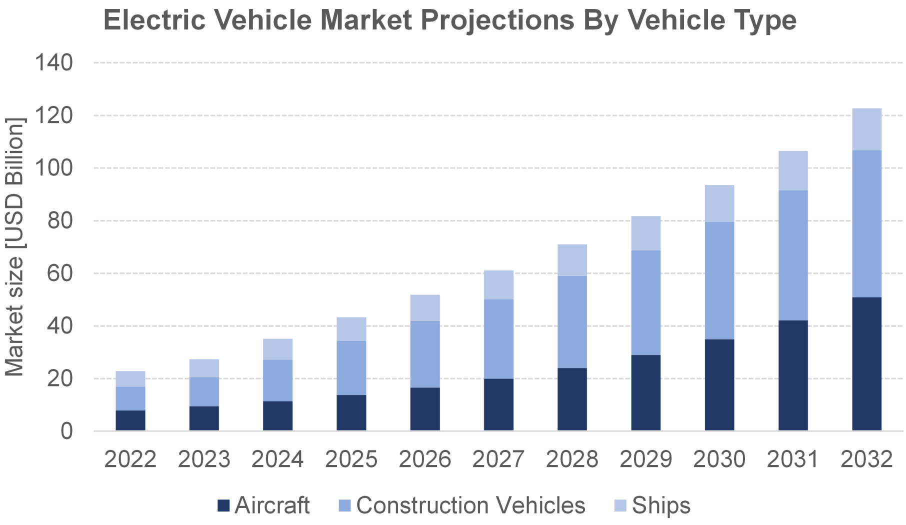


Figure 1.1: Electric vehicle market projections by vehicle type. Compiled from data from [6]–[8].

interactions that may negatively impact system behavior. Because of this multi-domain coupling, electrified vehicle systems are referred to in this dissertation as integrated power, propulsion, and thermal systems (IPPTS). Notably, electro-thermal interactions heavily influence electronic components' performance and reliability. Electrical inefficiencies lead to heat generation causing component temperatures to rise. Yet electronics often become more inefficient as temperatures rise [11], [12], which in turn leads to increased heat generation. Without proper management of the generated heat, compounding thermal stresses can negatively impact component- and system-level performance. For example, due to their composition of heterogeneous materials, power converters are subject to thermomechanical stresses which are exacerbated by heating cycles [13]. In fact, thermal stresses such as these have been identified as the leading source of electronics failures in defense-related applications [14]. Likewise, unsafe operating temperatures are some of the most prominent stressors for battery aging, which can arise in the form of capacity reduction or power fade [11].

Additionally, due to their mobile nature, vehicles are inherently subject to uncertainty. A variety of unknown factors influence a vehicle's performance, including weather, traffic, and the operator’s inputs. These factors can have a significant impact on the vehicle's energy systems. Driving into a headwind can reduce range, as can rapid, repeated acceleration and deceleration. While predictions and estimates can be made regarding these conditions, their uncertain nature makes it impossible for an energy management method to have perfect knowledge of the external conditions acting on the vehicle. Energy management methods must be robust to this uncertainty and must quickly react to changes in external conditions to safely operate a vehicle.

While the challenges mentioned above and others have been addressed to enable operation of electrified automobiles, electrification of off-highway vehicles such as aircraft and ships has been limited because these vehicles face unique challenges compared to automobiles. One of the challenges these vehicles face is their increased complexity in terms of the number of subsystems comprising their power systems. Supporting high propulsion power demands and specialized equipment often requires multiple power sources. Redundancy requirements, particularly stringent for aircraft, necessitate backup power sources to ensure safe operation in case of failures [15]. In addition to the complexity introduced by multiple power sources, these vehicles also have a variety of subsystems and equipment to support. In addition to propulsion power demands, environmental control systems, and actuation systems, vehicles may need to support specialized equipment. Military vehicles may need to support sensing devices, weaponry, or medical equipment. Large passenger vehicles like cruise ships must support water lines, cooking and food storage equipment, and other amenities to enhance passenger comfort. Considering the multiple energy sources and sinks, energy management for this class of vehicles is challenging.

An emerging challenge for electrified aircraft and ships is the need to fulfill specialized and energy intensive tasks or missions. Mission-specific requirements may place constraints on the operation of certain components or subsystems, where these constraints may be active only during particular segments of the task or mission. A motivating example of these mission-specific requirements is illustrated by Figure 1.2, which depicts the flight path of a hybrid electric UAV. In this example, the UAV must travel through urban environments in which noise restrictions are in place. These noise restrictions, or “quiet zones,” forbid engine operation at particular locations. Additionally, the UAV is tasked with operating a high-powered sensor in a given region. Missions like this introduce additional coupling between the energy subsystems, as these location-specific constraints couple the task support systems to the propulsion systems. Feasible regions yielded by

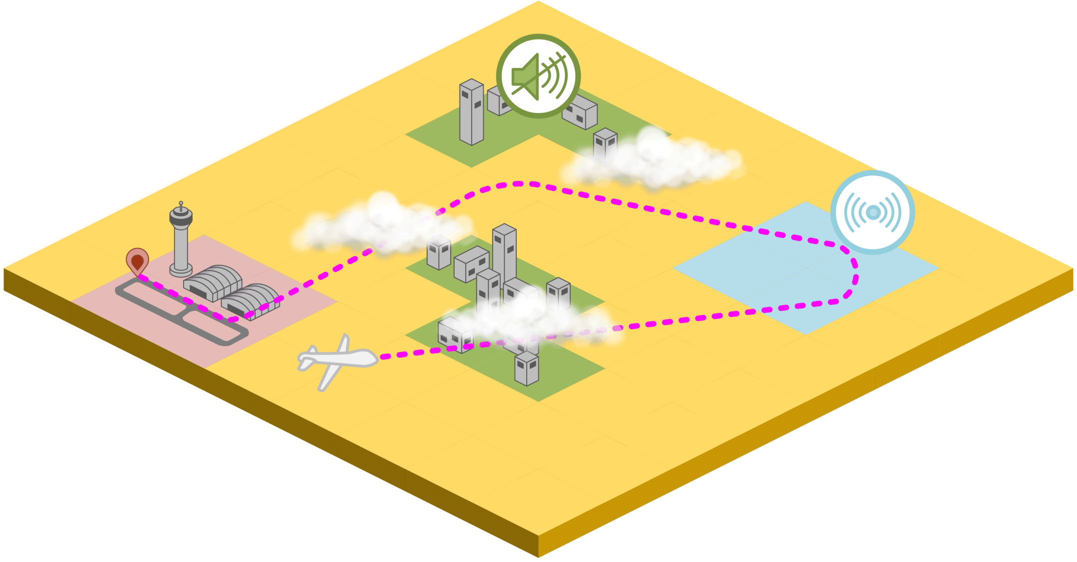


Figure 1.2: Example mission-specific constraints for a hybrid electric UAV may include quiet zones (green) and sensing zones (blue).

taking the complement of these task-specific constraints can exhibit nonconvexity, which impedes the use of many optimization-based energy management methods.

1.2. EXISTING METHODS

The existing methods for operation of IPPT vehicle systems focus heavily on energy management of the powertrain. Widely studied for automotive applications, conventional methods for energy management are classified as rule-based, optimization-based, and learning-based methods [16]–[19]. A brief overview of each class of methods, including examples from aerospace and marine applications, is given below.

The broad category of rule-based methods encompasses those which do not rely on *a priori* knowledge of the driving cycle or future conditions, such as logic-based, setpoint-regulating, and lookup-table-based methods. Examples of rule-based methods for hybrid electric aircraft energy management include battery charge sustaining and depleting strategies [20] and state-machine methods [21]. Similarly, some ship energy management methods include state machines and sets of rules used to select operating points based on the current task [22], [23]. Making decisions based on the current state of the system, these methods are inherently robust to uncertainty and very computationally efficient. While these methods can be designed to bias the system towards desired or safe operating points, their lack of predictive planning capabilities prevents rule-based methods from guaranteeing satisfaction of complex mission-specific constraints such as those discussed above.

Optimization-based methods employ mathematical models and additional information to improve high-level metrics such as fuel consumption or component health. This class of methods can be divided into offline and online optimization methods. Offline optimization methods assume perfect knowledge of the system model, mission specifications, and environmental conditions to determine the optimal control policy over an entire mission. In the literature, dynamic programming has been used to determine optimal control policies for electrified aircraft [24], [25] and ships [26]. These methods have the advantage of theoretical optimality guarantees and some formulations can consider nonconvex constraints. Yet their implementation for real-time vehicle operation is intractable due to challenges including the lack of robustness to uncertainty and prohibitively high computational cost, especially for high-dimensional problems. Vehicles operate in changing environmental conditions requiring decisions to be made in short timeframes, which is difficult to achieve with computationally intensive optimal control methods. In contrast, online optimization methods employ computationally efficient numerical optimization strategies to approximate the optimal control policy. A popular approach is model predictive control (MPC), which is used to optimize energy management decisions online over a finite time horizon. Using MPC, researchers have demonstrated reduced fuel consumption and further improvements over rule-based methods for energy management of aircraft [27], [28] and ships [29]. To retain fast computation for online implementation, MPC formulations are often linear or quadratic optimization problems for which efficient numerical solvers are available. These formulations cannot accommodate nonconvex constraints such as the mission-specific constraints considered in this work.

Learning-based methods pre-process large amounts of data to learn optimal operation strategies that can be deployed online. Reinforcement learning approaches have been shown to approximate the optimal control policies for energy management of aircraft [30] and ships [31]. Yet without predictive planning capabilities, these learned policies do not guarantee satisfaction of mission-specific constraints.

Of the conventional methods, only the optimization-based approaches can directly incorporate constraints to guarantee safe operation. Most optimization-based energy management problems consider only bound constraints on state variables such as battery state of charge, battery discharge rate, and engine power, such that the feasible regions are convex interval sets. In contrast, the mission-specific constraints considered in this work yield nonconvex feasible regions. In these regions, the gradient-based optimization approaches commonly used for online optimization may not converge or may converge to a local optimum. Gradient-free algorithms are more likely to converge to an optimum in nonconvex feasible regions, but their convergence tends to be much slower than their gradient-based counterparts. Some researchers have shown near optimal energy management for hybrid electric aircraft in the presence of quiet zone constraints by discretizing the feasible space and using dynamic programming [32] or graph search methods [33], [34]. Yet the computational efficiency of these methods is dependent on the discretization resolution and the pre-processing required to discretize the map, and the curse of dimensionality may lead to significant increases in computational demand for more complex systems. As such, these methods are ill-suited for more complex powertrains and cases in which the mission-specific constraints change over time.

Energy management decisions are also influenced by thermal dynamics, so control strategies for thermal management are important to maintain safe operation. For aircraft, thermal management including control is becoming an increasingly important consideration. Conventional control strategies for thermal management systems that employ rule-based or setpoint-tracking methods neglect coupling among subsystems. As additional electronic equipment and mission systems contribute to significant increases in heat generation on aircraft, these conventional methods are becoming insufficient to maintain thermal constraints and are contributing to reduced thermal endurance [35]–[37]. Researchers have demonstrated that, when considering the system-level thermal management problem, optimization-based methods like MPC can outperform the conventional approaches. MPC frameworks have been used, for example in [38]–[40], to demonstrate reduced temperature constraint violations and energy consumption compared to conventional methods. However, gradient-based optimization techniques used in MPC formulations remain incompatible with nonconvex feasible regions such as those considered in this work.

1.3. DISSERTATION OBJECTIVES AND SCOPE

The literature lacks methods for energy and thermal management that can guarantee satisfaction of complex, time-varying constraints. To address this need, energy and thermal management methods must achieve several desired characteristics:

- Rapid planning: able to quickly react to changing mission specifications
- Assured operation: guaranteed to satisfy mission specifications

- Agnostic to constraint complexity: able to accommodate nonconvex feasible regions with complex geometric representations
- Multi-timescale: capable of managing long- and short-term dynamics
- Scalable: can be applied to highly complex systems with multiple energy sources, sinks, and storage elements.

In this work, a two-stage framework for energy and thermal management is introduced which combines sampling-based planning methods and model predictive control (MPC) to provide these qualities. To achieve and demonstrate this, the objectives of this research are as follows.

1. Develop planning-oriented and control-oriented models of multi-domain systems to enable the implementation of control and planning methods;
2. Develop sampling-based methods for rapid, assured planning of IPPT systems;
3. Extend sampling-based planning methods to enable online re-planning;
4. Develop predictive control methods for guaranteed tracking performance;
5. Demonstrate, in simulation and in experiment, the two-stage framework for rapid, assured planning and online re-planning.

To demonstrate the fulfillment of these objectives, the dissertation is organized as follows. Chapter 2 provides an overview of the two-stage framework for rapid, assured planning. Chapter 3 discusses the modeling techniques used for the implementation of planning and control methods. Chapter 4 discusses sampling-based planning methods used for long-term planning of IPPT systems. Chapter 5 discusses robust model predictive control methods for guaranteed tracking of planned trajectories. Chapter 6 provides simulation and experimental case studies to demonstrate the methods. Chapter 7 concludes the dissertation.

CHAPTER 2: TWO-STAGE FRAMEWORK FOR PLANNING AND CONTROL

The IPPT systems comprising electrified vehicles consist of multi-domain components with a wide range of dynamic timescales. For example, thermal dynamics evolve on the order of minutes, while electrical dynamic timescales are often smaller than one second. To manage fast dynamics, control strategies must update frequently to prevent the system from quickly moving into an unsafe region. To manage slow dynamics, controllers must perform long-term predictive forecasting using mathematical models. Performing long-term predictions with small update rates or timesteps is computationally intensive, but is the standard approach with centralized control architectures. To mitigate computational complexity, centralized control architectures may employ simplified mathematical models, but this incurs model errors that can make the control strategy less robust. Decentralized control architectures consider decoupled dynamics to mitigate computational complexity, but neglecting coupling can lead to excessive conservatism or poor performance.

In contrast, hierarchical control approaches use timescale separation to decompose control problems. This allows for higher-level controllers to perform long-term planning for slower dynamic systems to achieve high-level objectives, while lower-level controllers manage faster dynamics to achieve the objectives of the higher levels and potentially additional objectives [41]. The benefits of hierarchical control architectures for energy and thermal management of IPPT systems in aircraft applications have been established in the literature. In [38], a hierarchical MPC approach was used for an aircraft fuel thermal management system, which included slow thermal dynamics (e.g., cold plate wall temperature) and fast hydraulic dynamics (e.g., fluid pressure). Compared to decentralized and centralized approaches, the hierarchical approach demonstrated reduced constraint violations. In [27], a hierarchical control approach was used for energy management of a hybrid electric UAV powertrain, simultaneously managing slow dynamics like battery state of charge (SOC) and faster dynamics like propeller speed. The hierarchical control was adept at managing the multi-timescale electro-mechanical dynamics and provided improvements in constraint enforcement and fuel consumption compared to a baseline controller. In [42], hierarchical MPC was used to solve the combined problem of energy and thermal management for a hybrid electric UAV. By managing the multi-timescale electro-thermal-mechanical dynamics, the hierarchy provided improved reference tracking, improved thermal performance, reduced fuel consumption, and reduced constraint violations compared to a baseline controller. Similar studies have demonstrated that hierarchical control is also beneficial for IPPT systems in marine applications. In [43], a hierarchical control strategy is used to simultaneously manage the power split and the power quality for a DC-distribution-based shipboard power system including multiple generators and energy storage systems. The hierarchical method demonstrates high fuel efficiency even in variable loading conditions such as dynamic positioning. In [44], a hierarchical method is used to plan operation of a hybrid power system for an all-electric ship. This method uses static optimization for long-term planning and model predictive control to reject disturbances for offset-free tracking of the planned trajectories, demonstrating high computational efficiency in long-term planning. Another hierarchical control strategy for hybrid ship power systems is found in [45], where dynamic programming and MPC are used for high level control, demonstrating computational benefits.

While these studies and others have shown the benefits of hierarchical control methods for energy and thermal management of IPPT systems, the existing methods have not demonstrated the ability to perform rapid, long-term planning to accommodate nonconvex, time-varying feasible regions. Our approach differs from these methods in the use of sampling-based planning at the top level to address this need. An overview of this approach is provided in this chapter.

2.1. OVERVIEW

Following the successes of other hierarchical control frameworks, this framework addresses the challenges of multi-timescale dynamics by decomposing system operation decisions into long-term mission plans and short-term tracking of plans. The framework is depicted in Figure 2.1. Mission specifications are defined based on the external conditions. These mission specifications include the constraints, both operational and mission-specific, initial condition, and goal region. The high-level mission specifications are communicated to the sampling-based mission planner. The mission planner is tasked with finding a feasible, long-term mission plan that starts at the initial condition, lands in the goal region, and satisfies constraints along the way. The mission plan is communicated to the robust tracking controller, which actuates the plant to achieve the mission plan. The goal of the tracking controller is to track references from the planner with a guaranteed error bound. In this way, the mission planner performs long-term planning of slow dynamics and the tracking controller manages faster dynamics at a faster update rate. Problem statements for these two stages are provided in the following sections.

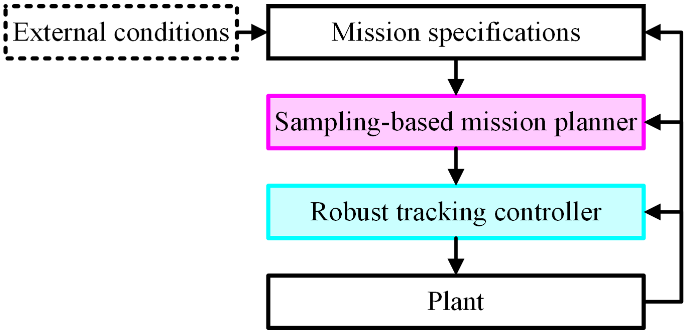


Figure 2.1: Overview of two-stage planning and control framework.

2.2. MISSION PLANNING

For the scope of long-term planning, energy and thermal management problems with mission-specific constraints can be posed as planning problems with differential constraints subject to uncertainty. A generic problem formulation, similar to the problems considered in [46], is provided below.

Problem (2.1): Mission planning under differential constraints and uncertainty

Let $\bar{X}$ define a reduced-order, nominal model of energy and power states of an IPPT system, with a nominal state transition equation $\dot{\bar{X}} = f(\bar{X}, A)$ for some action $A \in \mathcal{A}$. Let $\mathcal{X}$ define all possible energy and power states, and $\mathcal{X}_{con} \subset \mathcal{X}$ define forbidden energy and power states (i.e., constraints) such that the feasible region is given by $\mathcal{X} \setminus \mathcal{X}_{con}$. Let X_I define the system's initial condition and $\mathcal{X}_G$ define the goal region (i.e., terminal region). Assume that true energy and power states are given by $X = g(x,u,d)$, where x, u, and d are states, inputs, and disturbances of a full-order model. Assume that $X \in \bar{X} + \mathcal{X}_{\mathcal{W}}$ for all time, where $\mathcal{X}_{\mathcal{W}}$ is a closed, bounded set containing the origin that can be parameterized by another closed, bounded set $\mathcal{W}$ containing the origin. The IPPT planning problem is then to compute an action trajectory $A : t \rightarrow \mathcal{A}$ for which the true state trajectory X satisfies the following: $X(0) = X_I$; there exists some time T such that $X(T) \in \mathcal{X}_G$; and $X(t) \in \mathcal{X} \setminus \mathcal{X}_{con}$ for all $0 \le t \le T$.

In short, Problem (2.1) seeks to find a feasible trajectory (i.e., a trajectory that does not violate constraints) starting and ending at a defined initial point and goal region. Focusing on safe operation, this problem sacrifices the optimality of planned trajectories in order to quickly find a feasible trajectory that guarantees that no constraint violations will occur. Making such a guarantee is dependent on the assumption included in the problem statement that the true energy system satisfies the following condition.

$$X \in \overline{X} + \mathcal{X}_{\mathcal{W}} \tag{2.2}$$

Ensuring the satisfaction of this condition is the primary task of the robust tracking controller, which will be overviewed in the next section. Assumption (2.2) simplifies the planning problem, allowing the planner to use a reduced-order nominal model $\overline{X}$ to efficiently perform long-term predictions, rather than the full-order model defined by $X = g(x,u,d)$. In particular, the planner uses the set $\mathcal{X}_{\mathcal{W}}$, which parameterizes error in the reduced-order model, to introduce conservatism to guarantee that the true system will not violate constraints even in the worst case of tracking error. Note that the nominal model used by the planner considers actions $A \in \mathcal{A}$ which are distinguished from the inputs u applied to the true system. These actions can represent abstracted high-level behaviors, or energy primitives, which can be optimized or learned [47]. In this work, a library of primitives is used, where each primitive is a pre-defined action trajectory with a fixed time duration. The planner then constructs plans as sequences of primitives.

A key feature of this approach is the use of sampling-based methods at the mission planning level of the hierarchy to solve Problem (2.1). Compared to optimization-based methods, sampling-based methods offer increased computational efficiency and the ability to consider more complex constraints. This enables rapid mission planning in nonconvex feasible regions, allowing the planner to quickly react to changing external conditions to maintain feasibility. In this work, online re-planning is performed periodically throughout the mission. Because plans consist of sequences of finite-duration primitives, re-planning can take place while a primitive is being executed before transitioning to the next primitive. As long as the computation time for re-planning remains within a defined window, which is shorter than the time duration of the primitive, the re-planned mission can be updated and sent to the tracking controller with no loss of continuity.

A notable drawback of these sampling-based methods is that they achieve low computational expense by sacrificing optimality. While there exist sampling-based methods that asymptotically converge to optimal trajectories, many sampling-based methods find only a feasible trajectory. To guarantee safety despite the time-varying, uncertain mission specifications considered in this work, sampling-based methods are preferred. Waiving optimality to enable the rapid solution of feasible trajectories is a key aspect of the current dissertation and a differentiator from previous work in the field of energy management.

2.3. TRACKING CONTROL

In the second stage of this framework, summarized by the problem formulation below, a tracking control method is used to actuate the system to track reference trajectories from the planner.

Problem (2.3): Robust tracking of planned trajectory

Let the full-order state model $\dot{x} = f(x,d,u)$ correspond to the reduced-order model $X = g(x,d,u)$, and let $\overline{X}$ be a nominal trajectory generated by the solution of problem (2.1) . Find an input trajectory u to apply to the plant such that the actual trajectory is guaranteed to track the nominal trajectory within the bound prescribed by condition (2.2).

While the problem formulation of (2.3) is fairly generic, methods that are capable of solving this problem are limited due to the complexity of IPPT systems. In some cases, it might be possible to use formal methods (e.g., Lyapunov analysis) to analytically bound the error between the nominal and actual trajectory. In many cases, however, complex state dynamics and model error can make such an analytical bound intractable or inaccurate. In such cases, numerical optimization strategies can be used, and the set $\mathcal{W}$ which parameterizes the error bound of (2.2) can be treated as an adjustable parameter of the controller. For example, robust model predictive control (RMPC) provides a tractable alternative. RMPC numerically solves optimization problems online subject to constraints such as (2.2), where constraints are tightened to ensure robustness to model error.

A key consideration for the tracking controller is the ability to manage fast dynamics of the system. This requires computationally efficient control methods that can quickly compute inputs to apply to the plant. For RMPC, which predicts future behavior of the system over a finite time horizon, the computational requirement can restrict the length of the horizon. In this two-stage approach, long-term planning is done by the mission planner so this is not too restrictive. However, the controller horizon length affects the re-planning computation time window. To prevent changing reference trajectories between subsequent controller calls, the re-planning computation time should be one controller horizon less than the time duration of the primitive.

2.4. SUMMARY

In this chapter, an overview of the two-stage framework for planning and control of IPPT systems is provided. After the brief overview in Section 2.1, Section 2.2 introduced the mission planning problem formulation. Section 2.3 discussed the requirements of the tracking control method. Both the mission planning and tracking control stages rely on models of the system. To develop these models, Chapter 3 will discuss the use of a graph-based modeling framework.

CHAPTER 3: MODELING

Dynamical models of IPPT systems are vital to guarantee safe operation. Models are used to predict long-term behavior, optimize control decisions, and simulate system response to a given control decision. The computational requirements of simulating a mathematical model are highly influenced by the level of complexity, which is related to model elements such as the number of states and amount of nonlinearity. Selecting the level of complexity is a key decision in the development of a mathematical model, as increased complexity increases the computational demand of evaluating the model. For instance, a linear model with few states is suitable to predict overall system behavior with low computational expense yet is inaccurate for simulating nonlinear systems. On the other hand, a nonlinear model with many states may prove too computationally expensive for long-term prediction, but is likely more accurate, allowing the user to understand, verify, and optimize the immediate effects of control decisions.

At any level of complexity, the development of a mathematical model for IPPT systems can be challenging. A primary challenge is the inclusion of multiple, disparate energy domains. Understanding the interactions between energy domains such as electrical, thermal, and mechanical requires a multidisciplinary view of the physical mechanisms of energy transfer and storage. Yet the standard modeling techniques of one physical discipline often do not meaningfully translate to physical quantities of another discipline. Another challenge of modeling IPPT systems is that across disparate energy domains, dynamic timescales can vary significantly. In an electric vehicle, for example, it may take less than a second for a motor’s speed to increase. But the effect of this increased speed on the battery’s state of charge will surface on a much longer time scale.

To address these challenges, several desired features of mathematical modeling frameworks can be defined. First, the ability to represent varying levels of physical fidelity is beneficial for a multi-purpose modeling tool. Modularity is another helpful characteristic to enable the user to readily add, remove, or swap components. Additionally, it is helpful for a modeling framework to employ common analysis techniques across energy domains. Generalizability across domains in this way eases the tasks of developing model-based approaches for planning and control of IPPT systems.

A variety of toolboxes exist for modeling IPPT systems. Some of the major toolboxes that support modeling of these systems include MathWorks's Simscape [48] and the open source Modelica Standard Library [49]. These widely used multi-purpose modeling toolboxes have proven useful for modeling a variety of components and systems of multiple domains, including electrical, mechanical, fluidic, and thermal. As such, these toolboxes are widely used in automotive, aircraft, and power systems industries [50]. These toolboxes employ modular component models, making it easy to construct system models consisting of multiple interconnected components. Yet the component models within these toolboxes have predefined levels of fidelity, which does not allow the user to adjust fidelity to achieve different modeling objectives. Further, as numerical simulation tools these toolboxes do not allow the user to extract the underlying systems of dynamic equations. Some researchers and companies working in this realm have developed their own modeling toolboxes. Examples include the ATTMOSphere [51] and thermal management system (TMS) [52] modeling toolboxes for thermal systems and the PowerFlow toolbox [53] for electrical, mechanical, and thermal systems. Yet these simulation packages are also subject to the same difficulties in changing fidelity levels and extracting the underlying system of dynamical equations.

To address the limitations, a paradigm shift from simulation packages to wholistic modeling frameworks is needed. Graph- or network-based approaches have demonstrated utility for complex, interconnected systems, such as the IPPT systems considered here. Bond graphs, for example, have been applied as a useful modeling framework for electrical, mechanical, and thermal systems [54]. In [55], concepts of algebraic graphs and computing graphs are introduced to enable simulation of complex cyber-physical systems such as gas pipeline networks. Following along these lines, this work leverages an existing graph-based modeling approach that is modular, variable-fidelity-capable, and generalized to multiple energy domains.

In this chapter, a brief discussion of the graph-based modeling method is presented in the context of planning and control for safe operation of IPPT systems. Section 1 summarizes the fundamentals of the graph-based modeling approach. Section 2 discusses methods for linearizing the graph-based models, which is useful for the development of control design. To facilitate long-term planning, Section 3 discusses methods for reducing the order of a graph-based model. Section 4 provides examples of graph-based models of multi-domain systems, including a hybrid UAV power, propulsion, and thermal system, and a shipboard integrated power system that are used later in Chapter 6.

3.1. FUNDAMENTALS OF THE GRAPH-BASED MODELING FRAMEWORK

Fulfilling the desired qualities mentioned above and others, the graph-based modeling framework used in this work has been used for modeling, control, planning, and design optimization of a variety multi-domain systems, including hybrid energy systems [56], [57], ground vehicle powertrains [58], aircraft powertrain and thermal management systems [42], and vapor compression systems [59]. In this framework, modularity is achieved by representing components as subgraphs that can readily be inserted into or removed from a larger system graph.

Further, component subgraphs can be created with varying levels of fidelity, allowing the user to tailor the model to capture the appropriate level of fidelity for the desired application. Generalization to multiple, disparate energy domains is achieved by representing multi-domain interactions in of terms power and energy, which are universal across energy domains. In this section, a brief introduction to the graph-based modeling framework is presented, and the interested reader is referred to [60]–[65] for additional details and examples of the application of the framework.

In the graph-based modeling framework, a multi-energy-domain system can be modeled as an oriented graph $\mathcal{G}$, such as the one shown below in Figure 3.1, consisting of a set V of nV vertices and a set E of nE edges. A vertex v_i, $i \in \{1, \cdots, nV\}$, represents a single energy storage element. An edge e_j, $j \in \{1, \cdots, nE\}$, connecting two vertices represents power transferred (also referred to as power flow) between the respective energy storage elements. Edges are oriented to denote the assumed positive direction of transferred power. To represent interactions between a system model and the environment or other disturbances, external vertices and edges are used. These are shown as vertices and edges with dashed outlines in Figure 3.1. To quantify energy storage, vertices are associated to domain-specific state variables. While the graph-based modeling framework permits the association of multiple state variables to a single vertex [59], this survey focuses on the case in which each vertex v_i is associated to a single state variable. For internal vertices, the associated state variable x_i has nonnegative energy storage capacitance C_i. The variable associated with an external vertex is x_i^t and its value is exogenous. To quantify power transfer, each internal edge e_j is associated to a power flow P_j^g. For external edges, the associated power flow is denoted P_j^s. Conservation of energy can be applied to each internal vertex such that the rate of energy storage equals the amount of power transferred. In other words, the state variable corresponding to vertex v_i can be obtained by the following

$$C_i \dot{x}_i = \sum_{j \in E^{in}(v_i)} P_j^g - \sum_{k \in E^{out}(v_i)} P_k^g \tag{3.1}$$

where $E^{in}(v_i)$ denotes indices of edges that are oriented into vertex v_i and $E^{out}(v_i)$ denotes indices of edges that are oriented out of vertex v_i. Power transferred along an edge from one vertex to the next can be a function of the state variables associated with these vertices as well as the inputs applied to the system as shown below

$$P_j^g = P_j^g(x_j^{head}, x_j^{tail}, u) \tag{3.2}$$

where x_j^{head} is the vertex into which e_j is oriented, x_j^{tail} is the vertex from which e_j originates, and u is an actuator input.

To compactly represent interconnections between components, an incidence matrix is used. The incidence matrix M specifies the connections of internal vertices to internal edges and is defined as follows.

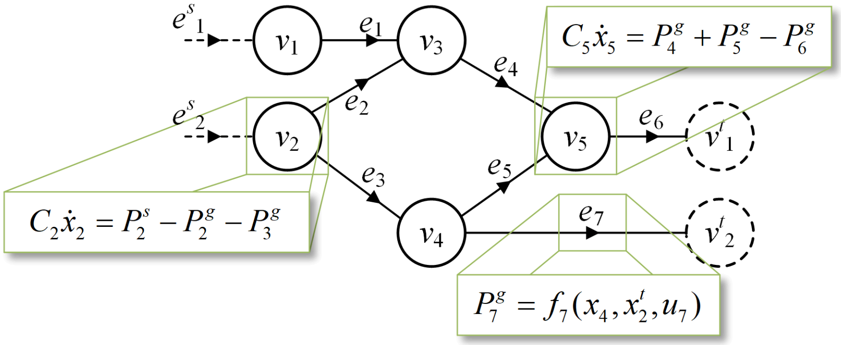

e^s_1
v_1
e_1
v_3
$C_5\dot{x}_5 = P_4^g + P_5^g - P_6^g$
e_4
e_2
e^s_2
v_2
v_5
e_6
v^t_1
e_3
e_5
$C_2\dot{x}_2 = P_2^s - P_2^g - P_3^g$
v_4
e_7
v^t_2
$P_7^g = f_7(x_4, x_2^t, u_7)$

Figure 3.1: Notional graph-based model (modified from [63], [86]*).*

$$M_{i,j} = \begin{cases} -1 & \text{if } e_j \text{ is oriented into } v_i \\ 1 & \text{if } e_j \text{ is oriented out of } v_i \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Defined in a manner analogous to that of the incidence matrix, the matrix D defines connections between internal vertices and external edges. Combining these with (3.1), the full state dynamics can be written as

$$C\dot{x} = -MP^{g} + DP^{s} \tag{3.4}$$

where C is a diagonal matrix of energy storage capacitances C_i, P^g is the vector of power flows, and P^s is the vector of external power flows.

The state dynamics (3.4) obtained using the graph-based modeling framework clearly have the advantage of treating multiple energy domains in a uniform manner. Yet the combination of state dynamics of multiple domains, whose dynamic timescales might vary by several orders of magnitude, can introduce numerical and computational challenges. An approach to address this challenge is to truncate the system by neglecting the dynamic behavior of those states with very short timescales and approximating their dynamic behavior using algebraic relationships. In other words, the algebraic states are assumed to satisfy $\dot{x}_a = 0$. After making such an approximation, (3.4) can be rewritten as

$$\begin{bmatrix} C\dot{x} \\ 0 \end{bmatrix} = -\begin{bmatrix} \overline{M} \\ \underline{M} \end{bmatrix} P^{g} + DP^{s} \tag{3.5}$$

where M has been partitioned into components $\overline{M}$ and $\underline{M}$ which map power flows to dynamic and algebraic states respectively.

A key feature of the graph-based modeling framework is its modularity. Once a model has been developed for a particular component, the model can readily be inserted into or removed from various system models using established graph combination algorithms. A variety of graph-based component models have been developed and validated for electrical, thermal, mechanical, and hydraulic components (see [59]–[63], [66]). By selecting and combining component models from this library of existing models, the process of developing a system model is accelerated. The system models presented in Section 3.4 include many of these existing component models. For the sake of brevity, these existing component models are not reproduced here.

3.2. MODEL LINEARIZATION

While the system of dynamic and algebraic equations provided by (3.5) is useful for numerical simulation, nonlinearities make such a model difficult to employ in model-based control strategies. Instead, to facilitate model-based control it is advantageous to approximate the system (3.5) with a linearized model. The process for linearizing a graph-based model has been presented in [27], [67]. For the sake of completeness, this process is provided here for the particular case in which the power flows' dependence on algebraic states has been eliminated. This can be achieved by solving the lower half of (3.5) to obtain an analytical expression for x_a as a function of x, x^t, P^s , and u, if such a solution exists, and substituting this solution into the upper half of (3.5). Note that a single linearization may not be accurate over the system's full range of operation. For the system models considered in this work, significant nonlinearities and changes in operating modes require linearization about a variety of operating conditions.

Given a system in the form of (3.5), linearization can be done by first linearizing power flows. This results in the following

$$P^g \approx P^g(x_0, x_0^t, u_0) + (x - x_0)\left.\frac{\partial P^g}{\partial x}\right|_0 + (x^t - x_0^t)\left.\frac{\partial P^g}{\partial x^t}\right|_0 + (u - u_0)\left.\frac{\partial P^g}{\partial u}\right|_0 \tag{3.6}$$

where (x_0, x_0^t, u_0) is the linearization point and the notation $\cdot|_0$ is used to denote a quantity evaluated at the linearization point. The Jacobian matrices in (3.6) can be obtained analytically or numerically. Analytical solutions are preferred for the case studies in this work, as these can be used to quickly re-linearize the model about a variety of operating points. However, it is important to note that for some systems (such as those with dynamics governed by lookup tables), it may be impossible to derive analytical solutions to these partial derivatives. In such cases, numerical differentiation can be used, albeit with increased computational expense as this method typically requires multiple finite differencing calculations. Alternatively, the non-differentiable dynamics can be approximated using analytical functions to derive analytical solutions for the partial derivatives.

The dynamic states can then be approximated by combining (3.5) and (3.6) to obtain the following.

$$\dot{x} = -C^{-1}\bar{M}\left(P^g(x_0, x_0^t, u_0) + (x - x_0)\left.\frac{\partial P^g}{\partial x}\right|_0 + (x^t - x_0^t)\left.\frac{\partial P^g}{\partial x^t}\right|_0 + (u - u_0)\left.\frac{\partial P^g}{\partial u}\right|_0\right) + C^{-1}DP^s \tag{3.7}$$

To simplify (3.7), the external states and power flows can be first be combined into a single term $d = [x^t; P^s]$. Then, the linear state dynamics can be written as

$$\dot{x} = Ax + A_d d + B_1 u + B_2 \tag{3.8}$$

where the matrices A, A_d, B_1, and B_2 are given by

$$A = -C^{-1}\bar{M}\left.\frac{\partial P^g}{\partial x}\right|_0, \tag{3.9}$$

$$A_d = -C^{-1}\left[\bar{M} \left. \frac{\partial P^g}{\partial d} \right|_0 \quad -D \right], \tag{3.10}$$

$$B_1 = -C^{-1} \bar{M} \left. \frac{\partial P^g}{\partial u} \right|_0, \tag{3.11}$$

$$B_2 = -C^{-1} \bar{M} \left(P^g(x_o, x_0^t, u_0) - x_0 \left. \frac{\partial P^g}{\partial x} \right|_0 - x_0^t \left. \frac{\partial P^g}{\partial x^t} \right|_0 - u_0 \left. \frac{\partial P^g}{\partial u} \right|_0 \right). \tag{3.12}$$

To use discrete time control strategies, (3.8) can be discretized using, for example, the zero order hold method. Linear, discretized state dynamics are given by the following

$$x_{k+1} = A^z x_k + A_d^z d_k + B_1^z u_k + B_2^z \tag{3.13}$$

where k indexes the discrete time step and superscript z is used to denote the discretized matrices analogous to those defined in (3.9)-(3.12).

Linear state space models such as (3.8), (3.13) are useful for the development of control strategies, but only provide state information. In many cases, control objectives might depend upon other physical signals of the model such as algebraic states or power flows. In fact, the control objectives considered in Chapter 5 of this work depend upon power flows. These can be considered as outputs of the system (3.8). To utilize strategies for linear control, it is advantageous to linearize these additional signals and rewrite them in a familiar state space form. Linearization of power flows has already been performed in (3.6). This can be rewritten in the following state space form

$$P^g = F_1 x + F_2 x^t + F_3 u + F_4, \tag{3.14}$$

where $F_1 = \partial P^g / \partial x|_0$, $F_2 = \partial P^g / \partial x^t|_0$, $F_3 = \partial P^g / \partial u|_0$, and $F_4 = P^g(x_0, x_0^t, u_0) - x_0 \partial P^g / \partial x|_0 - x_0^t \partial P^g / \partial x^t|_0 - u_0 \partial P^g / \partial u|_0$. The same can be done for algebraic states, but this process is omitted here for the sake of brevity.

3.3. REDUCED ORDER MODELING FOR LONG-TERM PLANNING

Graph-based modeling allows the user to develop high-fidelity system models, but the complexity required to achieve high levels of fidelity is not conducive to fast computation. In order to rapidly make decisions about the overall system behavior, using a lower-fidelity model is preferred. Methods for reducing the order of graph-based models have been shown to improve computational efficiency. In this work, a heuristic approach to model reduction is employed in which the user first identifies the subsystems whose energy or power may be subject to constraints in the intended mission planning use case. Then, model reduction is performed by selecting power flows from the graph model which represent the main mechanisms of energy storage, use, or dissipation for each subsystem. These power flows are summed to yield a single power state for each subsystem. The vector of all subsystem power states can be obtained as

$$P = \tilde{M}P^{g} \tag{3.15}$$

where $\tilde{M}$ is a matrix which selects power flows from the graph model. The sign convention of $\tilde{M}$ is as follows

$$\tilde{M}_{i,j} = \begin{cases} 1 & \text{if } P_j^g\text{>0 increases energy stored in subsystem } i \\ -1 & \text{if } P_j^g\text{>0 decreases energy stored in subsystem } i \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

such that a positive subsystem power state increases the respective subsystem's energy state. Energy states of each subsystem are obtained by integrating the power states as shown below.

$$\dot{E} = P \tag{3.17}$$

While (3.15) reduces the number of power flows to be considered in a model, its dependence on the states of the graph model, which may be governed by nonlinear dynamics, is not conducive to long-term planning. Rather, the planning strategies employed in this work assume that, within some error bound, the power states can be approximated as having a piecewise constant derivative (i.e. ramp rate). This assumption is justified by constraining the planner to consider only a predefined set of actions in which this approximation holds accurate. Further, a tracking control method is employed to guarantee that this approximation is satisfied within a predefined error bound. In other words, this assumption amounts to a bound on the error of the tracking controller. A similar approach is taken in [68], [69], in which a similar bound is obtained by assuming that the system's disturbances are bounded. The error bound is discussed in Chapters 4 and 5.

Under this assumption, the approximated power states are given by

$$\dot{\bar{P}} = R \tag{3.18}$$

where an upper bar is used to distinguish the approximated power states from the true subsystem power states obtained by (3.15), and R is a vector containing the power ramp rate of each subsystem. An energy state is also defined for each subsystem as the time integral of the power state as shown below

$$\dot{\bar{E}} = \bar{P} \tag{3.19}$$

where again the upper bar is used to denote approximated energy and power states. The reduced order model obtained by combining (3.18)-(3.19) treats power ramp rates as actuator inputs, yielding a system with simple linear dynamics. This enables fast computation of long-term trajectories for long-term planning.

Each subsystem in the reduced order model is classified into one of two types such that the planner incorporates tracking error for each subsystem differently based on the type. For type 1 subsystems, which correspond to electro-mechanical subsystems, the planner considers error in the power state. For type 2 subsystems, which correspond to thermal subsystems, the planner considers error in the energy state. A type 2 subsystem is associated with a single vertex of the graph model such that its power state includes the sum of all power flows entering and leaving the vertex. This allows the energy state of a type 2 subsystem to be derived from the associated vertex's state variable.

Apart from this heuristic method, additional graph-based model reduction methods include grouping, or clustering, vertices based on their relative time scale or using energy-based metrics [58], [70]. The effectiveness of these approaches has been demonstrated through the implementation of these approaches in hierarchical control and estimation frameworks [58], [70], [71]. The heuristic model reduction method described above is preferred in this work to guarantee that model reduction does not truncate any states that must be constrained in the intended mission planning use case.

3.4. SYSTEM MODELS

In this section, system models are presented to provide examples of the application of the graph-based modeling framework. These example system models are used in Chapter 6 of the dissertation for demonstrations of the planning and control methods. Multiple models are considered to address different aspects of the planning and control problems through the case studies. The first system model described in Section 3.4.1 represents a hybrid electric UAV powertrain. Based upon a physical testbed, this system model is included to permit experimental verification of the methods presented in this work. The second and third system models demonstrate the ability to consider more complex systems (i.e., systems having additional coupled energy domains and increased numbers of subsystems) using this method. In particular, the second system model described in Section 3.4.2 extends the hybrid UAV powertrain to consider dynamics in the thermal domain, in addition to the electrical and mechanical dynamics already considered in Section 3.4.1. The third system model, described in Section 3.4.3, provides an example of a more complex powertrain by more than doubling the number of subsystems compared to the system described in Section 3.4.1.

3.4.1. Hybrid UAV powertrain

A series hybrid UAV powertrain, shown schematically in Figure 3.2(a), is used in this work to demonstrate the methods for planning and control. This powertrain is physically realized by an experimental testbed, shown in Figure 3.2(b). This system includes a lithium-ion battery pack, internal combustion engine, propulsion motor, and auxiliary or avionic electrical load. The

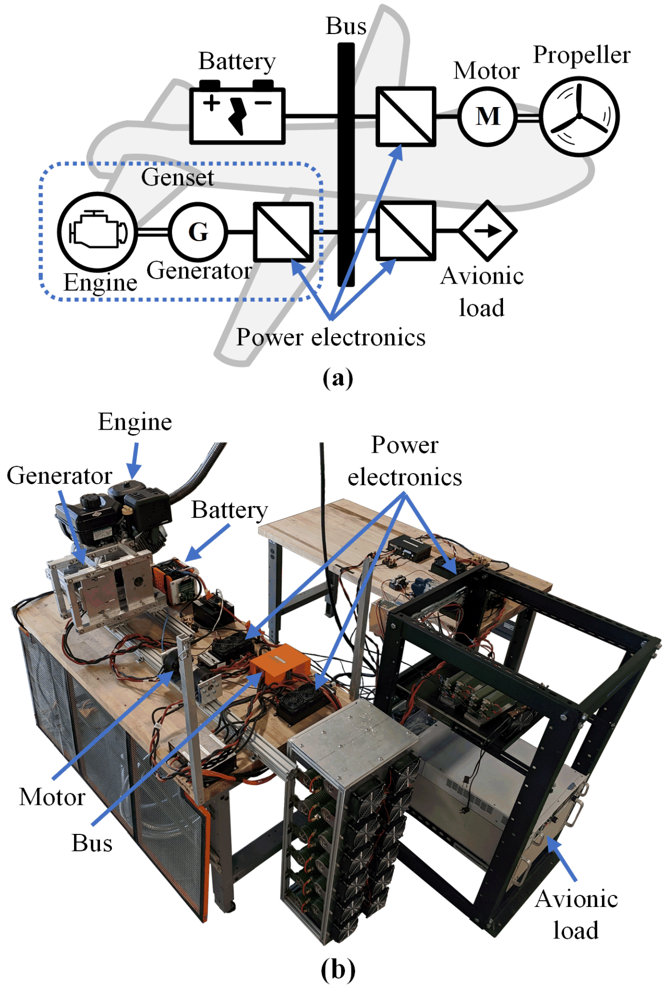


Figure 3.2: Series hybrid electric UAV powertrain (a) schematic and (b) experimental testbed. Adapted from [27], [60].

engine's mechanical power is converted to electrical power and rectified before being distributed among other components via the DC voltage bus. The subsystem consisting of engine, generator, and rectifier is collectively referred to as the genset. In the experimental system used for modeling and controller verification and validation, the dynamic behavior of the vehicle body is emulated using a dynamometer which is coupled to the propulsion motor. The graph-based model for this system was developed and experimentally validated by Aksland and Alleyne [27], [60]. For the sake of completeness, this model is reproduced in Figure 3.3. It includes graph-based component models for the battery, inverter, DC-DC converter, propulsion motor, wye-to-delta transformer, and avionic load. The genset is modeled outside of the graph-based framework and enters the graph-based model as an external current state. The genset current is modeled with an experimentally validated linear, first-order system. In this model, all thermal states are treated as external signals. The planning model for the hybrid UAV powertrain is obtained by selecting

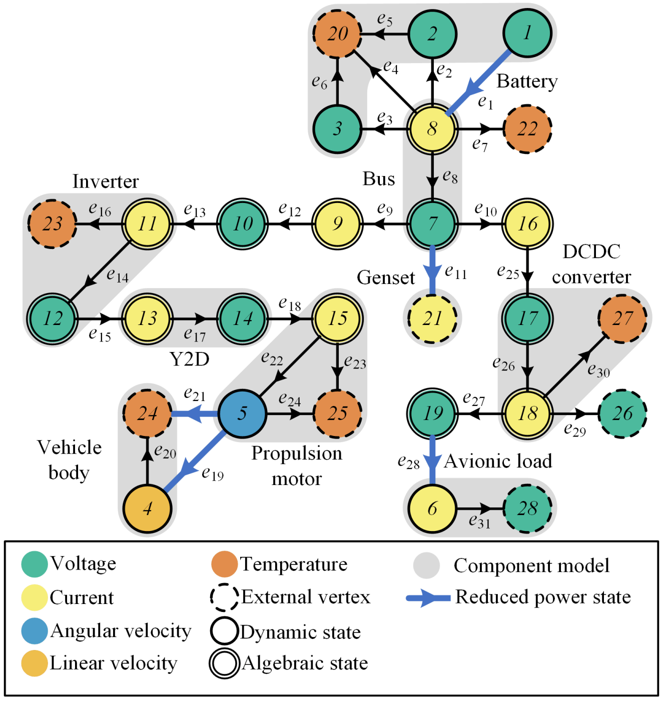


Figure 3.3: Graph-based model of hybrid UAV powertrain. Modified from [27], [60].

power flows for the genset, battery, propulsion, and avionic subsystems. The power flows used to generate the planning model are shown in blue in Figure 3.3.

3.4.2. Hybrid UAV power, propulsion, and thermal management system

The hybrid UAV power, propulsion, and thermal management system (PPTS) considered in this work is an extension of the powertrain discussed above that includes a forced air thermal management system. This extension aims to consider the coupled problem of energy and thermal management for the hybrid UAV system. The system is depicted by the schematic in Figure 3.4. The thermal management system consists of two fans, which separately cool the battery pack air bay and power electronics air bay. The fans are powered from the main voltage bus through two DC-DC converters. In Figure 3.4, shading is used to show the components and fans in each air bay.

To model the thermal response of these electronic components, modifications are made to the powertrain graph model of [27], [60]. With these modifications, the graph model of the hybrid UAV PPTS is provided in Figure 3.5. The genset is modeled using a rectifier, wye-to-delta transformer, and motor. The external temperature vertices for the electrical components (battery, inverter, rectifier, and converters) are treated as internal vertices with dynamic temperature states.

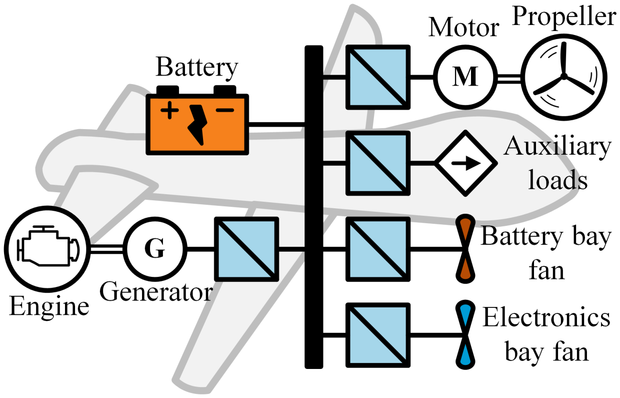
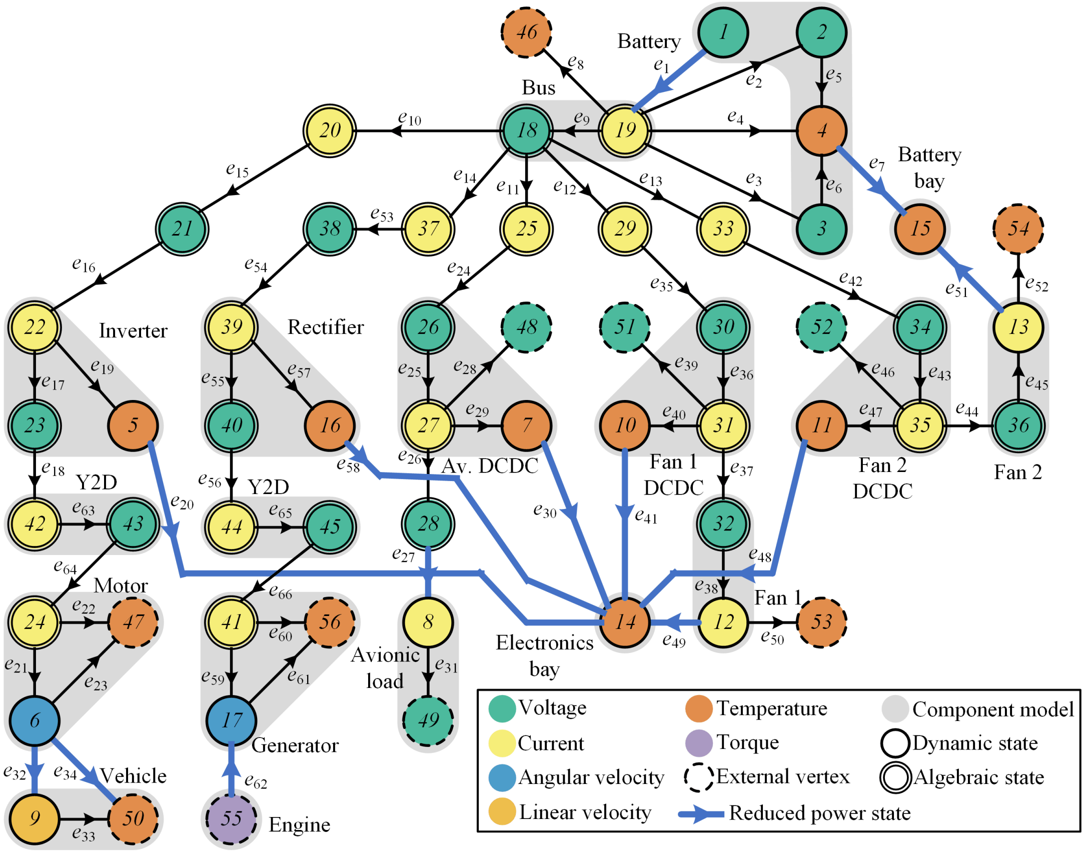


Figure 3.4: Hybrid UAV power, propulsion, and thermal system schematic.

Fans are modeled as resistive electrical loads. The fan model assumes that heat can be removed from the air bay and rejected to another heat sink with a fixed efficiency. In other words, the fan model assumes that a fixed percentage of input electrical power is converted to cooling for the air bay. Appendix B.1 provides more information regarding the fan model.

The hybrid UAV PPTS subsystems include the genset, battery, propulsion, avionics, electronics bay, and battery bay. Power flows from the graph model used to define the power state of each subsystem are depicted with blue outlines in Figure 3.5. This model contains 17 dynamic states, 28 algebraic states, and 11 disturbances. Additionally, there are five actuator inputs that are used to control the generator motor speed, propulsion motor speed, auxiliary load current, and fan



Battery
Bus
Battery bay
Inverter
Rectifier
Y2D
Y2D
Av. DCDC
Fan 1 DCDC
Fan 2 DCDC
Fan 2
Fan 1
Motor
Vehicle
Generator
Engine
Avionic load
Electronics bay
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
e_{1} e_{2} e_{3} e_{4} e_{5} e_{6} e_{7} e_{8} e_{9} e_{10}
e_{11} e_{12} e_{13} e_{14} e_{15} e_{16} e_{17} e_{18} e_{19} e_{20}
e_{21} e_{22} e_{23} e_{24} e_{25} e_{26} e_{27} e_{28} e_{29} e_{30}
e_{31} e_{32} e_{33} e_{34} e_{35} e_{36} e_{37} e_{38} e_{39} e_{40}
e_{41} e_{42} e_{43} e_{44} e_{45} e_{46} e_{47} e_{48} e_{49} e_{50}
e_{51} e_{52} e_{53} e_{54} e_{55} e_{56} e_{57} e_{58} e_{59} e_{60}
e_{61} e_{62} e_{63} e_{64} e_{65} e_{66}
Voltage
Temperature
Component model
Current
Torque
Dynamic state
Angular velocity
External vertex
Algebraic state
Linear velocity
Reduced power state

Figure 3.5: Graph-based model of hybrid UAV PPTS

currents. Controlling these states modifies relevant system outputs including the power levels of the electro-mechanical subsystems and temperatures of the air bays.

3.4.3. Shipboard integrated power system

The shipboard power system (SPS) considered here consists of two identical zones (i.e., power grids) that can be disconnected by opening a switch. The system, shown in Figure 3.6, is a serial hybrid such that all its energy sources are connected to the electrical bus [72]. Each zone consists of a main genset, auxiliary genset, battery pack, propulsion motor, hotel electrical loads, and a voltage bus to which all the components are connected. For the case studies considered in this work, the hotel loads include only base loads that do not exhibit quick ramps or pulses. However, the methods are generalizable to consider combinations of base loads and peak loads

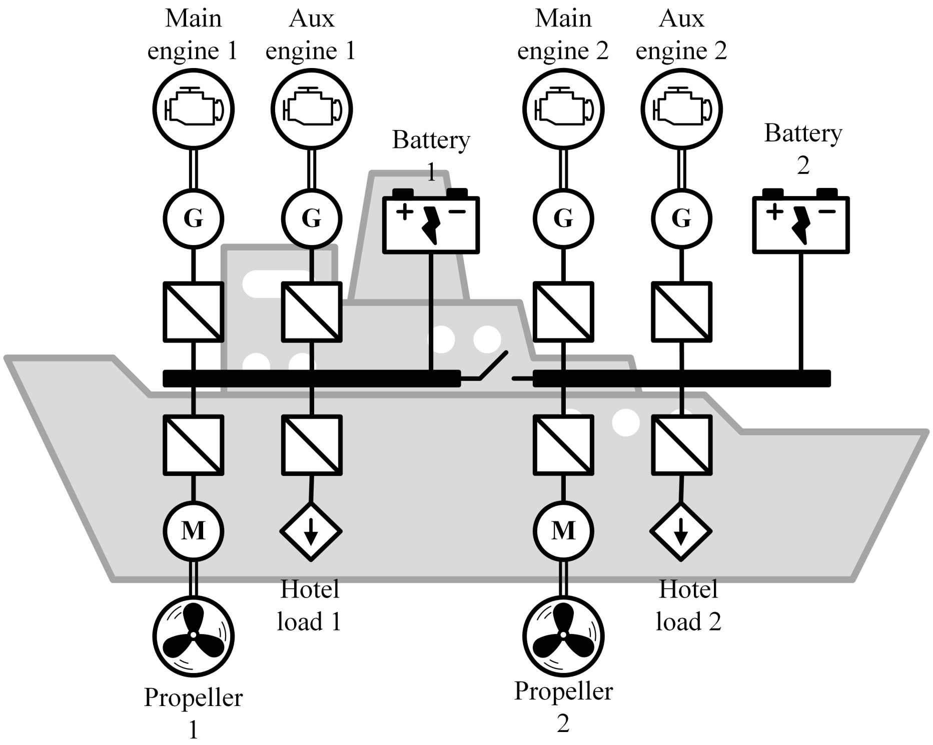


Figure 3.6: Shipboard power system schematic.

with quick ramps. The auxiliary genset in each zone, rated to provide less power than the main engine, provides redundancy and flexibility.

The graph-based model of the SPS is shown in Figure 3.7. As with the hybrid UAV powertrain discussed in Section 3.4, each genset in this system is treated as a single current state which is modeled outside of the graph-based framework using a first-order dynamic. The switch is modeled as a voltage bus with edges that can be switched on or off.

Subsystems of the SPS model are as follows: main genset 1, auxiliary genset 1, battery 1, propulsion 1, hotel 1, main genset 2, auxiliary genset 2, battery 2, propulsion 2, and hotel 2. In terms of number of subsystems, the SPS represents a 2.5X increase in complexity compared to the hybrid UAV powertrain. The power flows from the graph model used to define the reduced order

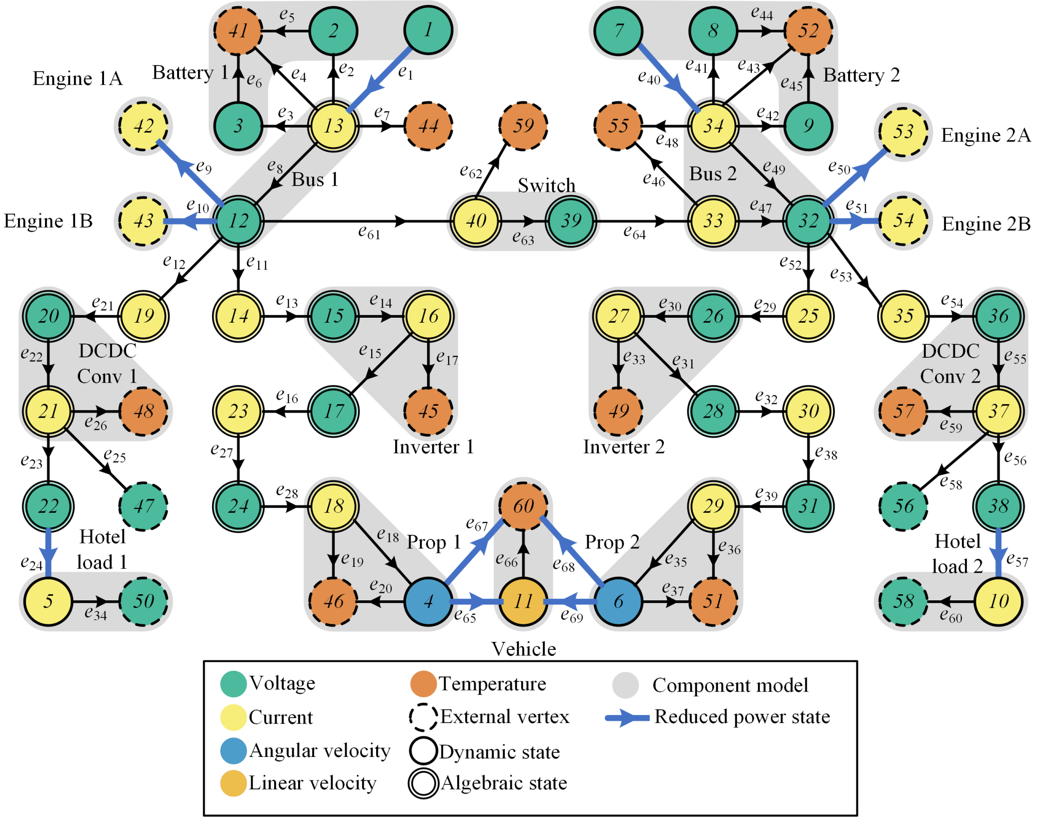


Figure 3.7: Graph-based model of shipboard power system.

power states are indicated with blue outlines in Figure 3.7. Including the genset current states, this model consists of 15 dynamic states and 29 algebraic states. In addition to the switch connecting zones 1 and 2, actuator inputs include genset throttles, converter duty cycles, and inverter duty cycles. These inputs are used to command the genset currents, hotel load currents, and propulsion motor speeds.

3.5. SUMMARY

In summary, modeling is a key aspect of the operation of multi-domain energy systems, yet few existing tools are suitable for the development of model-based control and planning methods. For the long-term planning and tracking control methods used in this work, an existing graph-based modeling tool proves useful. In Section 3.1, fundamentals of the graph-based modeling framework were summarized. Section 3.2 briefly summarized the process of linearizing graph-based models for the application of linear control methods. Section 3.3 described a heuristic method of reducing the order of graph-based models to enable long-term planning. Finally, Section 3.4 provided examples of graph-based system models that are used in Chapter 6 for the demonstration of the two-stage approach.

CHAPTER 4: SAMPLING-BASED MISSION PLANNING

Long-term planning is important to manage slow dynamics of the multi-timescale subsystems comprising IPPT systems. Using a mathematical model of system behavior, planning methods can predict the long-term effect of control decisions on the future system state. These predictions can be used to select control decisions that achieve desired metrics such as energy efficiency, maximal capability, or safety guarantees. There often exists a tradeoff between these high-level metrics. For example, maximizing capability can drive the system into an unsafe regime. Operating a component at its highest power rating for extended time periods can lead to overheating. Selection of a planning method for an IPPT system is influenced not only by these competing metrics, but also by the type of mission to be performed and the timeframe in which planning decisions should be made. For example, mathematically rigorous strategies such as optimal control can maximize the system's theoretical performance given some nominal mission. Yet for complex, high-dimensional IPPT systems, these methods require significant computational time and resources. As such, these methods are inappropriate for real-time operation of an IPPT system as considered in this work, where changing environmental conditions necessitate quick, periodic computations of long-term mission plans. The type of mission also influences the available methods for long-term planning, as many existing planning methods are ill-suited for the missions considered in this work that exhibit nonconvex feasible regions.

The existing methods for long-term planning of IPPT systems during online operation are optimization-based. Many hierarchical control strategies employ optimization methods for long-term planning at the highest level of the hierarchy, coordinating with lower-level controllers to achieve the optimized behavior. Examples of these optimization-based planning methods can be found in [27], [38], [42] for energy and thermal management of aircraft and in [44], [45] for ship energy management. Within hierarchical control frameworks, the long-term planners generally employ gradient-based optimization algorithms to efficiently generate optimal solutions. However, the gradient-based methods are efficient only when the feasible region is convex. When applied to problems with nonconvex feasible regions, gradient-based methods are likely to fail to converge to optimal, or even feasible, solutions. Recently, researchers have begun to consider missions similar to the one illustrated in Figure 1.2 of a hybrid electric aircraft navigating quiet zones as an example of an energy management problem with nonconvex constraints. The existing methods to solve this problem have relied on combinatorial graph search methods such as dynamic programming [32], Djikstra's algorithm [33], and the A* algorithm [34]. These studies demonstrated optimized fuel and/or energy consumption, but as these graph search methods rely on discretization of the feasible space, they are subject to steep increases in computational demand upon refinement of the grid or increase of the problem dimension.

In contrast with the studies discussed above, the focus of this research is to enable IPPT systems to satisfy mission-specific constraints, as stated in problem (2.1), that may change unpredictably over the course of this mission. As such, the ability to rapidly find a feasible solution that safely satisfies these mission-specific constraints is prioritized higher than the ability to optimize efficiency or capability, which may be computationally burdensome. Rather than employing optimization-based methods to solve problem (2.1), this research considers a novel application of sampling-based methods for long-term planning of IPPT systems.

The application of sampling-based planning methods to IPPT systems is discussed in this chapter. Section 4.1 provides a brief overview of sampling-based methods. Section 4.2 details the implementation of a particular sampling-based method, the rapidly-exploring random trees algorithm, for planning of IPPT systems. This implementation relies on a library of energy primitives, or pre-defined behaviors, that are sequenced together using the rapidly-exploring random trees algorithm to generate a feasible solution to problem (2.1). Section 4.3 discusses the use and modification of this algorithm for online re-planning to accommodate changing mission information. Section 4.4 discusses the process used in this work to generate energy primitives. Section 4.5 concludes the chapter.

4.1. OVERVIEW OF SAMPLING-BASED METHODS

Many planning methods use discretization to explicitly represent all feasible configurations in the planning space (e.g., as a grid) in order to find and evaluate the quality of feasible trajectories. Doing so, an optimal path from an initial point to a goal region can be found deterministically using combinatorial search methods such as A* [73] or Dijkstra's algorithm [74]. However, as the number of configurations increases, either through refinement of the discretization or increase in problem dimension, performing a search of all such configurations becomes increasingly computationally expensive and eventually intractable. In contrast, sampling-based planning methods are a class of algorithms that use samples of the planning space, rather than explicit geometrical representations, to plan trajectories. To avoid explicit mathematical representations of obstacles, these algorithms treat collision checking as a black-box routine. This enables the consideration of planning problems with arbitrarily complex obstacles. Further, these algorithms boast high computational efficiency even in high-dimensional planning problems. For these reasons, sampling-based methods are common in motion and path planning applications, enabling robotic systems to quickly find trajectories through environments cluttered with obstacles. Further, the computational efficiency of these methods enables online re-planning for scenarios in which obstacles may vary over time [46].

Sampling-based algorithms can be divided into single- and multiple-query options. Single-query algorithms, intended for unique problems that may be encountered only once, include rapidly-exploring random trees (RRT) and expansive space trees. Multiple query algorithms, on the other hand, are intended for planning scenarios in which multiple paths need to be found through the same set of obstacles starting from different initial conditions and ending at different goal conditions. The most prominent multiple query algorithm is the probabilistic road map algorithm, which performs an initial phase of roadmap construction before planning a trajectory. This roadmap can be used in subsequent queries [75]. RRT-based algorithms are used to solve the single-query problems considered in the case studies of this work. The next section discusses the implementation of these algorithms for energy systems.

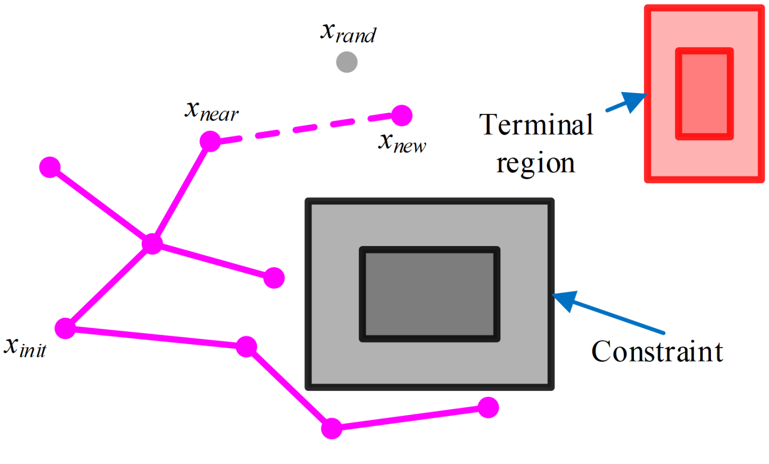
It is important to note that sampling-based methods are subject to shortcomings regarding the properties of plans they generate. Notably, some advanced sampling-based methods have been shown to asymptotically converge to optimal paths [76], but many sampling-based planning algorithms provide suboptimal feasible paths. Further, whereas combinatorial search algorithms provide completeness (the guarantee of returning a solution if one exists), many sampling-based algorithms are only probabilistically complete, meaning that a solution is guaranteed to be returned after an infinite amount of searching time [46]. Yet for the high-dimensional planning problems and time-varying constraints considered in this work, the high computational efficiency afforded by sampling-based algorithms overshadows these shortcomings.

4.2. RRT FOR PLANNING OF IPPT SYSTEMS

The RRT algorithm has proven applicability for a broad class of trajectory planning problems subject to differential constraints. Examples include autonomous driving [77] and exploration of unknown environments [78]. As introduced in [79], this method incrementally grows a search graph, or tree, starting from an initial state x_{init} and extending to a terminal region without colliding with obstacles (i.e., violating constraints). An outline of the basic RRT algorithm is provided in Figure 4.1(a) and an illustration is given in Figure 4.1(b). A search tree is initialized containing only the initial point. The RandomState function selects a random state x_{rand}, and the Extend function attempts to extend the tree towards x_{rand}. In the Extend function, the NearestNbr routine selects the node x_{near} of the tree closest to x_{rand}. Then, the NewState function attempts all actions u to see which action, if any, brings the system to a new, dynamically feasible state near x_{rand} without colliding with obstacles. If such a safe new state is found, it is added to the tree as a new node. This process repeats until a node in the tree reaches the terminal region.

Build_RRT(*X*)	**Extend(*x*, *Tree*)**
InitializeTree(X)→$Tree$	NearestNbr(x,$Tree$)→x_{near}
while $Tree \cap X_{goal} = \emptyset$	NewState(x, x_{near},u)→x_{new}
RandomState→x_{rand}	If Exists(x_{new})
Extend(x_{rand},$Tree$)→$Tree$	AddNode(x_{new},$Tree$)→$Tree$
End	End

(a)



(b)

Figure 4.1: (a) RRT algorithm outline and (b) illustration of tree extension (adapted from [79]*).*

In this work custom RRT algorithms are implemented to enable rapid planning of IPPT systems. In this planning framework, IPPT systems are described by reduced-order models as described in Section 3.3. In this modeling scheme, the system state is given by the following

$$\overline{X} = [\overline{E}^T, \overline{P}^T]^T \tag{4.1}$$

where $\overline{E}$ and $\overline{P}$ are vectors containing the energy and power states, given by (3.19) and (3.18) respectively. For a system consisting of N subsystems, planning is performed in the space $\mathcal{X} \subset \mathbb{R}^{2N}$. Operational and mission-specific constraints represent forbidden regions which must be avoided similarly to obstacles in motion planning problems. While the RRT algorithm permits the consideration of constraints with arbitrarily complex geometries, this work assumes that each constraint $\mathcal{C}_i$ can be represented as a convex polytope with a half-space representation as shown below

$$\mathcal{C}_i = \{X : H_i X \le K_i\} \subset X \tag{4.2}$$

where H_i and K_i define a matrix inequality corresponding to an intersection of half-spaces. This assumption permits the use of computationally efficient constraint-violation-checking routines, which are detailed in Section 4.2.3. The set of all constraints is referred to as the constraint region

$$\mathcal{X}_{cons} = \bigcup_{i=1}^{N_{cons}} \mathcal{C}_i \tag{4.3}$$

where N_{cons} is the total number of constraints in the planning problem. The goal region is also assumed to be represented as a convex polytope as shown below.

$$\mathcal{X}_G = \{X : H_G X \le K_G\} \tag{4.4}$$

A key feature of the RRT algorithm implemented here for planning of IPPT systems is the use of a discrete set of actions, also referred to as energy primitives, where each primitive follows a predefined structure. This yields computational benefits and helps to provide safety assurances, as the safety of each primitive can be verified through extensive testing or simulation. Before describing the RRT algorithm for planning of IPPT systems, this section begins with a definition of the energy primitives used in this work. Following the discussion of energy primitives, additional details are provided regarding the main routines, RandomState and Extend, of the RRT algorithm for planning of IPPT systems. In addition to these routines, termination criteria dictating when to stop extending the tree are discussed.

4.2.1. Definition of energy primitives

Actions in this work are defined as energy primitives that consist of two types of trajectories, trims and transitions. There is a finite set $\mathcal{H} = \{H_1, \cdots, H_{N_H}\}$ of pre-specified trim trajectories. Each trim trajectory corresponds to a constant power operating point in which the system can safely remain for long periods of time. There are N_H^2 transitions such that, for every ordered pair $\{H_i, H_j\}$ of trim trajectories, there is one transition G_i^j that moves the system from H_i to H_j. An energy primitive is defined in this work as a sequence $\{G_i^j, H_j\}$ that performs transition G_i^j, then operates the system at trim trajectory H_j. There are N_H^2 energy primitives such that there is one primitive for each transition, enabling any trim trajectory to be connected to any other trim trajectory. In other words, because each primitive is a pair of one transition and trim, and there are N_H^2 transitions, there are also N_H^2 energy primitives. For primitive $\{G_i^j, H_j\}$, transition G_i^j must start from prior trim trajectory H_i, which is referred to here as the “pre-condition.” The trim trajectory H_j is the “post-condition” of primitive $\{G_i^j, H_j\}$. Long-term mission plans are obtained by constructing sequences of primitives. A primitive sequence of $\{G_i^j, H_j, G_m^n, H_n\}$ is valid only if the post-condition of the first primitive is the pre-condition of the second primitive (i.e., $j = m$). To follow primitive $\{G_i^j, H_j\}$, admissible primitives are those that belong to the set $\mathcal{A}_j = \left\{\{G_j^1, H_1\}, \cdots, \{G_j^{N_H}, H_{N_H}\}\right\}$. If two consecutive primitives in a sequence share the same trim trajectory as in $\{G_i^j, H_j, G_j^j, H_j\}$, then the transition between trim trajectories is trivial and $\{G_i^j, H_j, G_j^j, H_j\} = \{G_i^j, H_j, H_j\}$.

Trim and transition trajectories are both obtained by applying constant ramp rates to the system model (3.18)-(3.19) in phases. Phase k of a trajectory is parameterized by a ramp rate and time duration. Each trim trajectory consists of only one phase with zero ramp rate for constant power operation. The constant power level of each trim trajectory defines the post-condition. All trim trajectories have the same fixed time duration T_H. Transition trajectories can have multiple phases, and the time duration of each transition phase is selected to be smaller than T_H.

Combining its transition and trim trajectories, an energy primitive A^i can be parameterized by the following

- $R^i = \{R_0^i, \ldots, R_{N_R^i - 1}^i\}$: a set of constant ramp rate values for each phase
- $T^i = \{T_0^i, \ldots, T_{N_R^i - 1}^i\}$: a set of times defining the time duration of each phase
- $\overline{P}_0^i$ and $\overline{P}_{N_R^i}^i$: the initial and final power states of the primitive (specified by the initial and final trim trajectories)

where N_R^i is the number of phases in the sequence for primitive i. The final phase corresponds to the trim trajectory, which has $R_{N_R^i - 1}^i = 0$ and $T_{N_R^i - 1}^i = T_H$. Note that the final power state of a primitive corresponds to the constant power operating point of the trim trajectory. If $A^k \in \mathcal{A}_j$ (i.e.,

A^k is admissible after applying primitive A^j), then the parameters R^j and T^j of primitive A^j are defined to ensure that $\overline{P}^j_{N^j_R} = \overline{P}^k_0$ for continuity. The process used in this work to define the parameters of each primitive is discussed at length in Section 4.4.

Example primitive trajectories for a battery subsystem are shown in Figure 4.2. In this example, each primitive includes at most one transition phase shown as a dotted line and one trim trajectory shown as a solid line. In general, there can be more than one transition phase if needed to approximate nonlinear power state behaviors between operating points (see Section 4.4). The operating point of each primitive corresponds to some high-level behavior. For example, the operating points of primitives A^1 and A^4 charge the battery at 500W. This increases the battery's

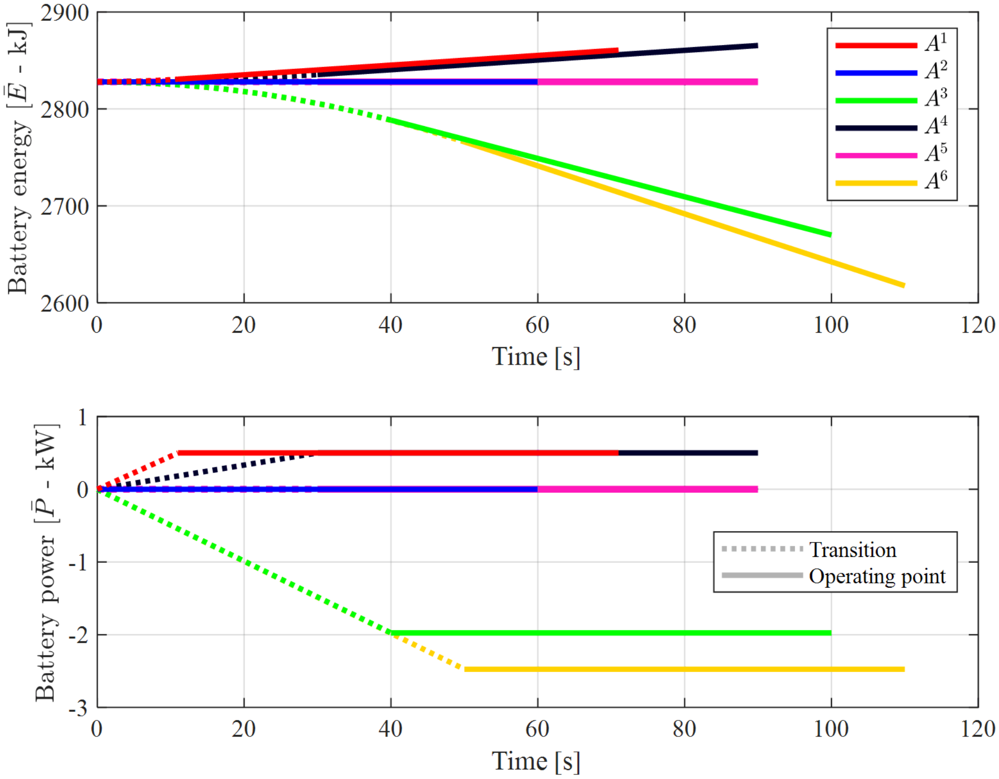


Figure 4.2: Examples of primitive trajectories for a battery subsystem. The initial phase of each primitive corresponds to a transition (shown as a dotted line) from the operating point of the previous primitive to the operating point of the current primitive (shown as a solid line).

energy state. Operating points of primitives A^2 and A^5 sustain the battery at its current energy state by keeping the battery power at 0W. For this example, the previous primitive is A^2, so the primitive sequence $\{A^2, A^2\}$ has a trivial transition phase and only the trim trajectory of primitive A^2 is shown as a solid blue line. A transition occurs at the beginning of primitive A^5 as, even though battery power remains at 0W, other subsystems not shown here move to a different operating point. Finally, primitives A^3 and A^6 discharge the battery at different power levels, reducing the battery energy state. In this example, the time duration of trim trajectories is 60s.

4.2.2. Random state selection

After initializing the tree, the first step of the RRT algorithm is the RandomState function's selection of a random state. The sampling strategy used in this step has a significant effect on the performance of the RRT algorithm [79]. Sampling from uniform distributions will cause the planner to explore the space in a rapid, uniform manner. Sampling distributions can also be biased towards desirable regions to improve convergence. The sampling strategy should be chosen carefully to balance exploration of the space with rapid progression towards the terminal region. While rapid progression towards the terminal region is desirable, exploration is also important to allow the algorithm to find trajectories that avoid constraints that might block the path to the terminal region.

In the case studies, random samples are taken from a normal distribution as given below

$$X_{rand} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4.5}$$

where the mean vector $\boldsymbol{\mu}$ is near the center of the goal region. The covariance matrix $\boldsymbol{\Sigma}$ of this distribution can be adjusted to improve the algorithm's performance, taking into account the differing magnitudes of elements of the state vector (4.1).

4.2.3. Tree extension

The second step of the RRT algorithm is to extend the tree towards the random state using the Extend function. In this section, the NearestNeighbor, NewState, and AddNode routines of the Extend function are described. The NewState function is further subdivided into subroutines including simulation of trajectories, checking for constraint violation, and selection of a new state.

4.2.3.1. Selection of nearest neighbor

The Extend function begins by finding the closest node in the tree to the random state using the NearestNeighbor function. The distance metric used to quantify nearness in this step is another consideration that significantly affects the performance of the algorithm and the quality of generated mission plans. When sample points are biased towards the goal region, a weighted norm distance metric, as shown below, can help to draw trajectories toward the goal.

$$\rho(X_1, X_2) = \left\| X_1 - X_2 \right\|_{\Gamma_X}^2 \tag{4.6}$$

In (4.6), the notation $\left\| x \right\|_{\Gamma_X}^2 = x^T \Gamma_X x$ is used, where Γ_X is a positive definite weighting matrix.

When selecting the nearest neighbor, it is important to take into consideration the possibility that some nodes in the tree may be "trapped" (i.e., all actions starting from this node lead to constraint violation(s)). These nodes should be neglected when selecting the nearest neighbor. Without doing so, the planner may repeatedly try to extend the tree from nodes that cannot be extended, reducing the algorithm's performance. Letting $V_{trapped}$ be the set of nodes in the tree that are trapped, the NearestNeighbor function returns the following.

$$\bar{X}_{near} = \arg \min_{\bar{X} \notin V_{trapped}} \left(\rho(\bar{X}, X_{rand}) \right) \tag{4.7}$$

4.2.3.2. Search for a new state

After selecting $\overline{X}_{near}$, the NewState function attempts actions (i.e. primitives) to move the system to a new state near X_{rand}. Let A^j be the primitive that was applied to bring the system state to $\overline{X}_{near}$. While the full set of admissible actions to extend $\overline{X}_{near}$ is $\mathcal{A}_j$, the NewState function is limited to consider only those admissible actions that have not been "expanded" in previous iterations. "Expanded" actions are those that have been attempted from $\overline{X}_{near}$ leading to either constraint violation or the successful addition of a new state to the tree. Neglecting expanded actions improves computational efficiency by preventing the repeated computation of trajectories that have already been found to violate constraints and limiting the addition of duplicate nodes to the tree [80]. Let $a_{expanded} \subset \{1,\cdots,N_H^2\}$ be the set of indices of admissible actions that have been expanded. The planning model is simulated to obtain a trajectory of energy and power states for each admissible action $A^i \in \mathcal{A}_j$, where $i \notin a_{expanded}$. Each of these trajectories is checked for violation of constraints, and a new state is selected as the terminal point of one of these trajectories. These steps are described at length below.

Simulation of primitive trajectories and model error

Defining primitives as in Section 4.2.1, the resulting state dynamics from (3.18)-(3.19) can be quickly and efficiently calculated to find the energy and power states at each phase. During phase k of primitive A^i, for $k = 1,\ldots,N_R$, the nominal energy and power states are given by

$$\overline{E}(k,i,t) = \overline{E}_{k-1}^i + \overline{P}_{k-1}^i t + \tfrac{1}{2} R_{k-1}^i t^2 \tag{4.8}$$

$$\overline{P}(k,i,t) = \overline{P}_{k-1}^i + R_{k-1}^i t \tag{4.9}$$

where $t \in [0, T_{k-1}^i]$ and $\bar{E}_k^i = \bar{E}(k, i, T_{k-1}^i)$, $\bar{P}_k^i = \bar{P}(k, i, T_{k-1}^i)$ are the final conditions of the energy and power states for phase k. Phase 1 of every admissible primitive has the same initial energy and power state, $\bar{X}_{near} = [\bar{E}_0^{iT}, \bar{P}_0^{iT}]^T$ for all $A^i \in \mathcal{A}_j$. After the final phase, the energy and power states are given by $\bar{E}_{N_R^i}^i$ and $\bar{P}_{N_R^i}^i$ respectively.

Before checking trajectories for violation of constraints it is useful to derive a bound for the propagation of tracking error between the planning model and the true system. This bound can be used to ensure that, even in the worst case of model error, constraints will not be violated. In this work, two types of model error are considered where each subsystem is classified as one of the two types. Type 1 subsystems have bounded power state error, while type 2 subsystems have bounded energy state error.

For subsystem j of type 1, error impacts the power state as shown below

$$P(j) = \bar{P}(j) + W(j) \tag{4.10}$$

where $W(j)$ is the model error, which is assumed to be bounded to a closed, bounded set $\mathcal{W}_k^i(j)$ containing the origin. Note that the error bound $\mathcal{W}_k^i(j)$ may vary for different stages k of primitive A^i. Define energy state error as the difference between the true and planned energy states, $e_E = E - \bar{E}$. Combining (3.17), (3.19), and (4.10), the derivative of energy state error is given by the following.

$$\dot{e}_E = W \tag{4.11}$$

Reachable sets are used to bound the model error at each stage k of the primitive as given below

$$\mathcal{E}_k^i(j) = \mathcal{E}_0^i(j) + \sum_{l=1}^{k-1} T_{l-1}^i \mathcal{W}_l^i(j) \tag{4.12}$$

$$\mathcal{P}_k^i(j) = \mathcal{W}_k^i(j) \tag{4.13}$$

where $\mathcal{E}_k^i(j)$ is the energy error set and $\mathcal{P}_k^i(j)$ is the power error set for subsystem j during stage k of primitive A^i. Note that the energy error set grows linearly with time, and (4.12) describes the propagation of energy state error up to the end of stage k. The initial condition of the energy state error set is $\mathcal{E}_0^i(j)$, which includes the error accumulated from all previous primitives by applying (4.12).

For a type 2 subsystem j, uncertainty impacts the energy states directly as shown below

$$E(j) = \bar{E}(j) + W(j) \tag{4.14}$$

where $W(j) \in \mathcal{W}_k^i(j)$, such that the propagation of energy state error is described by the following.

$$\mathcal{E}_k^i(j) = \mathcal{W}_k^i(j) \tag{4.15}$$

Error models of type 2 are well suited for thermal subsystems, as bounding the thermal energy of a component is similar to regulating its temperature. However, it is difficult in practice to tightly bound the power states of thermal systems, which correspond to heat transfer rates. Driven by temperature differences between a subsystem and its surroundings, the rate of heat transfer into a subsystem can be subject to large variations even for small changes in temperature. Because of these variations, heat transfer rates are typically more difficult to maintain within a tight bound than a single subsystem temperature state. In this study, the power state error set $\mathcal{P}_k^i(j)$ for type 2 subsystems j is a large interval set that is held constant for each primitive. For this reason, the power states of type 2 subsystems are unconstrained in this work.

For $\bar{X}$ given by (4.8)-(4.9), the state error $X - \bar{X}$ at stage k of primitive A^i is bounded to the following set.

$$X - \bar{X} \in \mathcal{X}_{\mathcal{W}_k^i} = \{[E^T, P^T]^T : E \in \mathcal{E}_k^i, P \in \mathcal{P}_k^i\} \tag{4.16}$$

Note that $\mathcal{X}_{\mathcal{W}_k^i}$ is parameterized by the uncertainty bound $\mathcal{W}_k^i$. In this work, each $\mathcal{W}_k^i$ is an interval set that is symmetric about the origin, such that $\mathcal{X}_{\mathcal{W}_k^i}$ is also a symmetric interval set that can be written as

$$\mathcal{X}_{\mathcal{W}_k^i} = \mathcal{I}\left([-L_k^i, L_k^i]\right) \tag{4.17}$$

where the notation $\mathcal{I}([x_l, x_u])$ is used to represent an interval set whose vector of lower bounds is x_l and upper bounds is x_u. This parameterization of $\mathcal{W}_k^i$ permits the use of highly efficient constraint-violation-checking routines as discussed below.

Check for constraint violation

For a primitive to be free of constraint violations (i.e., feasible), the system state must not enter the constraint region even in the worst case of model error. Satisfaction of the following sufficient condition ensures a constraint-violation-free trajectory for stage k of primitive A^i

$$\bar{X} + \mathcal{X}_{\mathcal{W}_k^i} \cap \mathcal{X}_{con} = \varnothing \tag{4.18}$$

for all $\bar{X}$ along the simulated trajectory between $\bar{X}_{k-1}^i$ and $\bar{X}_k^i$. Checking equation (4.18) requires a set intersection computation. To avoid set intersection calculations, which can be computationally expensive, an equivalent condition for feasibility is given by the following set membership condition

$$\bar{X} \notin \mathcal{X}_{con} + \mathcal{X}_{\mathcal{W}_k^i} \tag{4.19}$$

where $\mathcal{X}_{con} + \mathcal{X}_{\mathcal{W}_k^i}$ is a "bloated" constraint region, obtained by adding $\mathcal{X}_{\mathcal{W}_k^i}$ to every constraint in the region. This concept is illustrated in Figure 4.3 for a single polytopic constraint, $\mathcal{C}_j$, as defined in (4.2).

While (4.19) is easier to compute than (4.18), this condition must be ensured for each $\bar{X}$ in the trajectory. In the special case that the trajectory is given by a line segment, a sufficient condition to ensure that the line segment does not violate constraint $\mathcal{C}_j$ is given below in terms of the segment's endpoints X_1, X_2.

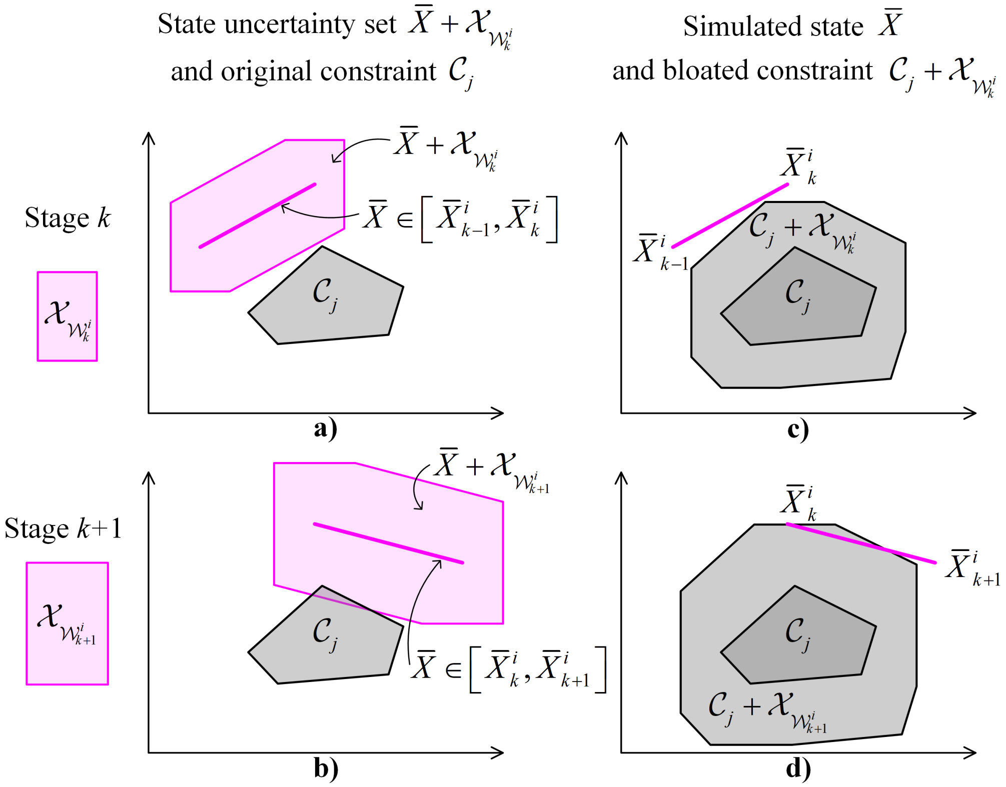

State uncertainty set $\bar{X} + \mathcal{X}_{\mathcal{W}_k^i}$
and original constraint $\mathcal{C}_j$
Simulated state $\bar{X}$
and bloated constraint $\mathcal{C}_j + \mathcal{X}_{\mathcal{W}_k^i}$
Stage k
$\mathcal{X}_{\mathcal{W}_k^i}$
$\bar{X} + \mathcal{X}_{\mathcal{W}_k^i}$
$\bar{X} \in \left[\bar{X}_{k-1}^i, \bar{X}_k^i\right]$
$\mathcal{C}_j$
a)
$\bar{X}_k^i$
$\bar{X}_{k-1}^i$
$\mathcal{C}_j + \mathcal{X}_{\mathcal{W}_k^i}$
$\mathcal{C}_j$
c)
Stage k+1
$\mathcal{X}_{\mathcal{W}_{k+1}^i}$
$\bar{X} + \mathcal{X}_{\mathcal{W}_{k+1}^i}$
$\mathcal{C}_j$
$\bar{X} \in \left[\bar{X}_k^i, \bar{X}_{k+1}^i\right]$
b)
$\bar{X}_k^i$
$\bar{X}_{k+1}^i$
$\mathcal{C}_j$
$\mathcal{C}_j + \mathcal{X}_{\mathcal{W}_{k+1}^i}$
d)

Figure 4.3: Illustration of bloating constraints to account for model error. Subplot a) shows, at stage k of primitive A^i, the state uncertainty set corresponding to potential values of the true state $X \in \bar{X} + \mathcal{X}_{\mathcal{W}_k^i}$. Checking this set for constraint violation is equivalent to checking whether the line segment in c) violates the bloated constraint $\mathcal{C}_j + \mathcal{X}_{\mathcal{W}_k^i}$. There is no constraint violation for stage k. Subplots b) and d) show the next stage, k+1, of primitive A^i. In this stage, the trajectory segment violates the bloated constraint $\mathcal{C}_j + \mathcal{X}_{\mathcal{W}_{k+1}^i}$.

$$\bigvee_{s=1}^{rows(H_j)} \left(\left(H_j^{(s)} X_1 > K_j^{(s)} + \left| H_j^{(s)} L_k^i \right| \right) \wedge \left(H_j^{(s)} X_2 > K_j^{(s)} + \left| H_j^{(s)} L_k^i \right| \right) \right) = \text{True} \tag{4.20}$$

In (4.20), H_j and K_j provide the half-space representation of $\mathcal{C}_j$ (see (4.2)), s indexes rows of H_j and K_j, $rows(\cdot)$ denotes the number of rows of a matrix, and the notation $A^{(s)}$ is used to denote row s of matrix A. In short, this condition checks whether there is any row s such that both endpoints lie a distance L_k^i outside of the half-space defined by $H_j^{(s)} X \leq K_j^{(s)}$. Condition (4.20) is modified from Lemma 4 of [81], to which readers are referred for a proof.

Condition (4.20) is valid for linear trajectory segments, which are obtained in trim trajectories when $R_k^i = 0$. During transitions, the nominal energy states (4.8) are quadratic with respect to time leading to curved trajectory segments for which it is more challenging to check for constraint violation. In this work these curved segments are approximated using the line segment $\hat{X}$ connecting their endpoints, and the error set $\mathcal{X}_{\mathcal{W}_k^i}$ is bloated to account for deviation between $\bar{X}$ and $\hat{X}$. This is illustrated in Figure 4.4. Denoted S_k^i for stage k of primitive A^i, this deviation is given by the following.

$$S_k^i = \begin{bmatrix} \frac{1}{8} R_{k-1}^i {T_{k-1}^i}^2 \\ 0 \end{bmatrix} \tag{4.21}$$

Note that the lower half of S_k^i is zero because power state trajectories are linear during transitions. Defining $\tilde{L}_k^i$ as follows,

$$\tilde{L}_k^i = L_k^i + S_k^i \tag{4.22}$$

the following condition is sufficient for a curved trajectory segment to be free of constraint violations,

$$\bigwedge_{j=1}^{N_{cons}} v_{safe}(\mathcal{C}_j, X_1, X_2, k) = \text{True} \tag{4.23}$$

where $v_{safe}(\mathcal{C}_j, X_1, X_2, k)$ is defined as follows.

$$v_{safe}(\mathcal{C}_j, X_1, X_2, k) = \bigvee_{s=1}^{rows(H_j)} \left(\left(H_j^{(s)} X_1 > K_j^{(s)} + \left| H_j^{(s)} \tilde{L}_k^i \right| \right) \wedge \left(H_j^{(s)} X_2 > K_j^{(s)} + \left| H_j^{(s)} \tilde{L}_k^i \right| \right) \right) \tag{4.24}$$

Note that, as shown in Figure 4.4, equation (4.24) introduces some conservatism by approximating the error between the curved and linear trajectory segments as an interval defined by the maximum deviation. However, the primitives are designed to remain at the operating points with $R^i_{N^i_R} = 0$ for long time durations so this conservatism is not very restrictive. In the case that $R^i_{k-1} = 0$, $v_{safe}(\mathcal{C}_j, X_1, X_2, k)$ simplifies to the lefthand side of (4.20).

To ensure that constraints are not violated when primitive A^i is applied, condition (4.24) is checked for each constraint $\mathcal{C}_j$, and for each consecutive pair of endpoints $\{\bar{X}^i_{k-1}, \bar{X}^i_k\}$ for

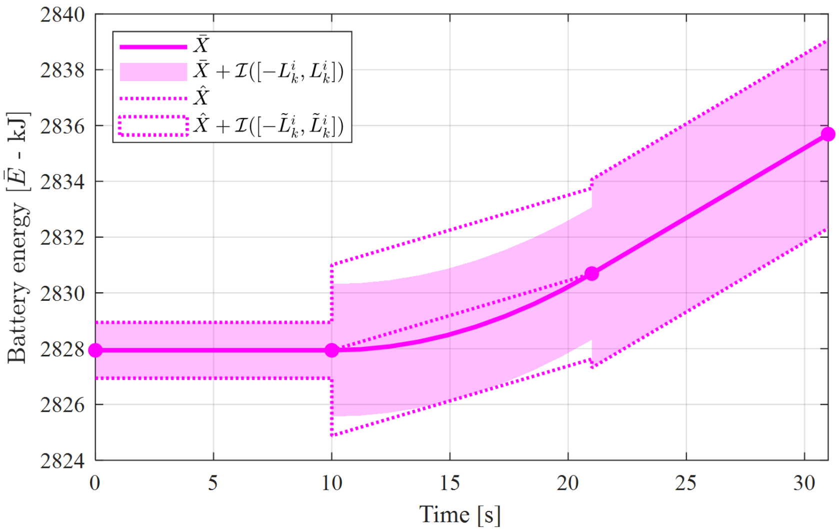


Figure 4.4: Illustration of type 1 error sets for a curved trajectory segment. The curved trajectory segment can be over-approximated by a line segment, adding the linearization error to the type 1 error set.

$k = 1, \ldots, N_R^i$. Indices of unsafe primitives (those for which the simulated trajectory violates constraints) and safe primitives are defined by the following.

$$a_{unsafe} = \left\{ i : \bigwedge_{k=1}^{N_R^i} \left(\bigwedge_{j=1}^{N_{cons}} v_{safe}(\mathcal{C}_j, \bar{X}_{k-1}^i, \bar{X}_k^i, k) \right) = \text{False for } A^i \right\} \tag{4.25}$$

$$a_{safe} = \left\{ i : \bigwedge_{k=1}^{N_R^i} \left(\bigwedge_{j=1}^{N_{cons}} v_{safe}(\mathcal{C}_j, \bar{X}_{k-1}^i, \bar{X}_k^i, k) \right) = \text{True for } A^i \right\} \tag{4.26}$$

A key advantage of this method is the ability to test the set-based condition (4.18) using vector algebra and logical conditions, rather than set-based calculations which can be computationally expensive. Because hundreds of thousands of constraint violation checks may be needed in a given planning scenario, the computational improvements afforded by making these conservative approximations are especially advantageous.

Selection of new state

After simulating the nominal trajectory for each unexpanded primitive and checking these trajectories for constraint violations, a new state can be selected. Define $X_F = \{\bar{X}_{N_R^i}^i : i \notin a_{expanded}, i \notin a_{unsafe}\}$ as the set of final states for all admissible primitives that are unexpanded and whose trajectories do not violate constraints. Then, the new state is selected as follows

$$\bar{X}_{new} = \arg\min_{X \in X_F} \rho(X, X_{rand}) \tag{4.27}$$

where ρ is given in (4.6). If no safe new state is found, the set X_F is empty. In this case, the NewState function returns a flag indicating that $\bar{X}_{near}$ is “trapped.”

4.2.3.3. Addition of new node

Once a new node has been found by the NewState function, it is inserted into the tree. Along with the new node, additional information is saved that may be useful for the next iteration, or to facilitate the construction of full trajectories upon algorithm termination. For this study, the following information is initialized each time a node is added to the tree.

- *Node:* Nodes are added to the tree by appending them to a matrix V where each column corresponds to one node. The node corresponds to the new state $\bar{X}_{new} = \bar{X}^{i}_{N_R^i}$, where i is the index of the primitive applied to move the system to $\bar{X}_{new}$. N_V is the number of nodes in V before inserting the new node.
- *Edge:* The edge to be added to the tree is defined as an ordered pair of node indices $[i_{near}, i_{new}]$, where i_{near} is the index of the node corresponding to $\bar{X}_{near}$, and $i_{new} = N_V + 1$ is the index of the new node. Edges are saved in an $N_V \times 2$ matrix. A new edge is added to the tree by appending the node indices of the new edge as new rows.
- *Action index:* The index $i \in \{1, \cdots, N_A\}$ of the action applied to move the state from $\bar{X}_{near}$ to $\bar{X}_{new}$ is appended to a vector of N_V elements consisting of the action indices for all previous nodes.
- *Trajectory segment:* Information about the trajectory segment is saved as a set of states and corresponding time points along the trajectory connecting $\bar{X}_{near}$ to $\bar{X}_{new}$.
- *Trapped flag:* A flag $v_{trapped}$, initialized at zero, indicates that the new node is not "trapped." The new entry $v_{trapped}$ is appended to a vector of N_V elements consisting of the trapped flags for all previous nodes.

- *Constraint bloating amount:* Because the energy state error accumulates for subsystems of type 1, the cumulative constraint bloating amount is saved after action A^i is used to move the state from $\overline{X}_{near}$ to $\overline{X}_{new}$. To save this information compactly, the vector $L^i_{N^i_R}$ (see (4.17)) is appended to a matrix containing the bloating amount for all previous nodes.
- *Expanded vector:* The set $a_{expanded}$, initialized as an empty set, contains the indices of all actions that have been "expanded" from the current node. For each node, this information is converted to a vector $e_{expanded}$, where element i of $e_{expanded}$ is set to 1 if $i \in a_{expanded}$ and 0 otherwise. For each new node, the vector $e_{expanded}$ is initialized to contain all zeros and is appended to a matrix storing the corresponding vector for all previous nodes.

In addition to this information that is initialized for each new node of the tree, the tree is updated to reflect the following new information regarding the previous node corresponding to $\overline{X}_{near}$.

- *Updated trapped flag:* If the NewState function indicates that $\overline{X}_{near}$ is trapped, the flag $v_{trapped}$ corresponding to that node is set to 1. In subsequent iterations, the set $V_{trapped}$ will include $\overline{X}_{near}$.
- *Updated expanded vector:* The NewState function returns either a flag indicating that $\overline{X}_{near}$ is trapped or a new state $\overline{X}_{new}$. In the former case, NewState has determined that all actions have been explored and there are no safe actions starting from $\overline{X}_{near}$. In this case, all elements of the vector $e_{expanded}$ for node $\overline{X}_{near}$ are set to 1. In the latter case,

an action A^i has been expanded to move the state from $\overline{X}_{near}$ to $\overline{X}_{new}$. Other actions may have also been expanded and found to violate constraints, such that $a_{unsafe} \neq \varnothing$. The set $a'_{expanded} = \{i, a_{unsafe}\}$ includes all actions that were expanded in this call of the NewState function. In this case, element k of the vector $e_{expanded}$ for node $\overline{X}_{near}$ is set to 1 for all $k \in a'_{expanded}$.

4.2.4. Termination criteria

The process described in Sections 4.2.2-4.2.3.3 repeats until some termination criteria are satisfied. In this work, the algorithm terminates if one of the following three termination criteria are met: 1) a node in the tree has reached the goal region; 2) all nodes in the tree are trapped; 3) a predefined number of iterations have been completed without any nodes reaching the terminal region. If criterion 1 is met, the algorithm has successfully found a feasible trajectory from the initial point to the goal region. If criterion 2 is met, the algorithm has determined that there is no such feasible trajectory. If criterion 3 is met, the algorithm has reached a timeout condition.

After each node is added to the tree, criterion 1 is checked to see if the new node has reached the goal region. For polytopic goal regions described by (4.4), a state $\overline{X}$ is within the goal region if the following condition is satisfied

$$H_G \overline{X} \leq K_G \tag{4.28}$$

for every row of H_G and K_G. However, satisfaction of this condition for a node $\overline{X}^i_{N^i_R}$ is not sufficient for the true state to land within the goal region. Rather, as in (4.18), model error must be considered to ensure that

$$\overline{X}^i_{N^i_R} + \mathcal{X}_{\mathcal{W}^i_{N_R}} \subset \mathcal{X}_G \tag{4.29}$$

where $\mathcal{X}_{\mathcal{W}_{N_R}^i}$ is a set that bounds model error after action A^i has been applied to move the state to $\bar{X}_{N_R^i}^i$. The set $\mathcal{X}_{\mathcal{W}_{N_R}^i}$ is calculated using the methods described in Section 4.2.3. To avoid checking the set-based condition (4.29), which can be computationally intensive, an equivalent condition to check is whether the nominal state lies in a goal region "shrunken" by $\mathcal{X}_{\mathcal{W}_{N_R}^i}$ as follows.

$$\bar{X}_{N_R^i}^i \in \mathcal{X}_G - \mathcal{X}_{\mathcal{W}_{N_R}^i} \tag{4.30}$$

This shrinking of the goal region is illustrated in Figure 4.5. To check this condition in a manner similar to (4.28), each row s of the vector K_G is modified as follows to shrink the goal region

$$\tilde{K}_G^{(s)} = K_G^{(s)} - \left| H_G^{(s)} \tilde{L}_{N_R^i}^i \right| \tag{4.31}$$

where i is the index of the primitive applied to move to the current node and $\tilde{L}_{N_R^i}^i$, given by (4.22)

, describes error accumulated by applying all primitives in the sequence up to primitive i. Doing so, the nominal state will land in the shrunken goal region if the following condition

$$H_G \bar{X} \leq \tilde{K}_G \tag{4.32}$$

holds for every row of H_G and $\tilde{K}_G$. If the nominal state of a node lands in this shrunken goal region, then the true state is guaranteed to land in the goal region, so the algorithm terminates. Upon successful termination, the sequence of primitives leading from the first node to the final node defines the feasible mission plan. The full planned trajectory is constructed using information described in Section 4.2.3.3 that is saved for each node.

Criteria 2 and 3 are also checked each iteration. Criteria 2 is checked after generating a random sample and prior to selecting the nearest neighbor. Criteria 3 is checked prior to selecting a random sample. If either condition is met, the algorithm terminates with an error flag.

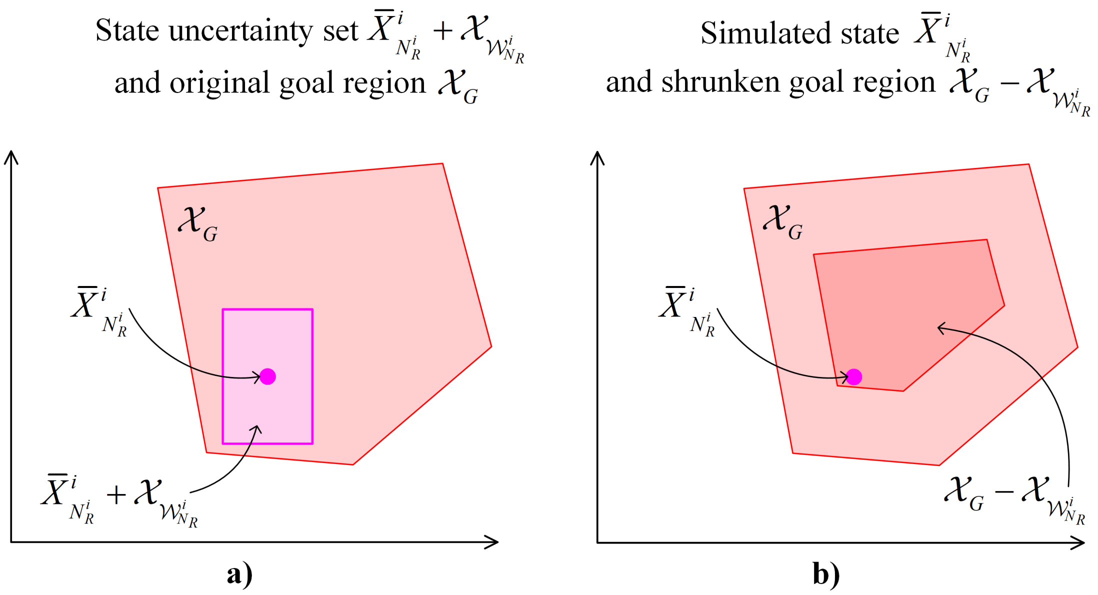

State uncertainty set $\bar{X}^i_{N_R^i} + \mathcal{X}_{\mathcal{W}^i_{N_R}}$ and original goal region $\mathcal{X}_G$
Simulated state $\bar{X}^i_{N_R^i}$ and shrunken goal region $\mathcal{X}_G - \mathcal{X}_{\mathcal{W}^i_{N_R}}$
$\mathcal{X}_G$
$\bar{X}^i_{N_R^i}$
$\bar{X}^i_{N_R^i} + \mathcal{X}_{\mathcal{W}^i_{N_R}}$
$\mathcal{X}_G$
$\bar{X}^i_{N_R^i}$
$\mathcal{X}_G - \mathcal{X}_{\mathcal{W}^i_{N_R}}$
a)
b)

Figure 4.5: Illustration of shrinking the goal region to account for model error. Checking a) whether the state uncertainty set $\bar{X}^i_{N_R^i} + \mathcal{X}_{\mathcal{W}^i_{N_R}}$ is fully included in the goal region $\mathcal{X}_G$ is equivalent to checking whether the point $\bar{X}^i_{N_R^i}$ is in the shrunken goal region $\mathcal{X}_G - \mathcal{X}_{\mathcal{W}^i_{N_R}}$.

4.3. ONLINE RE-PLANNING

The methods described in Section 4.2 generate a single, feasible mission plan based on the currently known mission information. However, changes in the mission information could cause this mission plan to lose feasibility. Due to its ability to rapidly explore a feasible space, RRT can be used for online re-planning to maintain feasibility in the case that mission specifications change over time. In the following section, a method for online re-planning using RRT is discussed. In Section 4.3.2, an improved method for online re-planning using a variant of the RRT algorithm is discussed.

4.3.1. Re-planning with RRT

The RRT algorithm is designed to rapidly explore complex geometrical spaces with no prior information. When re-planning, though mission information might look similar from one call of the RRT to the next, the RRT always starts “from scratch” using the system’s current state as the initial vertex in the tree. This permits the algorithm described in Section 4.2 to be used for re-planning without modification. The remaining consideration is how often to re-plan to generate new mission plans. In this work, re-planning is performed once for each primitive in the sequence, starting as the trim trajectory is reached.

To ensure that the planned trajectory remains continuous during re-planning, the re-planning computation must be completed prior to the conclusion of the trim trajectory to allow the tracking controller time to adjust to the new mission plan. In particular, let T_H be the time duration of each trim trajectory and let T_C be the time horizon of the tracking controller. The computation time T_j for re-planning during the j^{th} primitive in the sequence must satisfy $T_j < T_H - T_C$. When re-planning, the RRT algorithm is only constrained to finish performing the current primitive, and thereafter is free to choose any sequence of primitives to produce a feasible trajectory. In other words, the planner is only "committed" to performing the current primitive, and subsequent primitives in the sequence are "tentative," or might change, based on the result of re-planning. As soon as the planner generates a new mission plan, the "committed" trajectory is updated to include both the current primitive and the next primitive in the new mission plan. The "tentative" trajectory then begins after the second primitive in the new mission plan. This process repeats until the goal region is reached.

Figure 4.6 is included to provide an example of this timing convention. This example considers the case of the hybrid UAV powertrain, in which the planned power state trajectories $\overline{P}$ for the genset, battery, propulsion, and avionic subsystems are used as references for the tracking controller. For this example, $T_H = 20$s and $T_C = 5$s, leading to a re-planning time window of 15s. The timing of the first four re-planning runs is described below, with subsequent planning runs following the same pattern.

A. The initial mission plan is shown in the top left, where the system is already in a trim trajectory so re-planning starts immediately at 0s (shown by the vertical black line). Initially, the committed trajectory contains only the first primitive. The re-planning computation time T_1 must be less than the 15s re-planning window, shown by the gray region. The mission plan in plot A is valid until a new mission plan is found at T_1 seconds.

B. After T_1 seconds, the mission plan is updated as shown in plot B. In the second mission plan, a second primitive is appended to the committed trajectory, different from the second

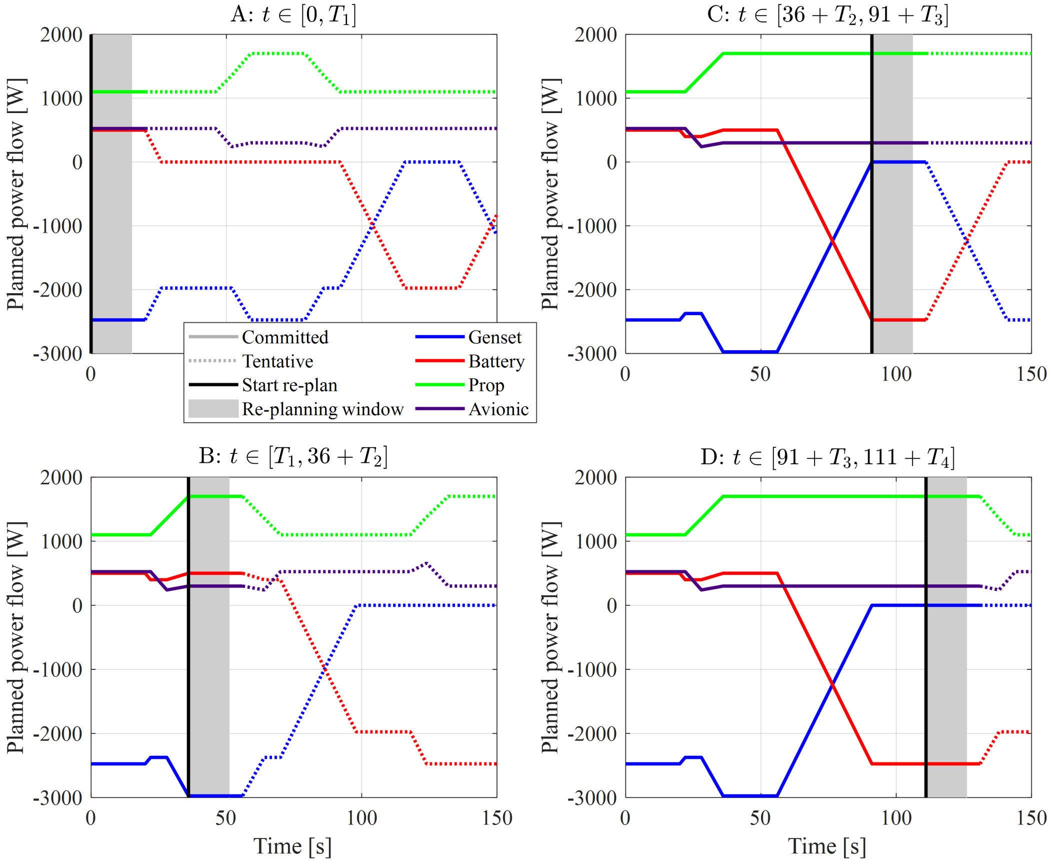


Figure 4.6: Example of re-planning timing for hybrid electric powertrain.

primitive of plot A which was tentative. The trim trajectory of the second primitive is reached at $t = 36$s, at which time RRT will run once more to generate a new mission plan. The mission plan shown in plot B remains valid until a new mission plan is found at $t = 36 + T_2$ seconds, where T_2 is the computation time for the second re-planning iteration.

C. After $t = 36 + T_2$ seconds, the mission plan is updated as shown in plot C. A third primitive is appended to the committed trajectory. At this point, the first primitive has been fully completed. The trim trajectory of the third primitive is reached at $t = 91$s, at which time the re-planner will again generate a new mission plan. The mission plan shown in plot C remains valid until a new mission plan is found at $t = 91 + T_3$ seconds.

D. After $t = 91 + T_3$ seconds, the mission plan is updated as shown in plot D. A fourth primitive is appended to the committed trajectory. At this point, the second primitive has been fully completed. The trim trajectory of the fourth primitive is reached at $t = 111$s, at which time the re-planner will run once more to generate a new mission plan. The mission plan shown in plot D remains valid until a new mission plan is found at $t = 111 + T_4$ seconds.

4.3.2. Re-planning with ERRT

Online re-planning using the basic RRT algorithm is not very efficient as it requires starting "from scratch" each time, neglecting information from previous iterations. Additionally, the distance metric does not consider the total cost from the root (i.e., first node) of the tree to the leaves, so the solution provided by RRT is widely known to be suboptimal. Improving upon these shortcomings is the execution-extended RRT (ERRT), one of the first variants of the basic RRT algorithm, which is tailored for online re-planning [82]. With respect to the basic RRT algorithm described in Section 4.2, ERRT makes modifications to the sampling strategy and distance metric as discussed below to leverage information from prior mission plans and improve the quality of mission plans.

4.3.2.1. Sampling strategy

ERRT leverages knowledge from previous planning iterations in the form of a waypoint cache. A fixed-size array of waypoints is maintained throughout the mission and updated with new waypoints each time a feasible mission plan is found. The waypoint cache is used to bias the tree towards these “desirable” points in the feasible space. In particular, the RandomState function has probability p_w of selecting a waypoint from the cache and probability $1-p_w$ of selecting a random state as before. This is described by the algorithm in Figure 4.7, where X_{WP} is a fixed-size matrix of waypoints saved from past mission plans.

After each run of ERRT, a set of waypoints is generated consisting of each node in the mission plan. These waypoints are inserted into the matrix of past waypoints X_{WP} with random replacement to be used in the next run of the re-planning algorithm.

```
RandomState($X_{WP}$, $p_w$ )
rand([0,1]) $\rightarrow p$
randi([0,length($X_{WP}$)]) $\rightarrow i$
If $p \leq p_w$
    $X_{WP}(i) \rightarrow x_{rand}$
Else
    $\mathcal{N}(\boldsymbol{\mu}, \Sigma) \rightarrow x_{rand}$
End
Return $x_{rand}$
```

Figure 4.7: Random state selection algorithm with ERRT. Modified from [82].

4.3.2.2. Distance metric

ERRT attempts to improve upon the quality of feasible missions plans by including in the distance metric a term quantifying the total cost from the root to the leaves. The modified distance metric is then a weighted sum of the pointwise distance and cumulative cost. For the IPPT systems considered in this work, a relevant cost function is the cumulative fuel consumption starting from the beginning of the mission. The associated distance metric is then given by

$$\rho(X_i, X_r) = \left\| X_i - X_{rand} \right\|_{\Lambda_X}^2 + \beta m_{fuel}(X_i) \tag{4.33}$$

where X_i is a node in the tree, X_{rand} is a state returned by the RandomState function, β is a scalar defining the importance of fuel consumption in the weighted sum, and $m_{fuel}(X_i)$ is the total amount of fuel consumed from the root of the tree to node X_i. The choice of weighting β has a significant impact on the algorithm's performance, as noted in [82], such that excessively large values of β can prevent the algorithm from finding feasible paths.

4.4. PARAMETERIZATION OF ENERGY PRIMITIVES

Nominal trajectories generated by the planning algorithms discussed in Sections 4.2-4.3 are dictated by energy primitives as defined in Section 4.2.1. To ensure that the true system state is feasible despite model error, condition (2.2) must be satisfied. Satisfaction of this condition is dependent on the performance of the tracking controller as well as the design of nominal trajectories, through appropriate parameterization of energy primitives, that are feasible for the system and easily tracked by the controller.

As mentioned in Section 4.2.1, energy primitives in this work consist of several phases. In the trim trajectory of each primitive, the system remains in a constant power operating point for a relatively long period of time. Phases preceding the trim trajectory are characterized by constant, nonzero power ramp rates, enabling a transition between the operating point of the previous trim trajectory and that of the current primitive. In this section, a process for defining desirable parameters for these operating points and transitions is discussed.

Note that, while each primitive in this work consists of a sequence of transitions followed by a trim trajectory, the sampling-based planning methods can be extended to include primitives that demonstrate more complex behaviors. Further, while a simple iterative approach to generating primitives is described here, more advanced approaches employing learning techniques might be used to reduce model mismatch between the primitives used in the planner and the true system [47], [83].

4.4.1. Operating points for trim trajectories

Because constant power operation during trim trajectories comprises the majority of system operation, operating points should correspond to desirable power states in which the system might remain for an extended period of time. Further, operating points must satisfy conservation of energy. The total energy (power) consumed, including that used by sink subsystems and dissipated through resistive elements, must equal the energy (power) provided by the sources. This requirement is stated below

$$\partial/\partial t\left(\sum_{i=1}^{N}\overline{E}^{i}+\overline{E}^{loss}\right)=0 \tag{4.34}$$

where $\overline{E}^{i}$ is the energy state of the i^{th} subsystem, N is the number of subsystems, and $\overline{E}^{loss}$ is the lumped energy lost to the environment. Note that (4.34) constrains the power states as well as ramp rates. The sign convention, described originally in Section 3.3, is such that a positive subsystem power state increases the respective subsystem's energy state.

Ensuring that an operating point satisfies (4.34) requires a reasonably accurate estimate of the system losses which may not be known *a priori*. In this approach, we begin with candidate operating points that may represent desirable regions of the planning space. These are defined using some initial estimate of the system losses, which may be inaccurate. The estimate of the system losses is then iteratively refined until the condition (4.34) is met within some threshold.

To generate candidate system-level operating points, one can begin by defining desirable operating modes for each subsystem. Desirable subsystem operating modes can be defined based on desirable properties of the system, operator experience, or some other heuristic. For the purposes of long-term planning, a desirable operating mode might achieve some high-level objective, such as operating the engine at a power level that minimizes fuel consumption. Alternatively, a desirable operating mode might be one that enables avoidance of constraints. For example, consider the motivating example in Figure 1.2, in which a hybrid electric UAV powertrain is required to traverse "quiet zones" with its engine off. To satisfy this constraint, at least one engine operating mode must operate the engine with zero power. Additionally, to prevent exhausting the battery, at least one battery operating mode should charge the battery.

After defining operating modes for each subsystem, and using some initial estimate of system losses, the subsystem operating modes can be combined in such a way that (4.34) is satisfied to generate candidate operating points for the full system. In general, (4.34) will not be satisfied if all subsystems are fixed to one of their predefined operating modes. To ensure satisfaction of (4.34), one can fix each subsystem to a predefined operating mode except for one or more 'free' subsystem(s). If the operating mode of a fixed subsystem i is given by $\overline{P}^{*i}$, the power state(s) of the 'free' subsystem(s) can be calculated as follows

$$\sum_{i \in free} \hat{P}^{free} = -\hat{P}^{loss} - \sum_{i \notin free} \overline{P}^{*i} \tag{4.35}$$

where $\hat{P}^{loss}$ is the current estimate of losses and $\hat{P}^{free}$ contains the power state(s) of the free subsystem(s). While all possible combinations of subsystem operating modes might be considered as candidate system-level operating points, this can lead to an arbitrarily large number of primitives, which reduces the planner's efficiency. For computational efficiency, it is advantageous to limit the number of candidate operating points to a small set that encompasses a wide range of operation.

Finally, to ensure (4.34) is satisfied, the loss estimate $\hat{P}^{loss}$ and corresponding free power state(s) $\hat{P}^{free}$ are iteratively refined. In this work, the following approach is taken to refine the estimates.

1. The full system model, given by (3.4) and (3.15), is simulated under the action of a tracking controller to steady state. This controller actuates the plant such that the true state X tracks the steady-state reference trajectories $\overline{X}$. Unlike the tracking control required to solve problem (2.3), the controller used here need not provide any guaranteed margin of tracking performance.
2. The closed-loop simulation results are used to calculate the resulting steady-state losses.
3. The loss estimate $\hat{P}^{loss}$ is updated using the simulated steady-state losses and corresponding free power state(s) $\hat{P}^{free}$ are updated according to (4.35).
4. This process repeats until some convergence criteria are met. Convergence criteria can consider reference tracking error and/or the change in $\hat{P}^{loss}$ from one iteration to the next.

4.4.2. Transitions

Transitions occur at the beginning of each primitive, enabling the system to move from one operating point to the next. In this work, transitions can consist of one or multiple phases, where each phase is defined by a fixed time duration and ramp rate (see Section 4.2.3.2). The time duration and ramp rate of each phase of the transition must be selected carefully to ensure the nominal power state trajectory given by (4.9) ends at the correct final state, which is the operating point as defined in the previous section. For primitive A^j, applied after primitive A^i, the time duration and ramp rate of each phase must satisfy the following

$$\sum_{k=0}^{N_R^j-1} R_k^j T_k^j = \overline{P}_{N_R^j}^j - \overline{P}_{N_R^i}^i \tag{4.36}$$

where R_k^j and T_k^j are the ramp rate and time duration, respectively, of phase $k+1$ of primitive j, $\overline{P}_{N_R^j}^j$ is the operating point of primitive A^j, and $\overline{P}_{N_R^i}^i$ is the operating point of the previous primitive A^i.

If the change in losses between two operating points is approximately linear, a transition can consist of a single phase. In this case, the ramp rate and time duration of this phase can be determined as follows

$$T_1^j = \max\left|\frac{\overline{P}_{N_R^j}^j - \overline{P}_{N_R^i}^i}{R^{max}}\right| \tag{4.37}$$

$$R_1^j = \frac{\overline{P}_{N_R^j}^j - \overline{P}_{N_R^i}^i}{T_1^j} \tag{4.38}$$

where R^{max} is a vector of maximum ramp rates for each subsystem.

Alternatively, transitions can consist of multiple phases. If the change in losses between two operating points is not approximately linear, a sequence of ramp values and time durations can be optimized to approximate the effects of the nonlinear losses.

4.5. SUMMARY

In summary, this chapter presented a novel implementation of sampling-based planning methods for long-term operation of IPPT systems to guarantee satisfaction of polytopic constraints that may change over time. Sampling-based planning methods efficiently compute feasible paths through complex regions by sampling the space, rather than using explicit geometric representations of all feasible nodes. Section 4.1 provided a brief overview of sampling-based planning methods. Section 4.2 discussed the implementation of the RRT algorithm for planning the operation of IPPT systems under model uncertainty by constructing sequences of energy primitives. Section 4.3 discussed methods for online re-planning using RRT and ERRT. Section 4.4 discussed the parameterization of energy primitives.

CHAPTER 5: ROBUST MODEL PREDICTIVE CONTROL

In this two-stage approach for safe operation of IPPT systems, robust tracking control methods are necessary to manage fast dynamics such that the true system achieves the planned trajectory within the predefined tracking error bound. While a simplified, reduced-order model of the system dynamics is used to enable computationally efficient long-term planning, the tracking controller focuses on accurate, short-term tracking of the mission plan. Because the tracking controller only considers short-term behavior, it can use a more complex, full-order model to ensure accurate tracking without incurring a significant computational burden.

A common and straightforward method for tracking reference trajectories is using a linear feedback controller such as a linear quadratic regulator (LQR). For example, in [37] LQR is used for tracking reference temperatures in an aircraft thermal management system. An LQR approach is also demonstrated in [84] for tracking of engine thrust and electrical power for a more electric aircraft power system. Yet these methods are subject to some significant limitations, which make them unsuitable to solve problem (2.3) which requires robust tracking of planned trajectories. Notably, these control strategies cannot directly incorporate constraints. Additionally, the gains used to parameterize these controllers are typically tuned at a single linearization point, such that the controller's tracking performance may degrade when the system moves away from the linearization point.

Alternatively, nonlinear control strategies can be used to prevent the controller's sensitivity to linearization, and in some cases an analytical bound of the error between the closed-loop system and reference can be obtained. For example, [85] used a nonlinear adaptive control strategy to regulate engine coolant temperature and, through Lyapunov analysis, bounded the error between the desired and actual temperature. However, the analysis methods used to derive nonlinear control strategies are specific to the system under study, and for highly complex systems these analyses may be intractable. In this study, planned trajectories are developed using reduced-order models that are abstracted from the full-order system dynamics, making it more difficult for such an analytical bound to be found.

Model predictive control (MPC) provides an optimization-based alternative to enable the consideration of constraints in addition to reference tracking. Examples of MPC approaches for reference tracking can be found in [27], [84] for electrified aircraft energy management, [38], [42], for electrified aircraft thermal management, and [44] for energy management of electrified ship power systems. Yet these MPC approaches typically use linear, discretized prediction models, subject to linearization error that can allow the true system to violate constraints. In contrast, robust MPC (RMPC) approaches account for model error to ensure that the true system remains feasible while tracking reference trajectories. RMPC approaches have successfully been demonstrated to enable robust tracking of power flow references [68] and temperature references [86] in hierarchical frameworks, assuming the system or power flows have a specific structure.

This chapter presents an RMPC formulation, similar to those of [68], [86], for tracking reduced-order reference trajectories to satisfy the constraints imposed by the predefined error bounds. This RMPC formulation enables the consideration of generic power flows under the assumption that their nonlinearity can be captured by the combination of a linear model and bounded uncertainty. The chapter is outlined as follows. Section 5.1 provides an overview of RMPC, beginning with a brief introduction of MPC. Sections 5.2 and 5.3 discuss the bounding of linearization error. Section 5.4 provides the generic RMPC formulation. Section 5.5 concludes the chapter.

5.1. OVERVIEW OF ROBUST MPC

In model predictive control, a model of system dynamics is used to predict future behavior in response to certain inputs over a finite time horizon. Using the predictive model, inputs can be optimized to achieve desired behavior which might include tracking of reference signals or low energy consumption. This optimization problem can be subject to constraints on the states, inputs, and outputs of the dynamic system to ensure feasibility. To mitigate the computational burden of solving such an optimization problem, many MPC formulations use linear, discrete prediction models of nonlinear system dynamics. When the optimization problem is solved, the first input in the optimized sequence is applied to the plant. The optimization problem is solved repeatedly at a fixed rate using updated measurements.

Robust MPC describes a class of MPC formulations that account for error between the true system and the predictive (i.e. nominal) model in order to ensure that the true system satisfies constraints. To do so, a common approach is to assume that the nonlinear system is governed by a linear system subjected to disturbances and apply a feedback control to regulate error between the nominal and true system [87]. The RMPC formulation used in this work, depicted in Figure 5.1, follows this approach. In this two-stage approach, reference trajectories from the planner are re-sampled at the controller update rate to yield references, denoted as $\bar{X}$ in Figure 5.1, for each step of the control horizon. The system model is linearized each time the controller updates, yielding a nominal model, denoted as $\bar{x}$ in Figure 5.1, of the full-order state dynamics that is used to optimize control decisions at each time step. The optimization problem aims to minimize error between the reference trajectory and the true system behavior. The optimal nominal states are passed as

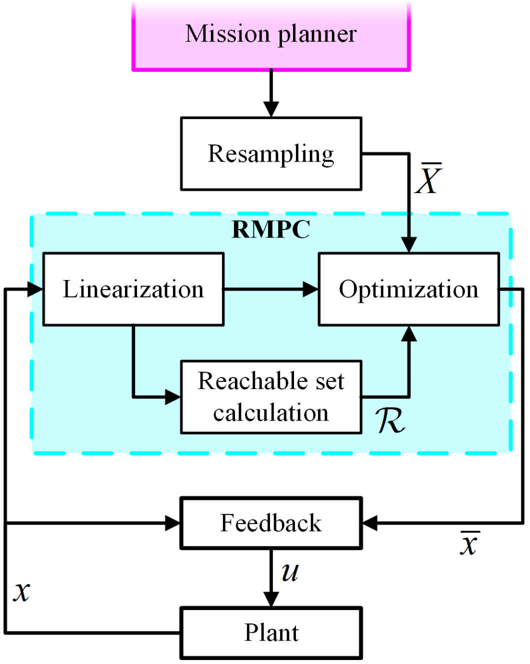


Figure 5.1: Overview of RMPC formulation used in this two-stage approach.

references to a feedback controller, which actuates the plant with input u such that the true full-order state x achieves the desired behavior. To bound the linearization error incurred using the nominal model, reachable sets, denoted as $\mathcal{R}$ in Figure 5.1, are calculated taking into account the behavior of the feedback controller. Constraints of the optimization problem are tightened to account for this linearization error. The following sections describe the methods used to bound linearization error and the controller formulation.

5.2. FULL-ORDER MODEL LINEARIZATION ERROR

To ensure that the true system satisfies constraints, RMPC considers the effects of linearization error between the nominal and true system. If linearization error can be bounded to a set, this set can be used to artificially tighten the constraints considered in RMPC. This section discusses bounding linearization error in the full-order system states and inputs following the approach of [87].

5.2.1. State linearization error

The nominal model of the full-order system state is given by

$$\dot{\overline{x}} = A\overline{x} + A_d d + B_1\overline{u} + B_2 \tag{5.1}$$

where upper bars are used to denote quantities associated with the nominal model. This model can be obtained by linearizing a graph-based model, as discussed in Section 3.2. Assume the true system state evolves according to

$$\dot{x} = Ax + A_d d + B_1 u + B_2 + h \tag{5.2}$$

where h is an uncertain term.

Rather than directly actuating the plant using the optimized control inputs from RMPC, feedback control is required to bound the error between the true and nominal states [87]. In particular, proportional-integral (PI) control is used in this work to reject disturbances that affect the plant at a higher frequency than that of RMPC. Let i, j be the index of the state regulated by the j^{th} control input. To regulate error between nominal state $\overline{x}_{i,j}$ and true state $x_{i,j}$, the PI control formulation for input u_j is given by the following

$$u_j = \overline{u}_j + K_P^j(\overline{x}_{i,j} - x_{i,j}) + K_I^j \int (\overline{x}_{i,j} - x_{i,j})dt \tag{5.3}$$

where K_P^j is the proportional gain and K_I^j is the integral gain corresponding to input u_j, and $\overline{u}_j$ is the nominal input.

To write the full vector of control inputs compactly, define $\tilde{x}$ as the error between the nominal model and true system as follows.

$$\tilde{x} = \overline{x} - x \tag{5.4}$$

Define e_I to contain the integrated error terms of (5.3) as follows

$$e_I = F\int \tilde{x}dt \tag{5.5}$$

where F is a matrix that selects elements of $\tilde{x}$. To define F, let vector $v_k \in \mathbb{R}^{1\times N_x}$ be defined such that its i^{th} element is given by the following.

$$v_k(i) = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Matrix F is given by the following

$$F = \begin{bmatrix} v_{i,1} \\ \vdots \\ v_{i,N_u} \end{bmatrix} \tag{5.7}$$

where N_u is the number of inputs. Define matrix K as

$$K = \left[diag(K_P^1,\ldots,K_P^{N_u})F \quad diag(K_I^1,\ldots,K_I^{N_u}) \right] \tag{5.8}$$

where the notation $diag(a_1,\ldots,a_N)$ is used to denote a diagonal matrix, whose diagonal entries are given by $a_1,\ldots,a_N$. The control input vector can be written as follows.

$$u = \overline{u} + K\begin{bmatrix} \tilde{x} \\ e_I \end{bmatrix} \tag{5.9}$$

Note that the discussion above assumes that all feedback controllers are PI, but other feedback control strategies, such as LQR, can be used to define the matrix K.

Combining (5.9) with (5.1), (5.2), and (5.5), the linearization error and integrated error dynamics are given by

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{e}_I \end{bmatrix} = A_R\begin{bmatrix} \tilde{x} \\ e_I \end{bmatrix} + B_R h \tag{5.10}$$

where A_R, B_R are given below.

$$A_R = \begin{bmatrix} \begin{bmatrix} A & 0^{N_x \times N_u} \end{bmatrix} - B_1 K \\ F \quad 0^{N_u \times N_u} \end{bmatrix} \tag{5.11}$$

$$B_R = \begin{bmatrix} -I^{N_x} \\ 0^{N_u \times N_x} \end{bmatrix} \tag{5.12}$$

The notation $0^{n\times m}$ is used to denote an n-row, m-column matrix of zeros and I^n denotes the size n identity matrix.

The uncertain term is assumed bounded such that $h \in \mathcal{H}$, where $\mathcal{H}$ is a compact set. While the exact computation of this bound may be intractable for complex IPPT systems, $\mathcal{H}$ can be approximated by sampling states, inputs, and disturbances in a neighborhood of the linearization point and computing the maximum deviation between the nominal model (5.1) and the nonlinear model (3.5) for each sample. Given a set $\mathcal{H}$, reachability computations can be used to bound the propagation of linearization error over time. For discrete-time RMPC formulations, a discrete-time model of error propagation can be used to obtain linearization error reachable sets at each timestep of the controller. To this end, (5.10) can be discretized to yield

$$\begin{bmatrix} \tilde{x}_{k+1} \\ e_{I,k+1} \end{bmatrix} = A_R^z \begin{bmatrix} \tilde{x}_k \\ e_{I,k} \end{bmatrix} + B_R^z h_k \tag{5.13}$$

where subscript k is used to indicate values associated with timestep k and A_R^z, B_R^z are the discretized matrices corresponding to (5.11) and (5.12) respectively.

Given a discrete-time model of error propagation such as (5.13), a common approach to bound linearization error in RMPC is to compute the disturbance invariant set [87]. A disturbance invariant set is a compact set S satisfying the following.

$$A_R^z s + B_R^z h \quad \text{for all} \quad s \in S, h \in \mathcal{H} \tag{5.14}$$

If K is selected such that A_R^z is stable, then there exists at least one disturbance invariant set for (5.13) [88]. While a disturbance invariant set provides a bound for linearization error, its use in an RMPC formulation is subject to some challenges. First, methods for computing such a set can be computationally intensive, requiring recursive calculations [89]. The nonlinear, IPPT systems considered in this work benefit from repeated linearization, and online calculation of disturbance invariant sets for each linear system model is intractable. Second, conservatism is introduced when disturbance invariant sets are used to tighten constraints in RMPC. This is because disturbance invariant sets bound model error for all time, while an RMPC controller considers only a finite time horizon in which model error needs to be bounded [90].

Rather than using disturbance invariant sets to bound model error for all time, this work adopts the approach of [90] in which linearization error reachable sets are used to bound linearization error at each step of the control horizon. The reachable set for linearization error at timestep $k+1$ is given by

$$\mathcal{R}_{k+1} = A_R^z \mathcal{R}_k + B_R^z \mathcal{H} \tag{5.15}$$

where the initial linearization error set is $\mathcal{R}_0$. These sets can be efficiently computed online using computational tools such as CORA [91]. The reachable set $\mathcal{R}_k$ can be partitioned as follows to separate full-order state error terms from integrator error terms.

$$\mathcal{R}_k = \begin{bmatrix} \mathcal{R}_k^x \\ \mathcal{R}_k^I \end{bmatrix} \tag{5.16}$$

Error reachable sets are used in this work to tighten constraints. Satisfaction of a constraint $x \in \mathcal{X}$ for the true state is ensured if the nominal state satisfies the following tightened constraint.

$$\overline{x}_k \in \mathcal{X} - \mathcal{R}_k^x \tag{5.17}$$

Computing set differences such as this online can be challenging because the reachable set (5.15) can be geometrically complex. To reduce computational complexity, a sufficient yet conservative condition for (5.17) is given by

$$\overline{x}_k \in \mathcal{X} - box(\mathcal{R}_k^x) \tag{5.18}$$

where the function $box(\mathcal{A}) = \mathcal{I}\left(\left[A_{lb}, A_{ub}\right]\right)$ tightly over-approximates the set $\mathcal{A}$ as an interval set with lower bounds A_{lb} and upper bounds A_{ub}. Upper and lower bounds of the interval set $box(\mathcal{R}_k^x)$ are denoted by r_{ub}^x and r_{lb}^x, respectively.

Examples of full-order state linearization error reachable sets are provided in Figure 5.2 for the full-order states of the hybrid UAV powertrain. The states of this system correspond to: 1) battery state of charge SOC, 2-3) voltages V_1, V_2 of the battery equivalent circuit model (see [60]), 4) vehicle velocity v, 5) propulsion motor angular speed ω, 6) avionic load current I_{load}, and 7) genset current I_{gen}. The reachable sets (solid lines) and their interval over-approximations (dotted lines) are plotted for each step k of the control horizon. To display the seven-dimensional reachable sets, each set $\mathcal{R}_k^x$ is projected onto four two-dimensional axes corresponding to pairs of states, such that each axis $\tilde{x}_i$ corresponds to the linearization error of the i^{th} full-order state of the model. Note that axis $\tilde{x}_6$ appears twice because there is an odd number of states. For this system, feedback controllers regulate the motor angular velocity, load current, and genset current. The effect of this feedback control can be seen in Figure 5.2, as the projections of error reachable sets corresponding to these states shrink.

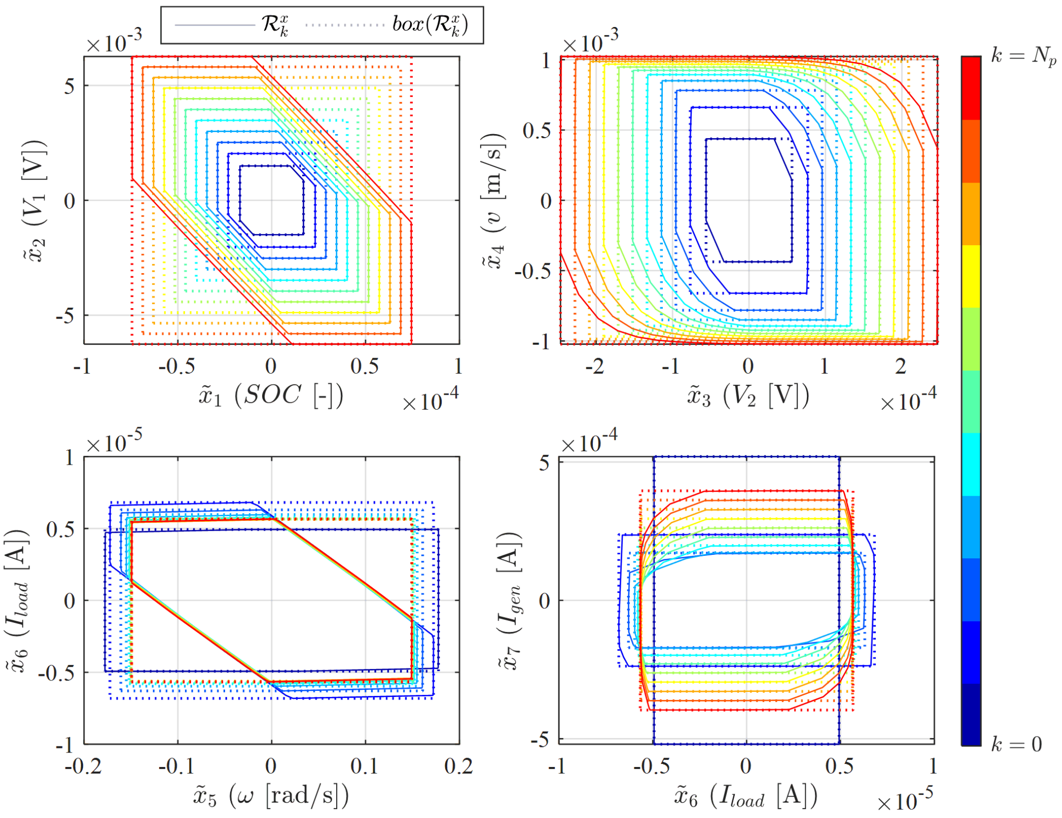


Figure 5.2: Examples of linearization error reachable sets for the seven full-order states of hybrid electric UAV powertrain. Reachable sets $\mathcal{R}_k^x$ and their interval over-approximations $box(\mathcal{R}_k^x)$ are plotted for each step k of the RMPC horizon of length $N_p = 10$.

5.2.2. <u>Input linearization error</u>

Inputs of the system (5.2) may be subject to constraints. Linearization error for inputs must also be bounded to ensure satisfaction of input constraints. From (5.9), error $\tilde{u}_k = \overline{u}_k - u_k$ is given by

$$\tilde{u}_k = K \begin{bmatrix} \tilde{x}_k \\ e_{I,k} \end{bmatrix} \tag{5.19}$$

and can be bounded to the following reachable set.

$$\mathcal{R}_k^u = K\mathcal{R}_k \tag{5.20}$$

5.3. REDUCED-ORDER MODEL LINEARIZATION ERROR

In order to solve problem (2.3), it is necessary for RMPC to satisfy constraint (2.2) by ensuring that the true reduced-order states track those generated by the planner within a predefined error bound. In (2.2), the error bound $\mathcal{X}_{w}$ is constructed of error bounds for the energy and power states of each subsystem, which may take one of two forms depending on the subsystem type as discussed in Section 4.2.3.2. For type 1 subsystems, the power state error is assumed to be bounded, while energy state error is assumed to be bounded for type 2 subsystems. RMPC uses a nominal model of the power states for type 1 subsystems and a nominal model of the energy states for type 2 subsystems. To ensure that the energy and power state error bounds are satisfied, RMPC tightens these constraints by calculating the deviation between the nominal and true values of the reduced-order energy and power states.

5.3.1. Power states

If subsystem j is of type 1, the tracking error bound constraint (2.3) takes the form of (4.13) , which requires the tracking error between the planned power state and true power state to be bounded. RMPC uses a nominal (i.e., linearized) model to enforce the power state tracking error bound (4.13) for all type 1 subsystems. The reader is referred to Section 4.2.3.2 for a description of the two types of subsystems. Because the reduced-order power states are given by nonlinear functions (3.15), the nominal model incurs linearization error. The power state linearization error is bounded using error reachable sets, which are then used to tighten the constraint (4.13) ensuring feasibility despite linearization error.

Power states are treated as outputs of the dynamic system (5.1). The nominal power state model is given by

$$\hat{P}^{\mathrm{I}} = C\overline{x} + C_d d + D_1\overline{u} + D_2 + \hat{v} \tag{5.21}$$

where superscript I is used to denote those power states corresponding to type 1 subsystems, $\hat{v}$ is an estimate of nonlinearity, $C = \tilde{M}F_1$, $C_d = \tilde{M}F_2$, $D_1 = \tilde{M}F_3$, $D_2 = \tilde{M}F_4$, $\tilde{M}$ is defined as in (3.16) , and F_1, F_2, F_3, and F_4 are defined as in (3.14). The estimate of nonlinearity is included in (5.21) to eliminate output reference tracking offsets due to steady-state nonlinearities [92], [93]. The estimate can be obtained using a first-order filter as follows

$$\hat{v}_k = K_{v1}\hat{v}_{k-1} + K_{v2}\left(\hat{P}_{k-1} - P_{k-1}\right) \tag{5.22}$$

where K_{v1}, K_{v2} are estimator gains selected to yield a stable estimator, $\hat{P}_{k-1}$ is the power state prediction obtained from (5.21) at the previous timestep, and P_{k-1} is the true power state at the previous timestep. The true power states are assumed to be given by the following model

$$P = Cx + C_d d + D_1 u + D_2 + v \tag{5.23}$$

where v captures nonlinear terms and is treated as an uncertain term. Error between the nominal and true power state model is given by the following.

$$\tilde{P} = \left(\begin{bmatrix} C & 0^{N_P \times N_x} \end{bmatrix} - D_1 K\right)\begin{bmatrix} \tilde{x} \\ e_I \end{bmatrix} + \hat{v} - v \tag{5.24}$$

Inaccuracy of the steady-state nonlinearity estimate is assumed to be bounded such that $\hat{v} - v \in \mathcal{V}$ for a compact set $\mathcal{V}$. Note that analytical derivation of a tight bound $\mathcal{V}$ would require restrictions on the nonlinear term v. Rather, a conservative over-approximation of $\hat{v} - v$ can be found through appropriate design of the estimator (5.22) and simulation-based verification. This over-approximation can be used to define the set $\mathcal{V}$. Reachable sets for power state error are then given by the following.

$$\mathcal{R}_k^P = \left(\begin{bmatrix} C & 0^{N_P \times N_x} \end{bmatrix} - D_1 K \right) \mathcal{R}_k + \mathcal{V} \tag{5.25}$$

5.3.2. Energy states

If subsystem j is of type 2, constraint (2.3) takes the form of (4.15). This requires the tracking error between the planned and true energy state of subsystem j to be bounded. A nominal, linearized model is used to predict the energy states. Energy state linearization error must be bounded in order to tighten constraint (4.15) to ensure its satisfaction.

As mentioned in Section 3.3, each type 2 subsystem is associated with a vertex such that the energy state can be derived from the state variable. The nominal energy state of subsystem i associated with state variable x_j is given by the following

$$\hat{E}_i^{\text{II}} = C_j \overline{x}_j \tag{5.26}$$

where C_j is the capacitance of vertex j of the graph model, and the superscript II is used to denote energy states of type 2 subsystems. Note that in this work (5.26) does not require an estimate of nonlinearity, as in (5.21), because energy states are linearly related to the states of the graph model. The vector of true energy states of type 2 subsystems is given by the following,

$$\hat{E}^{\text{II}} = C_E x \tag{5.27}$$

where C_E is a matrix containing capacitances of the state variables associated with these energy states, and x is the true state vector.

Reachable sets describing the linearization error for energy states can then be calculated as follows.

$$\mathcal{R}_{k+1}^E = C_E \mathcal{R}_k \tag{5.28}$$

5.4. CONTROLLER FORMULATION

In this work, RMPC tracks references for the reduced-order model computed by the mission planner while satisfying constraints. Constraints apply to both the full-order and reduced-order models. For the full-order model, states and inputs are constrained to interval sets as given below.

$$x \in \mathcal{I}\left(\left[x_{lb}, x_{ub}\right]\right) \tag{5.29}$$

$$u \in \mathcal{I}\left(\left[u_{lb}, u_{ub}\right]\right) \tag{5.30}$$

For the reduced-order model, constraints ensure that the energy and power states remain near the reference trajectory from the planner within some error bound defined by (4.13) for power states and (4.15) for energy states. RMPC accounts for linearization error, as illustrated in Figure 5.3, to ensure satisfaction of these constraints. Note that the tracking error bound $\mathcal{X}_{\mathcal{W}}$ for the reduced-order states is an interval set (see (4.17)). With a slight abuse of notation, here we denote by L_k and $-L_k$ the upper and lower bounds of this interval set, respectively, at timestep k. For a type 1 subsystem j, the tracking error bound constraint is given below for the respective power state.

$$P_k(j) \in \bar{P}_k(j) + \mathcal{I}\left(\left[-L_k(j), L_k(j)\right]\right) \tag{5.31}$$

Similarly, for a type 2 subsystem j, the tracking error bound constraint is given below for the respective energy state.

$$E_k(j) \in \hat{E}_k(j) + \mathcal{I}\left(\left[-L_k(j), L_k(j)\right]\right) \tag{5.32}$$

To solve problem (2.3), the RMPC cost function penalizes tracking error between the planned trajectory and nominal states of the reduced order model, constraint violation, and changes in states and inputs between consecutive steps in the control horizon. The cost function is given below

$$J = \sum_{k=0}^{N_p-1}\left(\left\|\bar{X}_k - \hat{X}_k\right\|^2_{\Lambda_X} + \left\|s_k^X\right\|^2_{\Lambda_{s^X}} + \left\|s_k\right\|^2_{\Lambda_s} + \left\|\bar{x}_{k+1} - \bar{x}_k\right\|^2_{\Lambda_{dx}} + \left\|\bar{u}_k - \bar{u}_{k-1}\right\|^2_{\Lambda_{du}}\right) \tag{5.33}$$

where N_p is the number of timesteps in the prediction horizon. The notation $\|x\|_A^2$ is used to denote the weighted norm $x'Ax$. The first term in the cost function is the norm, weighted by positive semidefinite matrix Λ_X, of error between the planned trajectory $\bar{X}_k$ and nominal trajectory $\hat{X}_k$ at timestep k. The second term is the norm, weighted by positive semidefinite Λ_{s^X}, of slack variable s_k^X corresponding to violation of constraints for the nominal reduced-order state $\hat{X}_k$. Third is the Λ_S-weighted norm of slack variable s_k corresponding to violation of constraints for the nominal full-order state x_k. Fourth is the Λ_{dx}-weighted norm of changes in the nominal full-

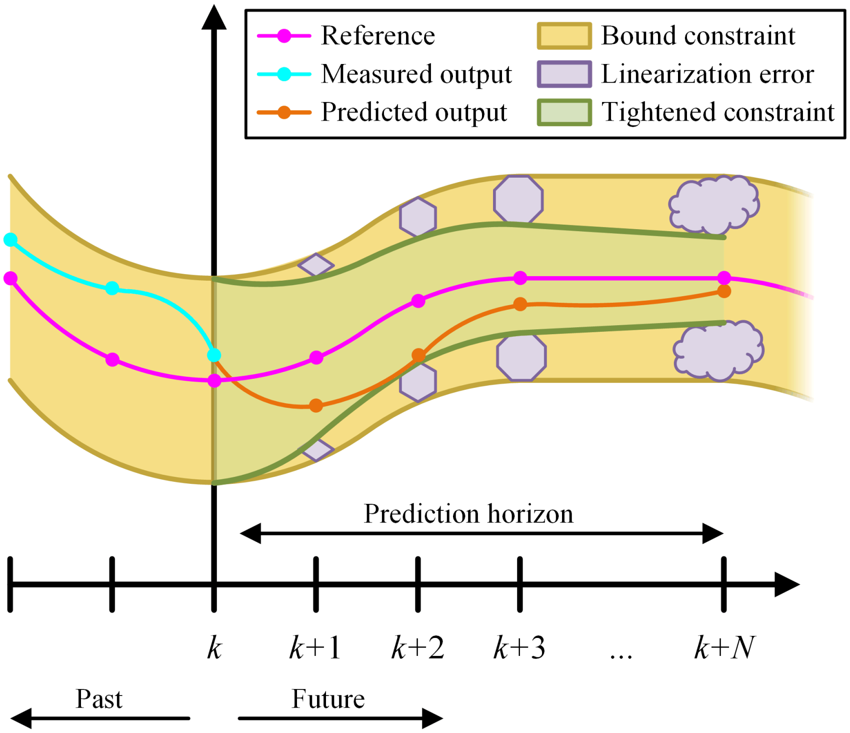


Figure 5.3: Illustration of the RMPC formulation used in this work. Reachable sets describing the propagation of linearization error (purple) are calculated and used to tighten the bound constraints (yellow). To ensure the bound constraints are satisfied, the nominal model (orange) is subjected to these tightened constraints, shown in green.

order state between timesteps k and $k-1$. The final term is the Λ_{du}-weighted norm of changes in inputs between timesteps k and $k-1$.

The optimization problem defining the RMPC formulation is given below.

$$
\begin{array}{rll}
\min\limits_{u,s,s^X} & J & (5.34) \\
\text{s.t.} & & \\
\text{for all } k = 0,\dots,N_p - 1: & \bar{x}_{k+1} = A^z \bar{x}_k + A_d^z d_k + B_1^z \bar{u}_k + B_2^z & \text{(a)} \\
& \hat{P}_k = C\bar{x}_k + C_d d_k + D_1 \bar{u}_k + D_2 + v_k & \text{(b)} \\
& \hat{E}_k^{\mathrm{II}} = C_E \bar{x}_k & \text{(c)} \\
& x_{lb} - r_{lb,k}^x - s_{k+1} \leq \bar{x}_{k+1} \leq x_{ub} - r_{ub,k}^x + s_{k+1} & \text{(d)} \\
& \left.\begin{array}{l} \hat{P}_k^I(j) \geq \bar{P}_k^I(j) - L_k(j) - r_{lb,k}^P(j) - s_k^X(j) \\ \hat{P}_k^I(j) \leq \bar{P}_k^I(j) + L_k(j) - r_{ub,k}^P(j) + s_k^X(j) \end{array}\right\} \text{for type 1 subsystems } j & \text{(e)} \\
& \left.\begin{array}{l} \hat{E}_k^{II}(j) \geq \bar{E}_k^{II}(j) - L_k(j) - r_{lb,k}^E(j) - s_k^X(j) \\ \hat{E}_k^{II}(j) \leq \bar{E}_k^{II}(j) + L_k(j) - r_{ub,k}^E(j) + s_k^X(j) \end{array}\right\} \text{for type 2 subsystems } j & \text{(f)} \\
& u_{lb} - r_{lb}^u \leq \bar{u}_k \leq u_{ub} - r_{ub}^u & \text{(g)} \\
& v_{k+1} = v_k & \text{(h)} \\
& s_k \geq 0 & \text{(i)} \\
& s_k^X \geq 0 & \text{(j)} \\
& v_0 = \hat{v}_0 & \text{(k)} \\
& \bar{u}_0 - \delta_u \leq \bar{u}_k \leq \bar{u}_0 + \delta_u & \text{(l)} \\
& \bar{x}_0 = x_0 & \text{(m)} \\
& \bar{E}_0 = E_0 & \text{(n)} \\
& \bar{u}_0 = \bar{u}_{-1} & \text{(o)}
\end{array}
$$

In problem (5.34), constraints (a), (b), and (c) give the nominal, discretized models of the full-order states, reduced-order power states, and reduced-order energy states, respectively. Constraint (d) imposes state constraints (5.29) that are tightened using the upper and lower bounds r_{ub}^x and r_{lb}^x of the interval approximation of $\mathcal{R}_k^x$. To prevent the optimization problem from losing feasibility, constraint (d) is imposed as a softened constraint through the introduction of nonnegative slack variable s_k (see constraint (i)). Positive values of s_k enable the nominal state $\overline{x}_k$ to move outside of its tightened constraint. This is discouraged by penalizing positive slack variable values in the cost function (5.33). Constraint (e) enforces the tracking error bound constraint (5.31) for type 1 subsystems. Acting on the nominal power state, this constraint is tightened using an interval approximation of the power state error reachable set, $box(\mathcal{R}_k^P) = \mathcal{I}\left(\left[r_{lb}^P, r_{ub}^P\right]\right)$. Constraint (f) enforces constraint (5.32) for type 2 subsystems, tightened using an interval approximation of the energy state error reachable set, $box(\mathcal{R}_k^E) = \mathcal{I}\left(\left[r_{lb}^E, r_{ub}^E\right]\right)$. Constraints (e) and (f) are also enforced as soft constraints through the introduction of nonnegative slack variable s_k^X (see constraint (j)), which is heavily penalized in the cost function (5.33). Constraint (g) imposes input constraints (5.30) as tightened constraints using the interval approximation of the input error reachable set, $box(\mathcal{R}_k^u) = \mathcal{I}\left(\left[r_{lb}^u, r_{ub}^u\right]\right)$. Constraint (h) ensures that the estimate of power state nonlinearity remains at its initial value, given in constraint (k), over the control horizon. Constraints (i) and (j) ensure the slack variables are nonnegative. Constraint (l) prevents the system from moving far from the linearization point by constraining inputs to remain within a δ_u-neighborhood of the initial input. Constraints (m) and (n) initialize the nominal full-order states and energy states at their current measured values. Constraint (o) implements a single time step delay in input application that accounts for the computational delay resulting from solving the problem (5.34).

Note that additional constraints may be incorporated in (5.34) to achieve desired behaviors depending on the currently active energy primitive. In particular, in each of the case studies that follow, some energy primitives are defined to turn off an engine. Turning off an engine requires the control input associated with the engine to be identically zero for the duration of the energy primitive. Let i be the index of the input corresponding to the engine that is turned off. Then, constraint (g) of (5.34) is modified as follows

$$u_i \in \mathcal{U}_i \tag{5.35}$$

where $\mathcal{U}_i$ is given by

$$\mathcal{U}_i = \begin{cases} \{0\} & \text{if } \sigma_i = 0 \\ [u_{lb}(i) - r_{lb}^u(i), u_{ub}(i) - r_{ub}^u(i)] & \text{otherwise} \end{cases} \tag{5.36}$$

and $\sigma_i \in \{0,1\}$ is a binary variable associated with each primitive dictating whether the engine mode is on ($\sigma_i = 1$) or off ($\sigma_i = 0$).

Problem (5.34) is quadratic and can be efficiently solved using a variety of numerical optimization solvers, including MATLAB's quadprog solver [94] or Gurobi [95]. For ease of implementation and speed of numerical computation, the generic setup of problem (5.34) can be pre-compiled into an object or function that requires only the parameters of the specific problem instance (e.g., linear state matrices, error sets, and initial state values at the current time instance) in order to solve the problem. This can be done, for example, using YALMIP [96].

5.5. SUMMARY

This chapter presented an RMPC formulation that is generic to IPPT systems enabling the solution of problem (2.3). The RMPC formulation uses a nominal, or linearized, model of the full-order and reduced-order system dynamics to optimize the system's reference tracking behavior. To ensure robustness to model error, Sections 5.2 and 5.3 discuss methods to bound the deviation between the nominal model and the true system for both the full- and reduced-order models.

Section 5.3 provides the generic RMPC formulation, which uses these bounds to tighten constraints on full-order states, inputs, and reduced-order states.

CHAPTER 6: CASE STUDIES

In this chapter, three case studies are presented to demonstrate the fulfillment of research objectives outlined in Section 1.3. Focusing on aerospace and marine applications, these case studies apply the two-stage approach to ensure safe operation of shipboard power systems and hybrid electric aircraft powertrains and thermal management systems. In each case study, the two-stage approach is shown to achieve the desired characteristics of rapid planning in nonconvex feasible regions to assure constraint satisfaction. It is shown that the two-stage approach adeptly manages multi-timescale dynamics and is scalable to highly complex systems.

This chapter is organized as follows. Section 6.1 discusses the first case study, in which simulation results demonstrate safe energy management of a shipboard power system, including offline planning and online re-planning to adapt to time-varying mission information. In Section 6.2, a second simulation-based case study demonstrates the extension of the approach to enable simultaneous management of energy and thermal systems of a hybrid UAV power, propulsion, and thermal system. In Section 6.3, the final case study demonstrates the application of this method for real-time operation of energy systems through experimental validation of the two-stage approach for safe operation of a hybrid UAV powertrain testbed. This case study includes experimental and simulation results for offline planning and online re-planning to adapt to changing mission information. Section 6.4 concludes the chapter with a discussion and summary of the results.

6.1. SHIPBOARD POWER SYSTEM

In this case study we consider operation of an integrated shipboard power system, described in Section 3.4.3, traveling between ports. Yielding a 20-dimensional planning problem, this case study demonstrates the scalability of the two-stage method for rapid planning of highly complex IPPT systems. To demonstrate the two-stage approach, simulation demonstrations of sampling-based methods are provided both for offline planning and for online re-planning in the presence of uncertain, varying constraints. Both scenarios use RMPC to track the planned trajectories. The parameters of the simulation model, including validation results, are provided in Appendix A.1.

6.1.1. Sampling-based planning problem setup

A brief overview of the setup for the SPS sampling-based planning problem is given below. Additional parameters used to solve this planning problem can be found in Appendix A.2.

6.1.1.1. Reduced order model

The ten subsystems of this system are listed in Table 6.1, including the edges of the graph model shown in Figure 3.7 corresponding to the power state of each subsystem. Note that each subsystem of the SPS is of type 1. The reduced order model for this system consists of ten energy states and ten power states, corresponding to a 20-dimensional planning problem. In Table 6.1, an asterisk is used to denote those subsystems for which the power state polarity is opposite to the corresponding graph model power flow (i.e., the corresponding entry of the incidence matrix $\tilde{M}$ in (3.16) is negative).

6.1.1.2. Mission specifications

Operational constraints for the SPS ensure safe operation of the battery packs and engines. As discussed in Section 4.2, each constraint $\mathcal{C}_j$ represents a forbidden region of the space. The first four operational constraints ensure that the energy state of each genset remains positive, such that the engines do not run out of fuel.

Table 6.1: Description of SPS subsystems.

Subsystem number	*Subsystem description*	*Subsystem type*	*Edge(s) corresponding to subsystem power state*
1	Main genset 1	1	9
2	Aux genset 1	1	10
3	Battery pack 1	1	1*
4	Propulsion system 1	1	65, 67
5	Hotel load 1	1	34
6	Main genset 2	1	9
7	Aux genset 2	1	10
8	Battery pack 2	1	40*
9	Propulsion system 2	1	68, 69
10	Hotel load 2	1	60

$$\mathcal{C}_j = \{X : X_{i(j)} \leq 0\}, j \in \{1, \ldots, 4\},\ i = [1, 2, 6, 7] \tag{6.1}$$

The following operational constraints provide a lower bound for the energy states of each battery pack, such that the battery packs are not discharged beyond their capacity.

$$\mathcal{C}_{4+j} = \{X : X_{i(j)} \leq 3600 Q V_{ocv}(0.3)\},\ j \in \{1, 2\},\ i = [3, 8] \tag{6.2}$$

In (6.2), Q is the battery capacity in amp-hours and $V_{ocv}(SOC)$ is the battery open circuit voltage as a function of SOC. Finally, the following operational constraints provide an upper bound for the energy states of each battery pack, such that the battery packs are not charged beyond their capacity.

$$\mathcal{C}_{6+j} = \{X : 3600 Q V_{ocv}(0.8) \leq X_{i(j)}\},\ j \in \{1, 2\},\ i = [3, 8] \tag{6.3}$$

In this case study, the region surrounding each port is subject to environmental restrictions forbidding high speeds and emissions. An illustration of this scenario is provided in Figure 6.1. Emissions and speed restrictions are considered as task-specific constraints, which are enforced within one nautical mile of each port. The ports are separated by 20 nautical miles. To consider these location-based constraints in the planner, which reasons only about energy and power states,

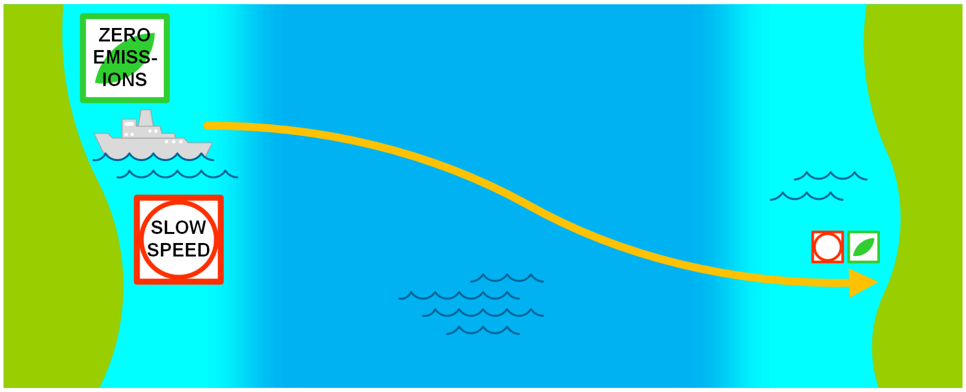


Figure 6.1: Illustration of emissions and speed restrictions, yielding task-specific constraints for SPS planning scenarios.

a linear mapping is used to estimate the total propulsion energy required to travel these distances. The total propulsion energy is obtained by summing states X_4 and X_9, which are the energy states of propulsion subsystems 1 and 2 respectively.

The emissions constraints for the departure port are given as follows

$$\begin{gathered} \mathcal{C}_{8+j} = \left\{ X : 0\text{MJ} \le X_4 + X_9 \le 150\text{MJ},\, X_{i(j)} \le -0.07\text{MW} \right\}, \\ j \in \{1,\ldots,4\}, i = [11,12,16,17] \end{gathered} \tag{6.4}$$

where X_{11}, X_{12}, X_{16}, and X_{17} are the power states of main genset 1, aux genset 1, main genset 2, and aux genset 2, respectively. The speed constraint for the departure port is given by

$$\mathcal{C}_{13} = \left\{ X : 0\text{MJ} \le X_4 + X_9 \le 150\text{MJ}, 0.3\text{MW} \le X_{14} + X_{19} \right\} \tag{6.5}$$

where X_{14} and X_{19} are the power states of propulsion subsystems of zones 1 and 2 respectively. For the arrival port, the emissions constraints are given by

$$\begin{gathered} \mathcal{C}_{13+j} = \left\{ X : 2850\text{MJ} \le X_4 + X_9,\, X_{i(j)} \le -0.07\text{MW} \right\}, \\ j \in \{1,\ldots,4\}, i = [11,12,16,17] \end{gathered} \tag{6.6}$$

where 2850MJ is an estimate of the total propulsion energy required to travel 19 nautical miles. Hence, these emissions constraints restricting the genset power states only become active when enough energy has been provided to the propulsion subsystems to approach the arrival port. The speed constraint is given by the following.

$$\mathcal{C}_{18} = \left\{ X : 2850\text{MJ} \le X_4 + X_9, 0.3\text{MW} \le X_{14} + X_{19} \right\} \tag{6.7}$$

The goal region is described by the propulsion energy required to reach the second port as follows.

$$\mathcal{X}_G = \left\{ X : 3000\text{MJ} \le X_4 + X_9 \right\} \tag{6.8}$$

The initial condition of the reduced order model, $X_0 = [E_0^T \quad P_0^T]^T$, is given below.

$$E_0 = [9.90 \quad 4.90 \quad 2.12 \quad 0 \quad 0 \quad 9.90 \quad 4.90 \quad 2.12 \quad 0 \quad 0]^T \times 10^3 \text{MJ} \tag{6.9}$$

$$P_0 = [0 \quad 0 \quad -64.4 \quad 25.2 \quad 25.0 \quad 0 \quad 0 \quad -64.4 \quad 25.2 \quad 25.0]^T \text{kW} \tag{6.10}$$

This initial energy state E_0 corresponds to full engine fuel capacity, 78% battery pack SOC, and zero energy use by the propulsion and hotel subsystems. The initial power state P_0 corresponds to low speed, low hotel power operation with all engines off.

6.1.1.3. Primitive parameterization

Trim trajectories for the SPS were generated by first defining a set of desirable operating modes for each component. These are summarized in Table 6.2 for all subsystems of zone 1. Operating modes for subsystems of zone 2 are defined identically. Modes for the main gensets include operating with the engine off, at max power, and at a fuel-optimal power rating. The fuel-optimal power rating is defined to correspond to 75% of the maximum engine power. Auxiliary genset modes include operating with the engine off and at max power. Battery pack operating modes charge the battery or sustain its charge by operating with zero power. The battery charging mode corresponds to a C/2 charge rate (i.e., fully charging the battery at this rate would take 2

Table 6.2. Operating modes for subsystems of the SPS.

Subsystem	*Operating modes*	*Descriptions*
Main genset 1	$P_{1a} = 0\text{MW}$	Engine off
	$P_{1b} = -1.16\text{MW}$	Fuel-optimal engine power
	$P_{1c} = -1.55\text{MW}$	Max engine power
Aux genset 1	$P_{2a} = 0\text{MW}$	Engine off
	$P_{2b} = -0.58\text{MW}$	Max engine power
Battery pack 1	$P_{3a} = 0\text{kW}$	Battery charge sustaining
	$P_{3b} = 28\text{kW}$	Battery charging
Propulsion system 1	$P_{4a} = 0\text{MW}$	Prop motor off
	$P_{4b} = 25.2\text{kW}$	Low speed (6 knots)
	$P_{4c} = 0.202\text{MW}$	Medium speed (12 knots)
	$P_{4d} = 6.83\text{MW}$	High speed (18 knots)
Hotel load 1	$P_{5a} = 10\text{kW}$	Low hotel demand
	$P_{5b} = 25\text{kW}$	Medium hotel demand
	$P_{5c} = 50\text{kW}$	High hotel demand

hours). Propulsion system operating modes are defined by the steady-state speed reached by the vehicle if both propulsion systems operate at the same mode. Propulsion modes include zero speed, low speed (6 knots), medium speed (12 knots), and high speed (18 knots). Low, medium, and high hotel power demands comprise the hotel operating modes, where the hotel loads operate at 110VDC.

From the operating modes provided in Table 6.2, a set of trim trajectories were selected following the process described in Section 4.4.1. To encompass a wide variety of operating regimes while preventing the number of trim trajectories from growing too large, only a subset of possible combinations of the operating points provided in Table 6.2 were considered. This subset of trim trajectories was truncated further to remove those that violated or nearly violated operational constraints, including battery charge rate constraints and genset maximum power constraints. The final set of 51 trim trajectories, listed in Table A.2 of Appendix A.2, includes 'symmetric' operating points, or those in which the power states of zone 1 are identical to those of zone 2, and 'asymmetric' operating points', in which power states of zone 1 are not identical to those of zone 2. Asymmetric operating points allow the engines and propulsion motors to operate at different modes in different zones while all other subsystems operate at the same mode. In each energy primitive, the time duration of the trim trajectory is 150s.

To fully define the energy primitives, transitions must be defined for every pair of trim trajectories. For many pairs of trim trajectories, the losses are nearly linear such that transitions consisting of a single phase with a constant ramp rate can be achieved within the tracking error bounds. This is done using (4.37), (4.38) with $R^{max} = 5\text{kW/s}$. However, due to significant nonlinearities in losses at propulsion power levels greater than P_{4b} (see Table 6.2), constant ramp rate transitions are not achievable for some pairs of trim trajectories. Particularly, if a pair of trim trajectories has P_{4i} and P_{4j}, respectively, as the power levels for propulsion system 1, then linear transitions are not achievable for $i \in \{c, d\}, j \neq i$ (see Table 6.2). Rather, piecewise constant sequences of ramp rates were used to approximate the nonlinear losses for propulsion power transitions $P_{4b} \to P_{4c}$ (i.e., low to medium propulsion power) and $P_{4c} \to P_{4d}$ (medium to high propulsion power). To define the ramp sequences, a genetic algorithm was used to minimize the error between the reference power state and the simulated power state under the action of a closed-loop tracking controller, as described by the following optimization problem

$$\min_{R_0,\ldots,R_{N_R-2},T_0,\ldots,T_{N_R-2}} \left(\max_{t \in T_{transition}} \left| \sum_{i \in 1,\ldots,N} \left(\overline{P}_i(t) - P_i(t) \right) \right| \right) \tag{6.11}$$

where R_k is the ramp rate of phase $k+1$ in the sequence, T_k is the time duration of phase $k+1$, N_R is the number of phases in the sequence, $T_{transition}$ denotes the time interval during which the system is transitioning between trims, N is the number of subsystems, $\bar{P}_i(t)$ is the reference power state for subsystem i at time t, and $P_i(t)$ is the simulated power state for subsystem i at time t, obtained using a closed-loop controller. The optimal ramp rate sequence for the low-medium prop power transition, consisting of $N_R = 3$ phases, is shown in Figure 6.2a). Figure 6.2b) shows the improvement in reference tracking using piecewise ramps. Similarly, the optimal ramp rate sequence for the medium-high prop power transition, consisting of $N_R = 4$ phases, is shown in Figure 6.3a) with improved reference tracking performance shown in Figure 6.3b). In each case, N_R was manually selected to balance complexity and accuracy. By approximating nonlinearities, the piecewise constant ramp rate sequences yield significant improvements in reference tracking error.

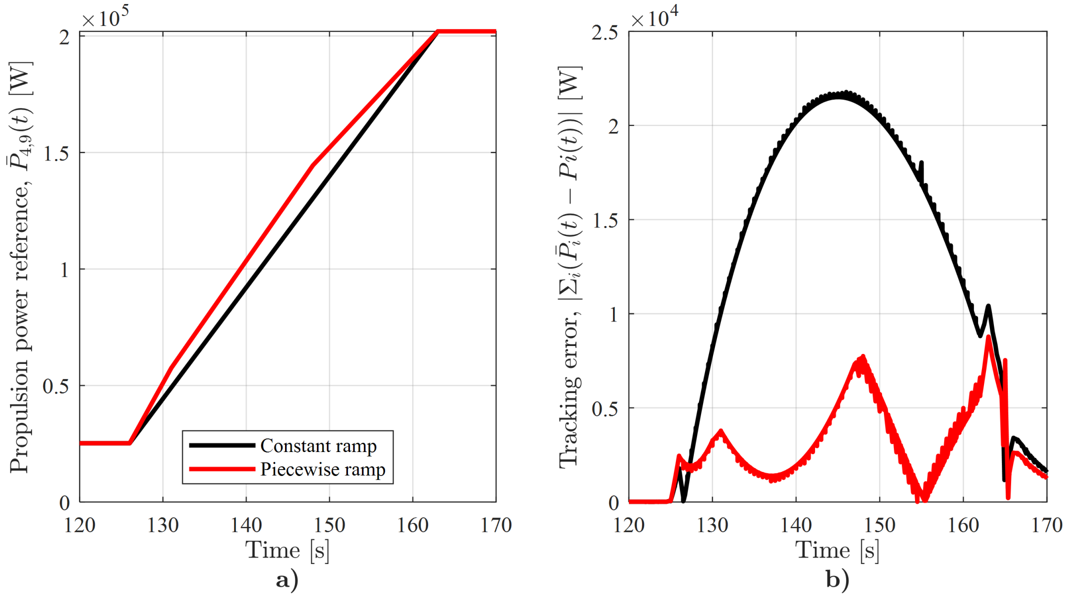


Figure 6.2: Optimized ramps for SPS low-medium propulsion power primitive transitions. Subplot a) shows the optimized ramps and b) shows the reduction in tracking error compared to constant ramp references.

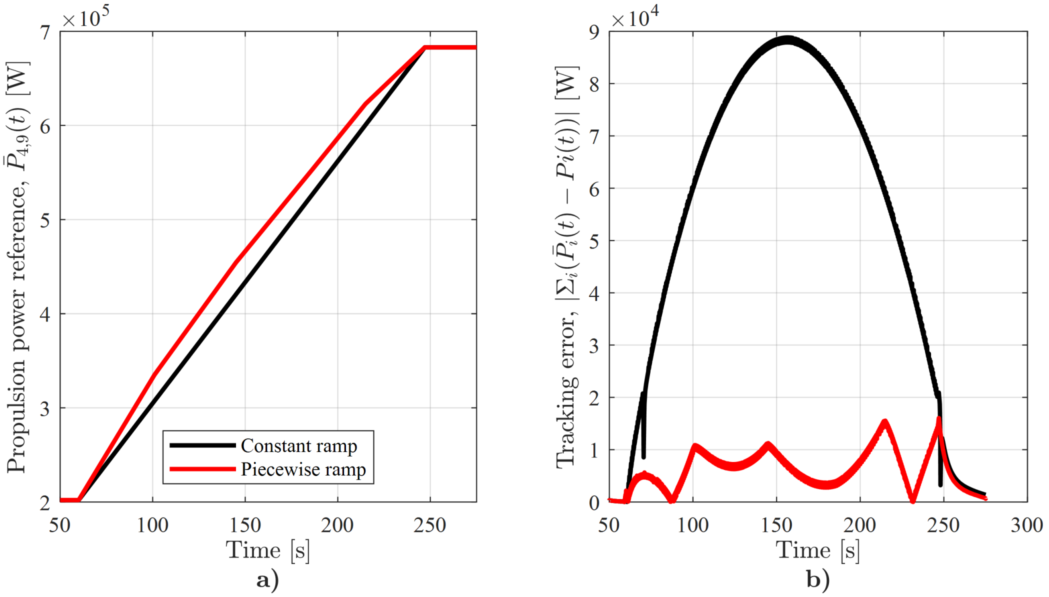


Figure 6.3: Optimized ramps for SPS medium-high propulsion power primitive transitions. Subplot a) shows the optimized ramps and b) shows the reduction in tracking error compared to constant ramp references.

All subsystems of the SPS are type 1, so the tracking error bounds for the power and energy states of each subsystem are given by (4.12)-(4.13), where $\mathcal{W}_k^i$ is the error bound for power states during stage k of primitive i. As noted in Section 4.2.3.2, $\mathcal{W}_k^i$ is an interval set which can be written as $\mathcal{W}_j^i = \mathcal{I}\left(\left[-L_k^i, L_k^i\right]\right)$. The vector L_k^i defining the upper and lower bounds of this interval set for stage k of primitive i is given in Appendix A.2.

6.1.2. RMPC problem setup

The RMPC nominal model is obtained by linearizing and discretizing the graph-based model shown in Figure 3.7. Feedback controllers for the SPS track state references for selected states of the full-order model. Table 6.3 lists the regulated state and control gains corresponding to each control signal. Note that the control gains corresponding to gensets are negative as the sign

Table 6.3: SPS feedback controller description.

Input description	*Regulated state*	K_P	K_I
Inverter 1 duty cycle	Prop 1 speed	2.0E-3	1.6E-2
DC-DC converter 1 duty cycle	Hotel 1 current	2.0E-2	2.0E-1
Inverter 2 duty cycle	Prop 2 speed	2.0E-3	1.6E-2
DC-DC converter 2 duty cycle	Hotel 2 current	2.0E-2	2.0E-1
Main genset 1 throttle	Main genset 1 current	-1.0E-3	-1.0E-2
Aux genset 1 throttle	Aux genset 1 current	-1.0E-3	-1.0E-2
Main genset 2 throttle	Main genset 2 current	-1.0E-3	-1.0E-2
Aux genset 2 throttle	Aux genset 2 current	-1.0E-3	-1.0E-2

convention is such that all genset currents are negative. Each input u_j is constrained by the following.

$$u_j \in [0,1] \tag{6.12}$$

Reachable sets describing error between the nominal model and true system, under the action of this feedback controller, are obtained following the processes described in Sections 5.2 and 5.3.

To ensure safe operation, RMPC applies constraints to certain states of the full-order model. Many of the safety-critical states, such as propeller speeds and genset currents, are ensured to operate in safe regimes through the design of the energy primitives as detailed in Section 6.1.1.3. On the other hand, following a planned trajectory may cause the battery states of charge to approach unsafe regimes due to error in the planning model. To ensure the batteries remain in a safe operating regime, the battery states of charge are constrained as follows.

$$x_{1,7} \in [0.3, 0.9] \tag{6.13}$$

Additional RMPC parameters, including objective function weights, update rate, and prediction horizon length, are listed in Appendix A.3.

6.1.3. Offline planning simulation results

This section provides a simulation-based demonstration of the two-stage method to plan operation of the SPS during travel between ports using the problem setup parameters described above. For this demonstration, all computations were performed on an Intel i7 processor with 32GB RAM. The RRT strategy described in Section 4.2 was used to plan a feasible trajectory using a custom MATLAB code. To solve the RMPC optimization problem described in Section 5.4, CORA [91] was used to calculate linearization error reachable sets, the YALMIP toolbox [96] was used to formulate the optimization problem, and Gurobi [95] was used to solve the optimization problem online.

Simulation results of the two-stage method for offline planning of the SPS demonstrate satisfaction of the task-specific constraints, shown in Figure 6.4. In both subplots of Figure 6.4, the horizontal axis is the sum of propulsion energy states, $X_4 + X_9$. In Figure 6.4a), the vertical axis is the sum of propulsion power states, $X_{14} + X_{19}$. In Figure 6.4b), the vertical axis is the genset power state for X_i, $i \in \{11,12,16,17\}$. Task-specific constraints $\mathcal{C}_{13}$ and $\mathcal{C}_{18}$, corresponding to slow speed operation near the departure and arrival ports respectively, are shown as gray regions in Figure 6.4a). Likewise, task-specific constraints $\mathcal{C}_i$, $i \in \{9,10,11,12,14,15,16,17\}$, corresponding to emissions-free operation near ports, are shown in Figure 6.4b). The propulsion energy states start at 0MJ on the left-hand side and move towards the goal region shown as the red region on the right-hand side of both subplots. To satisfy the slow speed constraints, the planned propulsion power states shown in pink in Figure 6.4a) remain below 0.3MW during the beginning and ending phases of the mission while the vehicle is near a port. Likewise, to satisfy the emissions restrictions, the planned genset power states, shown as dashed lines in Figure 6.4b), remain at

0MW during the beginning and ending phases of the mission while the vehicle is near a port. Note that the planned power states of auxiliary gensets 1 and 2 are identical in these results. Both subplots of Figure 6.4 show the planner's consideration of the power state error bound. Recall that for the type 1 subsystems of the SPS, energy state error accumulates, growing larger over the course of the mission. As the vehicle approaches the arrival port, a significant amount of error has accumulated for the propulsion energy states, so the planner must account for this by preemptively reducing the total propulsion power and turning off all the gensets before the task-specific constraints become active. This can be seen in both subplots of Figure 6.4 as the propulsion power and genset power states remain at more conservative distances from the task-specific constraints at the end of the mission than at the beginning. Further, due to this error the mission plan does not terminate immediately at the edge of the goal region, but terminates after the planned trajectory is

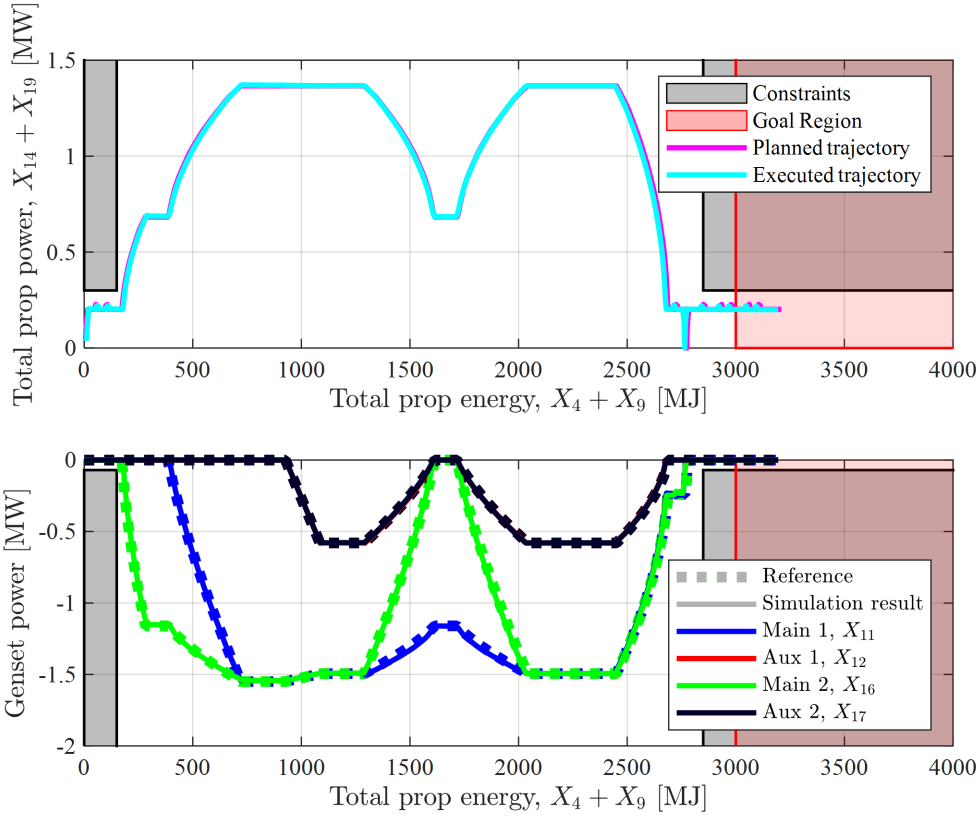


Figure 6.4: SPS task-specific constraints and offline planning simulation results. Task-specific constraints, corresponding to a) slow speed and b) emissions restrictions near ports are satisfied as guaranteed by the two-stage approach.

well within the interior of the goal region. Outside of the port regions, only the operational constraints are active, allowing the planner more freedom to select energy and power states. Because the sampling distribution is biased towards the goal region, the planner chooses high propulsion power in this regime to rapidly move towards the goal. In addition to the task-specific constraints shown in Figure 6.4, RRT assures satisfaction of operational constraints $\mathcal{C}_{1-8}$ to ensure safe operation of gensets and battery packs (see Appendix A.4).

In Figure 6.4, the tracking controller yields minimal deviations between the reference trajectory and simulation results. Figure 6.5 plots the time trajectories of all reference and executed power states of the reduced order model, showing the controller's tracking performance. To guarantee constraint satisfaction, the tracking controller is required to satisfy power state tracking error bounds. Figure 6.6 provides a magnified view of the tracking error and error bound corresponding to the power states shown in Figure 6.5, showing that the error bound is satisfied using RMPC. While the error bounds appear large in the magnified view of Figure 6.6, it can be noted that these error bounds are small in comparison to the magnitude of the corresponding power states shown in Figure 6.5.

Figure 6.4-Figure 6.6 show the simulation results for one 6000s mission, for which the computation time of the planner was under 6s. Highlighting the rapid, long-term planning capability of the RRT algorithm, the planner generated a feasible mission plan in under 1/1000th of the time taken to perform the mission. Due to the randomized nature of the planner, different feasible trajectories may be found for the same mission scenario with varying computation times.

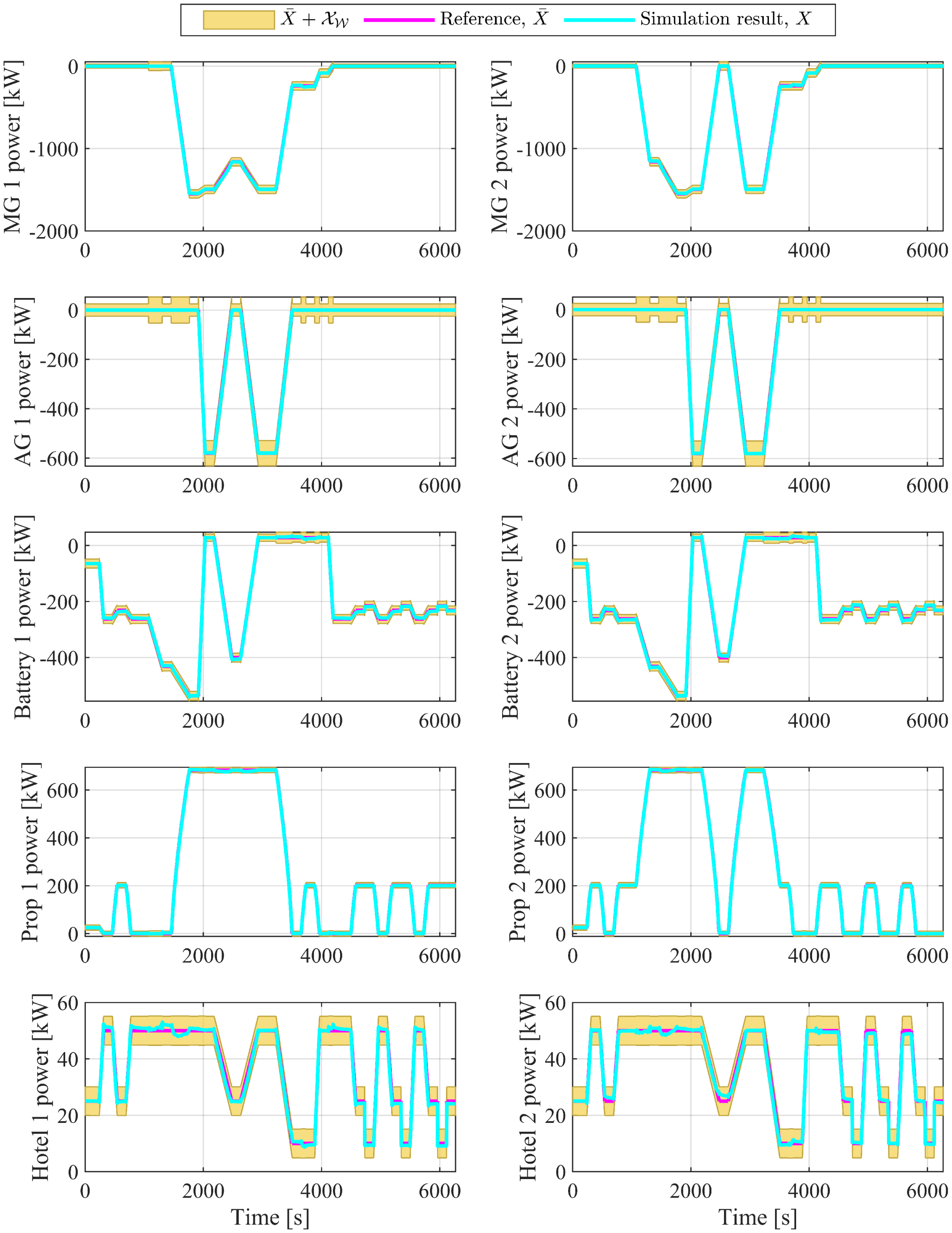


Figure 6.5: Power references, states, and error bounds for SPS offline planning scenario. RMPC ensures that the power states track the reference trajectory from the planner. Abbreviations MG, AG are used for main genset and aux genset,

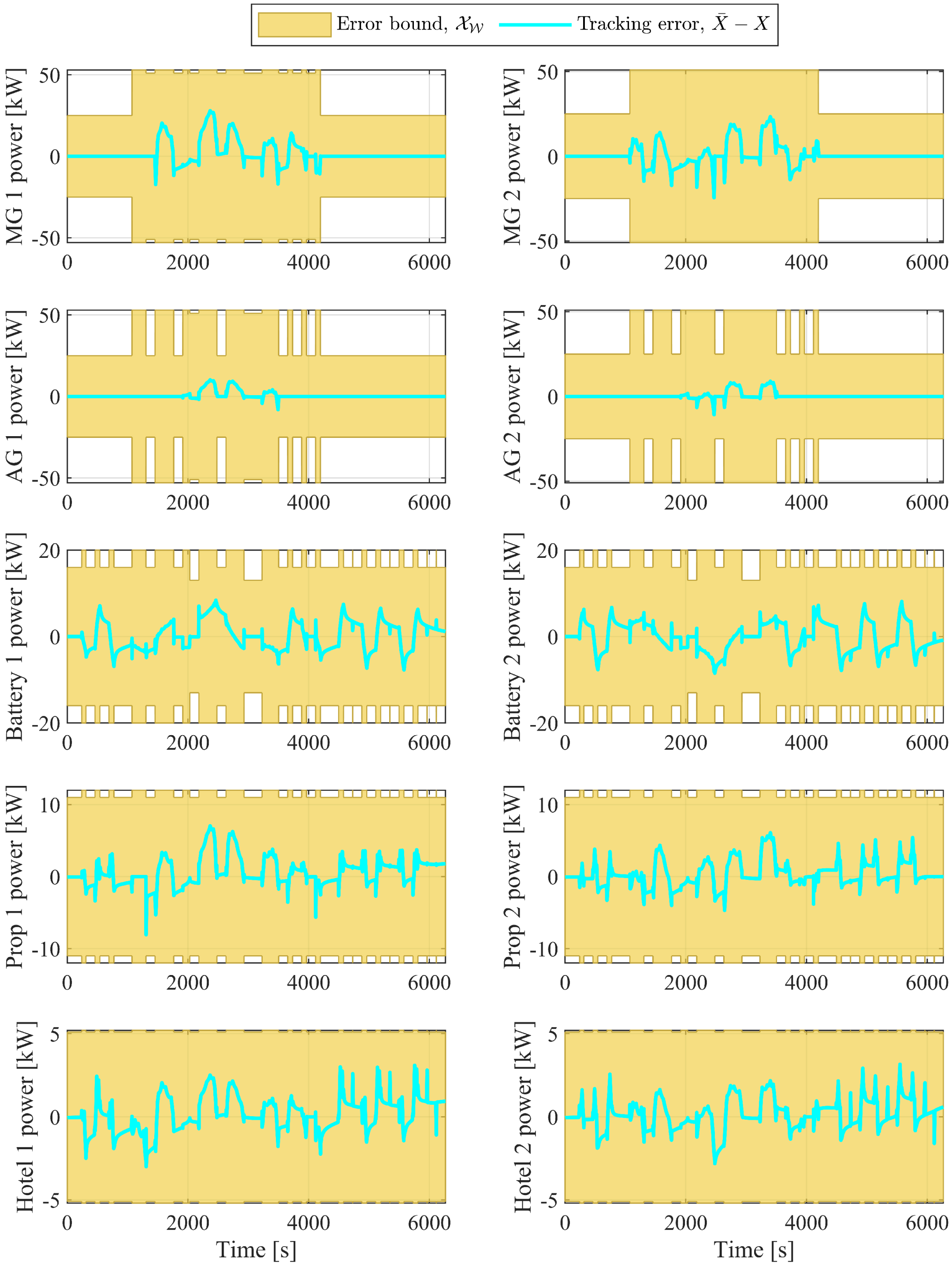


Figure 6.6: Simulated tracking error for SPS offline planning scenario. RMPC enables satisfaction of error bound to ensure the offline SPS mission plan is feasible. Abbreviations MG, AG are used for main genset and aux genset, respectively.

To investigate the computational statistics of the planner, computation times for 100 trials of this offline planning scenario were recorded. The results of these trials are summarized in Figure 6.7. These computation times do not include pre-processing steps taken to set up the constraints, goal region, and other parameters, considering only the time taken to run the RRT algorithm. Of these 100 trials, the median computation time was under 1 second and the maximum computation time was 5.92s corresponding to the results discussed above in Figure 6.4-Figure 6.6. In summary, Figure 6.7 shows that this worst-case planning time remains <1/1000th of the time taken to perform the mission, making this approach feasible for online implementation in practice.

6.1.4. Online re-planning simulation results

The demonstrated computational efficiency of the sampling-based mission planner enables its use to adjust the mission plan online if mission specifications change. In this section, uncertainty introduces time-varying changes in the task-specific constraints. Re-planning is performed online to adjust to these changes. One source of uncertainty is the inaccurate estimate of propulsion energy usage required to travel a given distance. As mentioned in Section 6.1.1.2, this estimate is obtained using a linear mapping $\hat{E}(x)$ which is subject to model error, especially when sea conditions such as currents or winds vary. The varying sea conditions considered in this scenario,

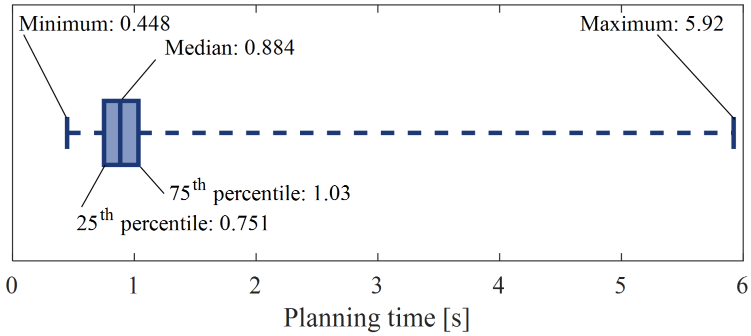


Figure 6.7: Computational statistics for 100 trials of offline planning of SPS using RRT.

which were randomly generated, are shown in Figure 6.8. Because the task-specific constraints and goal region are parameterized by the propulsion energy state, this uncertainty causes the mission specifications to shift over the course of the mission.

Additionally, uncertainty is incorporated in this scenario in the form of additional task-specific constraints, unknown at the beginning of the mission, that become detected while the mission is underway. The first additional constraint enforces an outage for main genset 1. The second additional constraint enforces slow speed operation for the propulsion subsystems. The final additional constraint requires high hotel power. These task-specific constraints are location-based, meaning that they are active at certain locations between the ports. The corresponding propulsion energy estimate is obtained using the linear mapping $\hat{E}(x)$ of the propulsion energy required to travel a distance x. Table 6.4 summarizes all location-based mission specifications considered in this online re-planning scenario. These constraints are considered in addition to the operational constraints $\mathcal{C}_{1-8}$ described by (6.1)-(6.3).

A simulation-based demonstration of the two-stage approach for online re-planning is described below. For this demonstration a modified ERRT algorithm was used which saved waypoints from previous planning iterations, using them to bias the planner's search. Unlike the

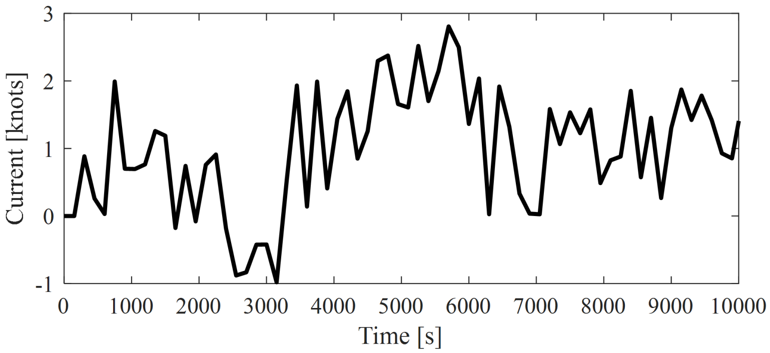


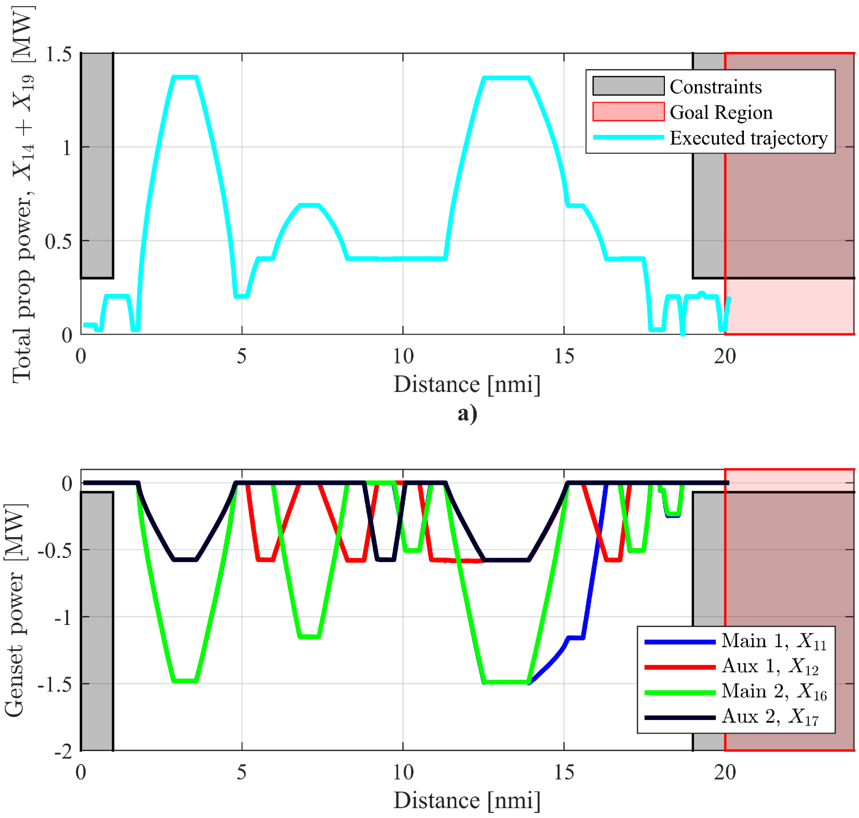
Figure 6.8: Ocean current data considered in SPS online re-planning case study.

original ERRT algorithm, however, the cost function used here considered only the distance metric ((4.33) with $\beta = 0$). These simulation results were obtained using the same computational resources and toolboxes as specified in the offline planning scenario of Section 6.1.3.

Simulation results of the two-stage method for online re-planning of the SPS shown in Figure 6.9 demonstrate satisfaction of the location-based constraints $\mathcal{C}_i$, $i \in \{9,\ldots,18\}$, near ports despite uncertainty in vehicle location. In both subplots of Figure 6.9, the horizontal axis is the distance

Table 6.4: Summary of location-based mission specifications for SPS online re-planning scenario.

Description	*Vehicle distance [nmi]*	*Detection time [s]*	*Constraint set*
No emissions constraint near port 1	0-1	0	$\mathcal{C}_{8+j} = \left\{X : \hat{E}(0) \le X_4 + X_9 \le \hat{E}(1), X_{i(j)} \le -0.07\text{MW}\right\},$ $j \in \{1,2,3,4\}, i = [11,12,16,17]$
Slow speed constraint near port 1	0-1	0	$\mathcal{C}_{13} = \left\{X : \hat{E}(0) \le X_4 + X_9 \le \hat{E}(1), 0.3\text{MW} \le X_{14} + X_{19}\right\}$
No emissions constraint near port 2	19	0	$\mathcal{C}_{13+j} = \left\{X : \hat{E}(19) \le X_4 + X_9, X_{i(j)} \le -0.07\text{MW}\right\},$ $j \in \{1,2,3,4\}, i = [11,12,16,17]$
Slow speed constraint near port 2	19	0	$\mathcal{C}_{18} = \left\{X : \hat{E}(19) \le X_4 + X_9, 0.3\text{MW} \le X_{14} + X_{19}\right\}$
Main genset 1 outage constraint	7-9	350	$\mathcal{C}_{19} = \left\{X : \hat{E}(7) \le X_4 + X_9 \le \hat{E}(9), X_{11} \le -0.07\text{MW}\right\}$
Slow speed constraint, open ocean	10-10.5	1050	$\mathcal{C}_{20} = \left\{X : \hat{E}(10) \le X_4 + X_9 \le \hat{E}(10.5), 0.5\text{MW} \le X_{14} + X_{19}\right\}$
High hotel load constraint	12-14	1350	$\mathcal{C}_{21} = \left\{X : \hat{E}(12) \le X_4 + X_9 \le \hat{E}(14), X_{15} + X_{20} \le 0.7\text{MW}\right\}$
Goal region (port 2)	20	0	$\mathcal{X}_G = \left\{X : \hat{E}(20) \le X_4 + X_9\right\}$



a)

b)

Figure 6.9: SPS task-specific port constraints and online re-planning simulation results. Despite uncertainty, these constraints, corresponding to a) slow speed and b) emissions restrictions near ports, are satisfied using the two-stage approach.

traveled by the vehicle. The vertical axes of Figure 6.9 are defined as in Figure 6.4. The executed trajectories are plotted in both subplots. To satisfy the slow speed constraints, the executed propulsion power states shown in cyan in Figure 6.9a) remain below 0.3MW during the beginning and ending phases of the mission while the vehicle is near a port. Likewise, to satisfy the emissions restrictions, the planned genset power states shown in Figure 6.9b) remain at 0MW during the beginning and ending phases of the mission while the vehicle is near a port. Figure 6.9a) shows that after leaving port 1, while the vehicle travel distance is approximately 10nmi, the propulsion power states remain smaller than in the offline planning scenario shown in Figure 6.4a). This choice made by the planner ensures satisfaction of the slow speed constraint $\mathcal{C}_{20}$ which is not known initially in the offline planning scenario.

The initially unknown constraints, $\mathcal{C}_{19}$, $\mathcal{C}_{20}$, and $\mathcal{C}_{21}$, are shown in Figure 6.10a), b), and c), respectively. In each subplot, the executed trajectories for the online re-planning and offline planned scenarios are shown in cyan and teal respectively. While the trajectory obtained via online re-planning satisfies all the location-specific constraints, the offline mission plan does not satisfy these constraints. The final conditions of both trajectories are shown as crosses. While the final

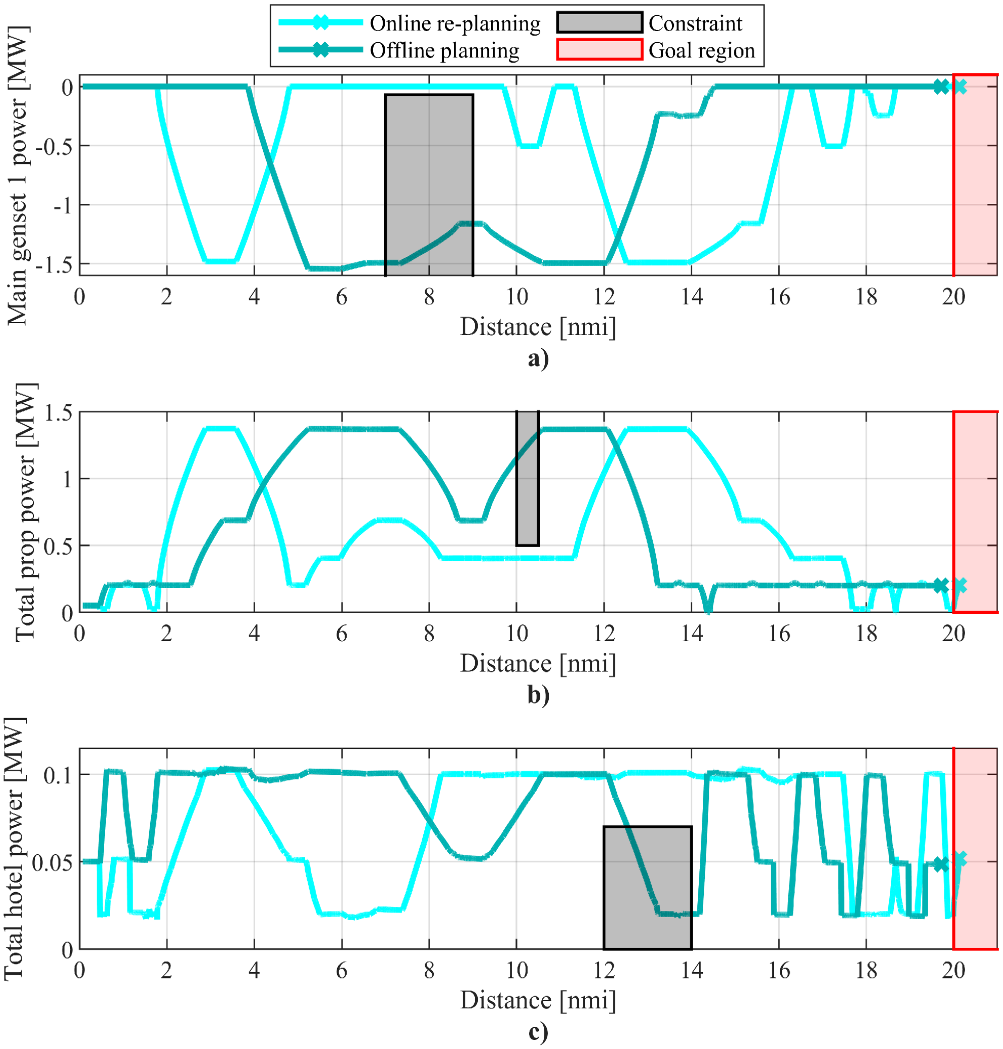


Figure 6.10: Initially unknown constraints for SPS are satisfied when re-planning is performed online, but offline planning does not satisfy these constraints.

condition of the trajectory obtained via online re-planning lands correctly in the goal region, the trajectory obtained via offline planning does not land in the goal region due to uncertainty in the propulsion energy state. In addition to task-specific constraints shown in Figure 6.9 and Figure 6.10, operational constraints are also satisfied during online re-planning to assure safety (see Appendix A.4).

RMPC is used to ensure that the executed trajectory tracks the reference trajectory even as the reference changes during re-planning. Figure 6.11 plots the time trajectories of all reference and executed power states of the reduced order model, showing the controller’s tracking performance. Figure 6.12 shows that the power state error bound is satisfied using RMPC. Figure 6.11 provides additional perspective on how the system operates to satisfy the mission specifications. For example, all gensets have zero power during the initial and final 1000s of the mission to satisfy the emissions constraints. Likewise, the power state of both propulsion subsystems remains near or below 200kW in these timeframes to satisfy the slow speed constraints.

Figure 6.13 shows the computation times required to update the mission plan each time the planner is called. In Figure 6.13, the horizontal axis corresponds to the time at which the planner began re-planning and the vertical axis corresponds to the computation time for each instance. These computation times include all pre-processing steps required to set up the planning problem. The mean of these planning times is 1.67s and the minimum is 0.776s, both of which are larger than those of the offline results reported in Figure 6.7. These computational increases can be attributed to pre-processing of mission specifications. Despite this increase, Figure 6.13 highlights the computational efficiency of the modified ERRT algorithm, which enables the SPS to satisfy complex and changing mission specifications.

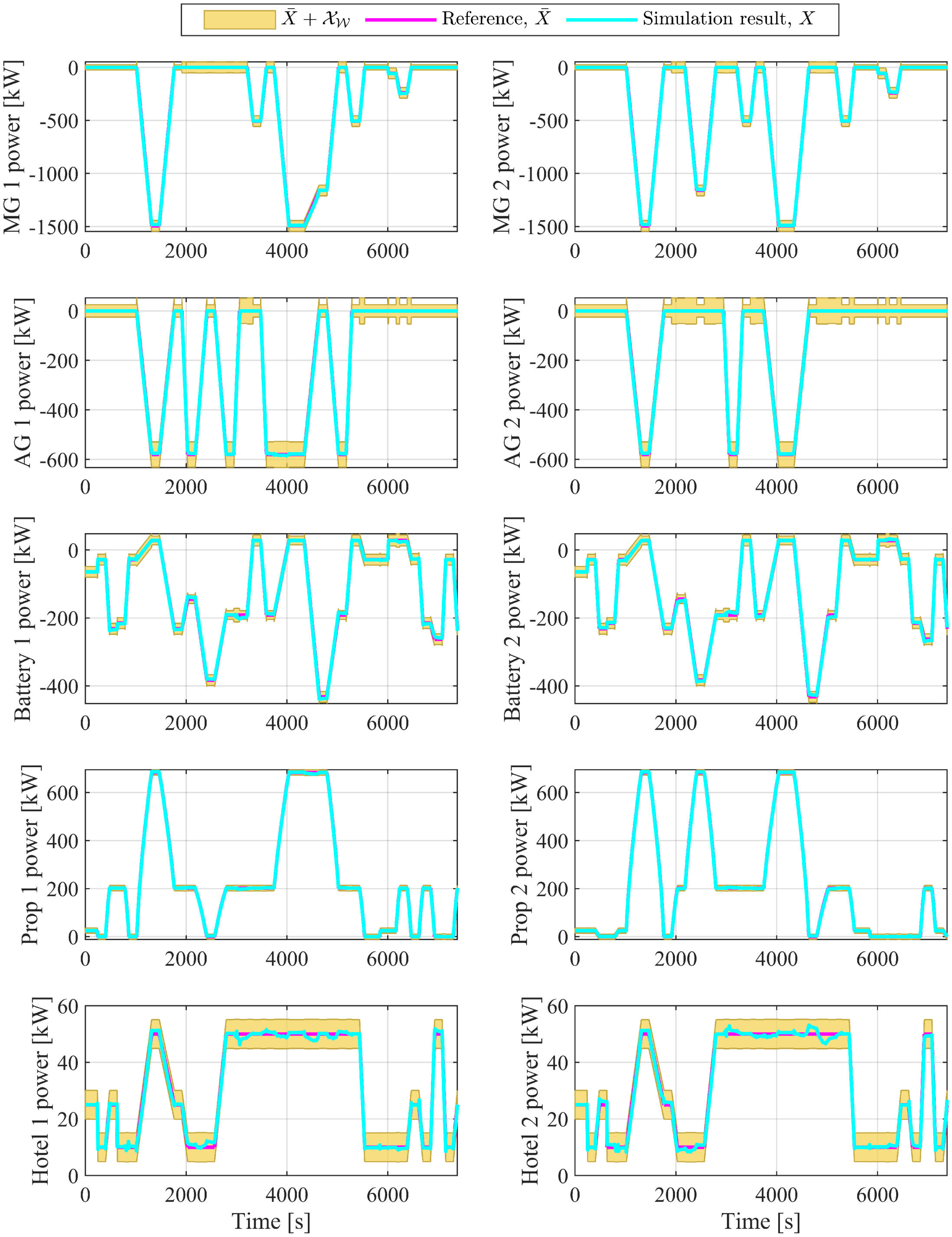


Figure 6.11: Power references, states, and error bounds for SPS online re-planning scenario. RMPC ensures that the power states track the reference trajectory from the planner. Abbreviations MG, AG are used for main genset and aux genset, respectively.

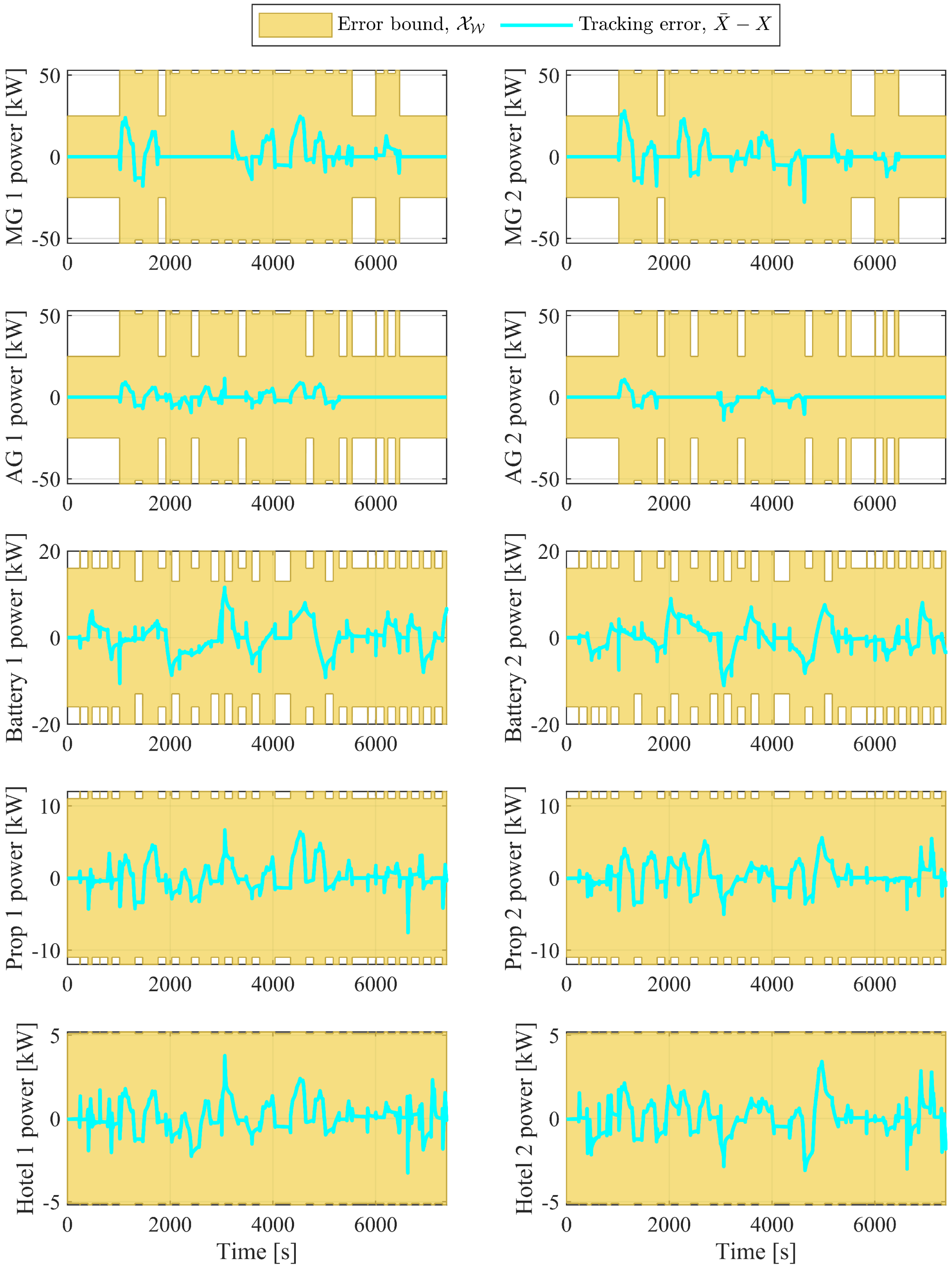


Figure 6.12: Simulated tracking error for SPS online re-planning scenario. RMPC enables satisfaction of error bound to ensure the SPS mission plan obtained via online re-planning is feasible. Abbreviations MG, AG are used for main genset and aux genset, respectively.

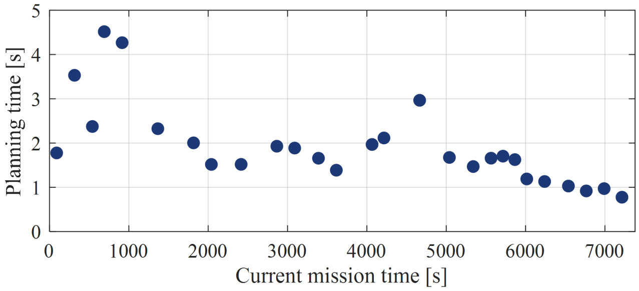


Figure 6.13: Planning computation times for SPS online re-planning. Plotted in terms of the current mission time (i.e., the time at which re-planning was initiated).

6.2. HYBRID UAV POWER, PROPULSION, AND THERMAL SYSTEM

This case study considers operation of a hybrid electric UAV power, propulsion, and thermal system, described in Section 3.4.3. While the previous study focused strictly on energy management for the electro-mechanical dynamics of the SPS powertrain, the present case study demonstrates the flexibility of this approach to perform long-term planning for the thermal domain in addition to electrical and mechanical domains. The two-stage framework is shown in this case study to enable thermal management in addition to energy management of the powertrain. Simulation results demonstrate rapid re-planning online in the presence of uncertain, varying constraints. Parameters of the simulation model are provided in Appendix B.1.

6.2.1. Sampling-based planning problem setup

A brief overview of the setup for the hybrid UAV PPTS sampling-based planning problem is given below. Additional parameters used to solve this planning problem can be found in Appendix B.2.

6.2.1.1. Reduced-order model

The six subsystems of the hybrid UAV PPTS are listed in Table 6.5, including the edges of the graph model shown in Figure 3.5 corresponding to the power state of each subsystem. In Table 6.5, an asterisk is used to denote those subsystems for which the power state polarity is

Table 6.5: Description of hybrid UAV PPTS subsystems.

Subsystem number	*Subsystem description*	*Subsystem type*	*Edge(s) corresponding to subsystem power state*
1	Genset	1	62
2	Battery pack	1	1*
3	Propulsion	1	32, 34
4	Avionic load	1	27
5	Power electronics bay	2	20, 30, 41, 48, 49, 58
6	Battery air bay	2	7,51

opposite to the corresponding graph model power flow. Note that subsystems 1-4, which correspond to electro-mechanical subsystems, are of type 1, while the air cooling bays of subsystems 5 and 6 are of type 2. The planning model for this system consists of six energy states and six power states, corresponding to a 12-dimensional planning problem.

6.2.1.2. Mission specifications

Operational constraints for the hybrid UAV PPTS ensure safe operation of the battery pack and engine. The following constraint ensures that the energy state of the genset remains positive such that the engine does not run out of fuel.

$$\mathcal{C}_1 = \{X : X_1 \leq 0\text{kJ}\} \tag{6.14}$$

The following constraint ensures that the genset is not operated beyond its maximum power rating of 3.3kW.

$$\mathcal{C}_2 = \left\{X : X_7 \leq -3.3\text{kW}\right\} \tag{6.15}$$

The following operational constraint provides a lower bound for the battery pack's energy state, such that the battery pack is not discharged beyond its capacity.

$$\mathcal{C}_3 = \{X : X_2 \leq 3600QV_{ocv}(0.3)\} \tag{6.16}$$

See (6.2) for definitions of Q and V_{ocv}. Constraint (6.16) corresponds to a lower SOC bound of 30%. The following operational constraint provides an upper bound for the battery pack's energy state, which corresponds to 80% SOC.

$$\mathcal{C}_4 = \{X : 3600QV_{ocv}(0.8) \le X_2\} \tag{6.17}$$

Battery charge rates greater than C/2, which are considered unsafe, are prevented by enforcing the following constraint.

$$\mathcal{C}_5 = \{X : 700\text{W} \le X_8\} \tag{6.18}$$

Unsafe battery discharge rates are prevented by enforcing the following constraint.

$$\mathcal{C}_6 = \{X : X_8 \le -5890\text{W}\} \tag{6.19}$$

To keep component temperatures in a safe range, operational constraints limit the energy states of the air cooling bays. The following constraint provides a lower bound for the energy states of the air bays, which keeps the temperature of the air bays above 20°C.

$$\mathcal{C}_{6+j} = \left\{X : X_{i(j)} \le 40\text{J}\right\}, j \in \{1,2\}, i = [5,6] \tag{6.20}$$

The following constraint provides an upper bound for the energy states of the air bays, which keeps the temperature of the air bays below 35°C.

$$\mathcal{C}_{8+j} = \left\{X : 70\text{J} \le X_{i(j)}\right\}, j \in \{1,2\}, i = [5,6] \tag{6.21}$$

The initial condition of the reduced order model, $X_0 = [E_0^T \quad P_0^T]^T$, is given below. The initial energy state E_0 corresponds to full engine fuel capacity, 70% battery pack SOC, zero energy use by the propulsion and avionic subsystems, and bay temperatures of 25°C. The initial power states P_0 correspond to low speed, low avionic power operation with the battery in charge sustaining mode.

$$E_0 = [19.5\text{MJ} \quad 3.37\text{MJ} \quad 0 \quad 0 \quad 50.0\text{J} \quad 50.0\text{J}]^T \tag{6.22}$$

$$P_0 = [-1.80 \quad 0 \quad 1.20 \quad 0.175 \quad 0 \quad 0]^T \text{kW} \tag{6.23}$$

Task-specific constraints restrict the operation of the genset, avionic, and electronics bay subsystems during segments of the mission based on the vehicle location, as illustrated in Figure 1.2. Some constraints force the engine to turn off by forbidding operation of the genset with negative power. Due to the sign convention, these constraints permit only those energy primitives that operate the genset with zero power. This corresponds to a mission segment in which engine disturbances due to noise or emissions cannot be tolerated (i.e., "quiet zone"). A constraint forbidding low power operation of avionic equipment represents a segment of the mission in which auxiliary equipment such as a sensor requires high power without interruption. Some constraints forbid operation of the avionic equipment at high power (e.g., 0.5kW). Likewise, some constraints prevent the genset from operating at large negative power levels, which correspond to high power operation. These constraints can prevent the generator and avionic equipment from moving close to a failure mode, for example due to increased thermal stresses, at high power.

To represent these location-based constraints in the energy domain, a linear mapping $\hat{X}_3(x)$ is used to estimate the propulsion energy X_3 required to travel a distance x. While some task-specific constraints are known at the beginning of the mission, this case study considers additional task-specific constraints that arise while the mission is underway. The initially known constraints are the quiet zone $\mathcal{C}_{11}$, requiring zero genset power ($X_7 = 0$), and sensing mode $\mathcal{C}_{12}$, requiring high avionic power ($X_{10} \geq 0.24\text{kW}$). After 120 seconds, the electronics bay limited constraint $\mathcal{C}_{13}$ is detected which enforces a tighter upper bound, corresponding to 27.5°C, for the electronics bay energy X_5. After 240 seconds, the genset limited constraint $\mathcal{C}_{14}$ prevents the genset from operating in high power modes. After 1140 seconds, the avionics limited constraint $\mathcal{C}_{15}$ is detected which prevents the avionic subsystem from operating in high power modes. In addition to the task-specific constraints, the goal region is also location-based. The goal region requires 65-75% battery SOC and 23.5 miles of travel. The full set of location-based mission specifications is summarized in Table 6.6.

6.2.1.3. Primitive parameterization

Trim trajectories were generated by first defining two operating modes for each component, as listed in Table 6.7. Genset modes correspond to operation with the engine off and at a fuel-efficient power level. Battery modes discharge the battery at a moderate rate or sustain its charge by operating with zero power. Propulsion system modes correspond to low and high vehicle speeds. Avionic load modes correspond to low and high avionic power demands. The first power electronics bay mode, P_{5a}, corresponds to operating the fan at a low power level, providing

Table 6.6: Summary of task-specific constraints for hybrid UAV PPTS case study.

Description	*Vehicle distance [mi]*	*Detection time [s]*	*Constraint set*
Quiet zone constraint	5.2-6.2	0	$\mathcal{C}_{11} = \left\{X : \hat{X}_3(5.2) \le X_3 \le \hat{X}_3(6.2), X_7 \le -0.12\text{kW}\right\}$
Sensing mode constraint	14-15	0	$\mathcal{C}_{12} = \left\{X : \hat{X}_3(14) \le X_3 \le \hat{X}_3(15), X_{10} \le 0.24\text{kW}\right\}$
Electronics bay limited constraint	16-30	120	$\mathcal{C}_{13} = \left\{X : \hat{X}_3(16) \le X_3 \le \hat{X}_3(30), 55\text{J} \le X_5\right\}$
Genset limited constraint	10-12	240	$\mathcal{C}_{14} = \left\{X : \hat{X}_3(14) \le X_3 \le \hat{X}_3(15), X_7 \le -3.0\text{kW}\right\}$
Avionic limited constraint	19.5-20.5	1140	$\mathcal{C}_{15} = \left\{X : \hat{X}_3(19.5) \le X_3 \le \hat{X}_3(20.5), 0.3\text{kW} \le X_{10}\right\}$
Goal region	23.5	0	$\mathcal{X}_G = \{X : \hat{X}_3(23.5) \le X_3, 3600QV_{ocv}(0.65) \le X_2, \ldots$ $X_2 \le 3600QV_{ocv}(0.75)\}$

minimal cooling and allowing the bay temperature to rise. The second electronics bay mode, P_{5b}, operates the fan at a higher power level, providing increased cooling to reduce the bay temperature. Battery bay modes are similar to power electronics bay modes but the low power fan mode, P_{6a}, is a function of the battery power, P_2, which is given in Appendix B.2. To generate trim trajectories, all possible combinations of operating modes listed in Table 6.7 are considered. These are listed in Table B.2 of Appendix B.2. Time durations of trim trajectories are 60s. Transitions between pairs of trim trajectories consist of a single phase, with ramp rate and time duration determined by (4.37), (4.38) with $R^{max} = 40\text{W/s}$.

6.2.2. RMPC problem setup

The RMPC nominal model of the hybrid UAV PPTS is obtained by linearizing and discretizing the graph-based model shown in Figure 3.5. Feedback controllers for the hybrid UAV PPTS track state references for selected states of the full-order model. Table 6.8 lists the regulated

Table 6.7: Operating modes for subsystems of the hybrid UAV PPTS.

Subsystem	*Operating modes*	*Descriptions*
Genset	$P_{1a} = 0\text{W}$	Engine off
	$P_{1b} = -2480\text{W}$	Fuel-optimal engine power
Battery pack	$P_{2a} = 0\text{W}$	Battery charge sustaining
	$P_{2b} = -500\text{W}$	C/3 battery discharge
Propulsion system	$P_{3a} = 1200\text{W}$	Low speed
	$P_{3b} = 2000\text{W}$	High speed
Avionic load	$P_{4a} = 175\text{W}$	Low avionic demand
	$P_{4b} = 350\text{W}$	High avionic demand
Power electronics bay	$P_{5a} = 0.02\text{W}$	Cooling fan low
	$P_{5b} = -0.01\text{W}$	Cooling fan high
Battery bay	$P_{6a} = P_{6a}(P_2)$	Cooling fan off/low
	$P_{6b} = -0.01\text{W}$	Cooling fan high

state and control gains corresponding to each control signal. Each input u_j is constrained by the following.

$$u_j \in [0,1] \tag{6.24}$$

Reachable sets describing error between the nominal model and true system under the action of this feedback controller are obtained following the process described in Sections 5.2 and 5.3. To ensure safe operation, RMPC applies constraints to certain states of the full-order model. These constraints are summarized in Table 6.9. Additional RMPC parameters can be found in Appendix B.3.

6.2.3. Online re-planning simulation results

This section provides a simulation-based demonstration of the two-stage method for online re-planning of the hybrid UAV PPTS using the problem setup described above. For this demonstration, all computations were performed on an Intel i7 processor with 32GB RAM. The

Table 6.8: Hybrid UAV PPTS feedback controller description.

Input description	*Regulated state*	K_P	K_I
Inverter duty cycle	Prop speed	1.6E-1	1.6E-2
Avionic DC-DC converter duty cycle	Avionic load current	3.9E-5	2.0E-1
Fan 1 DC-DC converter duty cycle	Electronics bay fan current	3.9E-5	2.0E-1
Fan 2 DC-DC converter duty cycle	Battery bay fan current	3.9E-5	-1.0E-2
Rectifier duty cycle	Generator motor speed	1.6E-1	1.6E-2

Table 6.9: RMPC state constraints for hybrid UAV PPTS.

State description [unit]	*Slack variable weight*	*Lower bound*	*Upper bound*
Battery state of charge [-]	1E1	0.3	0.9
Propulsion motor speed [rpm]	1E-5	1500	5000
Vehicle speed [m/s]	1E-2	1	100
Avionic load current [A]	1E-2	0	50
Fan 1 current [A]	1E-2	0	10
Fan 2 current [A]	1E-2	0	10
Generator motor speed [rpm]	1E0	0	7000

RRT strategy described in Section 4.3.1 was used for online re-planning using a custom MATLAB code. To solve the RMPC optimization problem described in Section 5.4, CORA [91] was used to calculate linearization error reachable sets, the YALMIP toolbox [96] was used to formulate the optimization problem, and Gurobi [95] was used to solve the optimization problem online.

As in the SPS case study of Section 6.1.4, this case study considers uncertainty which introduces time-varying changes in the task-specific constraints. One source of uncertainty is the detection of additional task-specific constraints, unknown at the beginning of the mission, as discussed in Section 6.2.1.2. Additional uncertainty arises from the inaccurate estimate of propulsion energy usage required to travel a given distance. This estimate is obtained using the linear mapping $\hat{E}_3(x)$ which is subject to model error, especially when wind conditions vary. The varying wind conditions considered in this scenario, based on data from [97], are shown in Figure 6.14. This uncertainty causes the location-based mission specifications to shift over the course of the mission. Re-planning is performed online to adjust to these changes.

To illustrate the planner updating the mission plan to adapt to changing mission specifications, Figure 6.15 and Figure 6.16 show the planned and executed trajectories at different times during the mission. Figure 6.15 shows the planned trajectory after 110s, and Figure 6.16 after 430s. Subplot a) of each figure shows the propulsion energy, genset power, and avionic power states. Subplot a) includes task-specific constraints $\mathcal{C}_{11}$ and $\mathcal{C}_{12}$, which correspond to the initially known quiet mode and sensing mode constraints, respectively, and the goal region. Subplot b) of each figure shows the propulsion energy, battery bay energy, and electronics bay energy, along with the operational constraints corresponding to the battery and electronics bays. Between 110s and 430s, constraints $\mathcal{C}_{13}$ and $\mathcal{C}_{14}$ are detected, which correspond to the electronics bay limited and genset limited constraints, respectively. Figure 6.16 shows the updated mission plan to ensure feasibility with respect to this changing mission information. The initial planned trajectory of Figure 6.15a) allowed the genset power to reach -3.1kW when the propulsion energy state reached 1177kJ, which would be forbidden by constraint $\mathcal{C}_{14}$. After $\mathcal{C}_{14}$ becomes known, the updated mission plan in Figure 6.16a) prevents the genset power from exceeding -2.81kW in this region to ensure that this constraint is satisfied. Figure 6.16a) also shows the effect of uncertainty in the propulsion energy state. Between 110s and 430s, this uncertainty causes slight shifts in the constraints and goal region toward higher propulsion energy states. In Figure 6.15b) the initially planned trajectory allowed the electronics bay energy to increase to a maximum of 62J as the propulsion energy increased, which would be forbidden by constraint $\mathcal{C}_{13}$. After $\mathcal{C}_{13}$ becomes

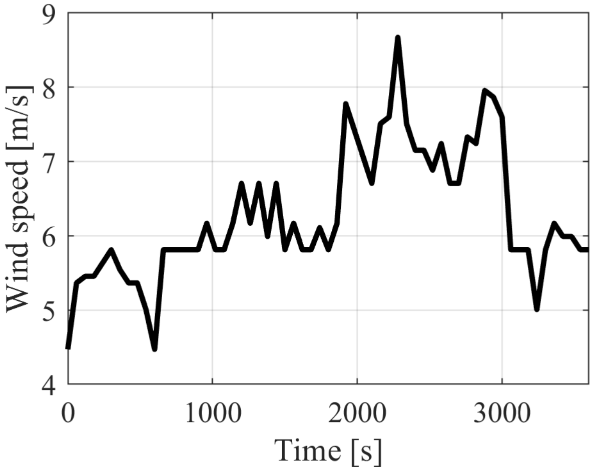


Figure 6.14: Observed wind speed conditions in Champaign, IL. From [97]. *Time scale shortened to one-hour mission duration of case study.*

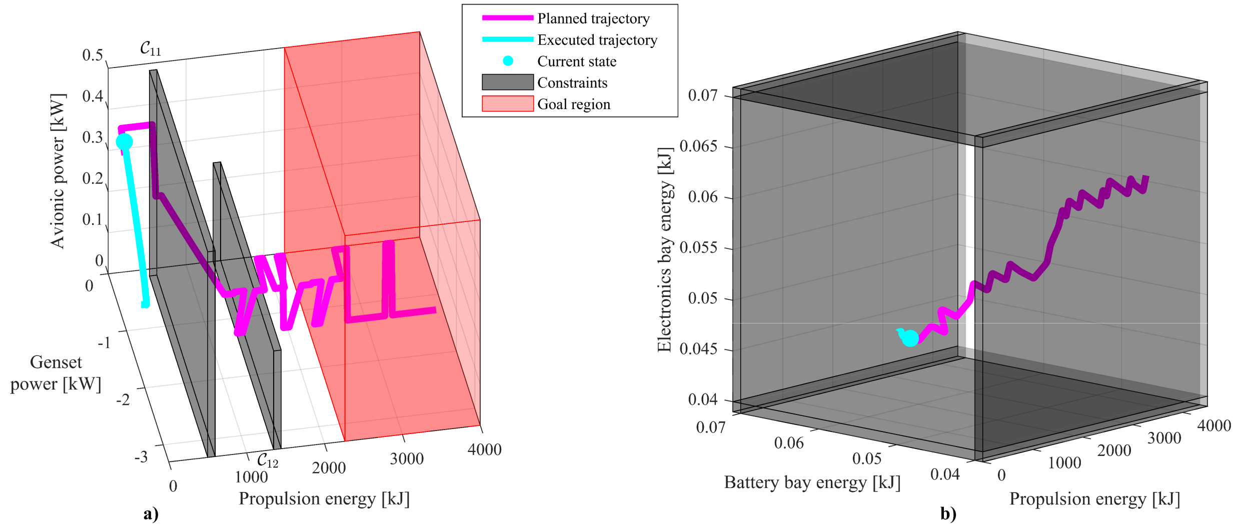


Figure 6.15: Task-specific constraints of hybrid UAV PPTS during re-planning at 110s. Only operational constraints and task-specific constraints $\mathcal{C}_{11}$ and $\mathcal{C}_{12}$ are known. The mission plan is feasible with respect to the currently known constraints.

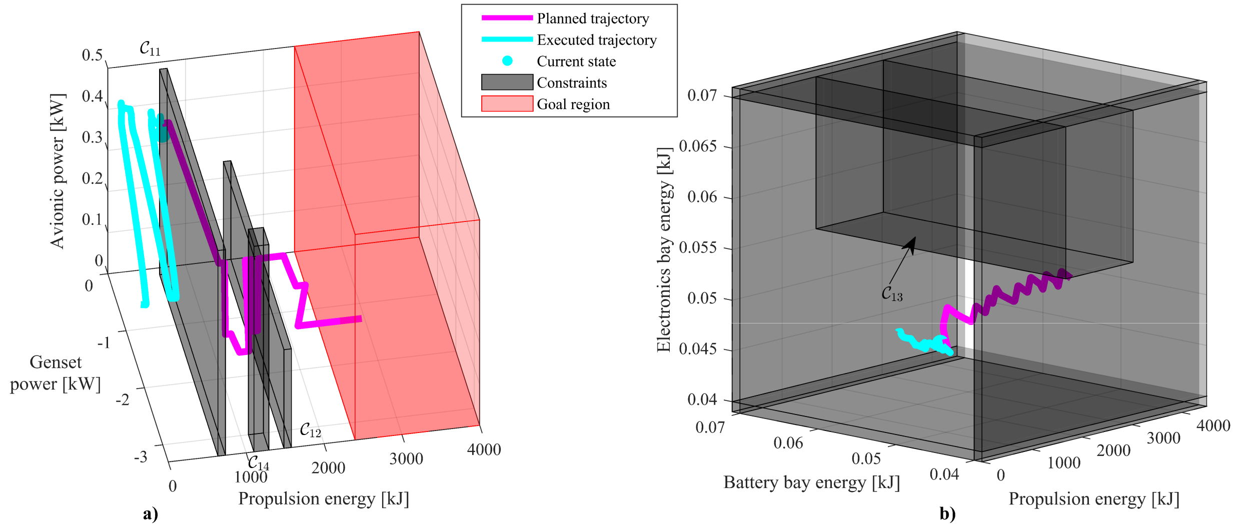


Figure 6.16: Task-specific constraints of hybrid UAV PPTS during re-planning at 430s. Only operational constraints and task-specific constraints $\mathcal{C}_{11}$ - $\mathcal{C}_{14}$ are known. The mission plan is feasible with respect to the currently known constraints.

known, Figure 6.16b) shows that the electronics bay is cooled more aggressively to keep its energy state below 55J.

Figure 6.17 and Figure 6.18 show that re-planning online enables the hybrid UAV PPTS to satisfy all of the location-based mission specifications. Figure 6.17 shows the constraints pertaining to the electronics bay subsystem, demonstrating that online re-planning enables satisfaction of initially unknown constraint $\mathcal{C}_{13}$, while the mission plan generated offline does not satisfy the constraint. The mission planned offline, shown as a dashed pink line, allows the temperature to rise throughout the mission due to heat generation by the power electronic components. When online re-planning is performed, the planner chooses to proactively cool the electronics bay to satisfy constraint $\mathcal{C}_{13}$. To demonstrate that planning using the reduced order model ensures safe operation in terms of the full order model, Figure 6.17 shows the constraints, planned trajectories, and executed trajectories in terms of the electronics bay temperature, which is proportional to the electronics bay energy state. In Figure 6.18, location-based constraints restricting the operation of the genset and avionic power states are shown. To satisfy the quiet mode constraint $\mathcal{C}_{11}$, the planner chooses to operate the genset at zero power zero (i.e., turn off the

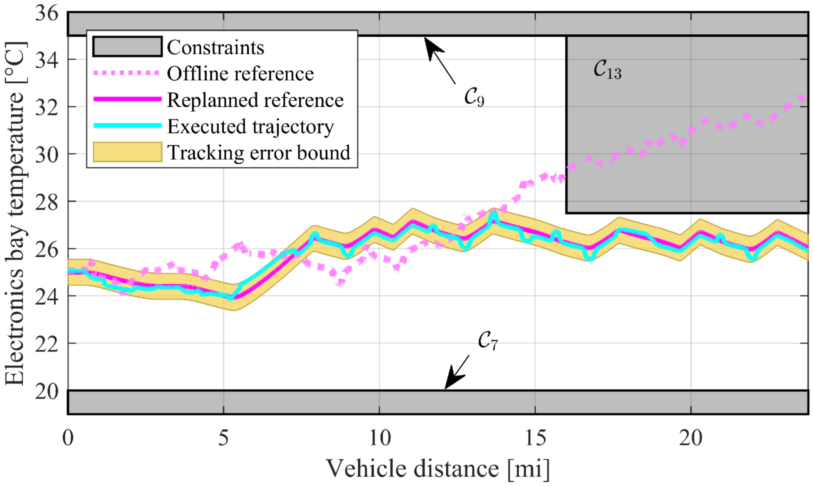


Figure 6.17: Hybrid UAV PPTS electronics bay temperature constraints are satisfied when re-planning is performed online. Operational constraints $\mathcal{C}_{7,9}$ and initially unknown task-specific constraint $\mathcal{C}_{13}$ are shown in terms of the electronics bay temperature and vehicle distance. While the offline mission plan does not satisfy the initially unknown constraint $\mathcal{C}_{13}$, online re-planning enables satisfaction of this constraint.

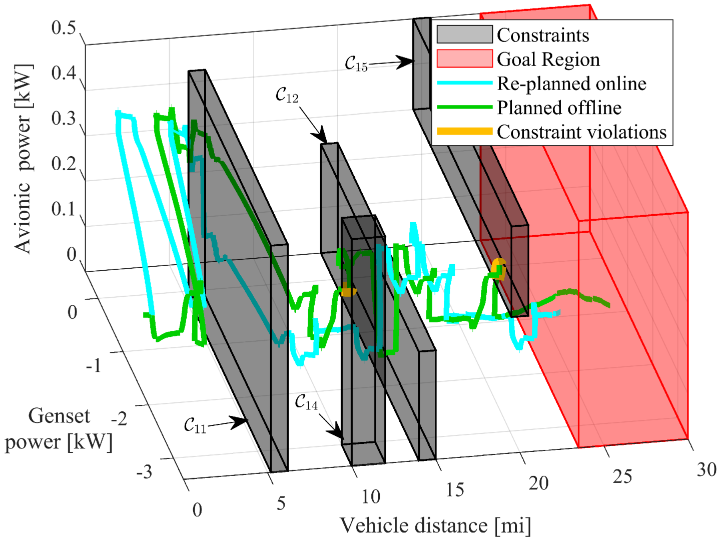


Figure 6.18: Location-based mission information for genset and avionic power states of hybrid UAV PPTS. While the offline mission plan does not satisfy the initially unknown constraints $\mathcal{C}_{14-15}$ *, online re-planning enables satisfaction of these constraints.*

genset) between 4.3-6.7 miles of vehicle travel. The planner continues to keep the genset power state well below its maximum of 3.3kW until the vehicle has traveled 14 miles, which assures satisfaction of constraint $\mathcal{C}_{14}$. To assure satisfaction of constraint $\mathcal{C}_{12}$, the avionic power state remains higher than 0.3kW between 12.7-16.8 miles of vehicle travel. To satisfy constraint $\mathcal{C}_{15}$, the avionic power state remains below 0.25 kW after the vehicle has traveled 16.9 miles. Figure 6.18 compares the executed trajectory achieved via online re-planning (shown in cyan) to one obtained from tracking a reference planned offline (shown in green). The trajectory obtained via offline planning exhibits violations (shown in yellow) of the initially unknown constraints $\mathcal{C}_{14}$ and $\mathcal{C}_{15}$. Online re-planning enables satisfaction of all constraints, including the operational constraints. For brevity, a discussion of operational constraint satisfaction is omitted but satisfaction of these constraints is shown in Figure B.3, Figure B.4, and Figure B.5 of Appendix B.4.

Figure 6.19 plots the time trajectories of all reference and executed states of the reduced order model, showing RMPC's tracking performance. Note that the air cooling bays are type 2 subsystems, so their energy states are required to satisfy a tracking error bound. Other subsystems are type 1 requiring their power states to satisfy an error bound. Figure 6.19 shows that RMPC tracks the reference trajectory keeping all relevant states within their respective error bounds. Figure 6.20 provides a detailed view of the tracking error of RMPC, demonstrating satisfaction of the error bound. The green regions of Figure 6.20 show the tightened error bound, where the amount of tightening corresponds to that of the first step of the control horizon at each time instance. Note that the battery power state exhibits the most conservativism in its tightened error bound, where the amount of tightening varies visibly during the simulation. This can be attributed to significant nonlinearities in the battery power state. With the exception of the propulsion power state, each state of the reduced order model approaches or crosses the edge of the tightened error bound at some point during the simulation. Yet violation of the tighter bound does not lead to violation of the original error bound shown in yellow.

Online re-planning is enabled by the high computational efficiency of the RRT algorithm. This is shown in Figure 6.21, where the computation time is displayed for each re-planning calculation. In each instance, the planner computes a feasible trajectory in under 10s. Considering that the time duration of this mission was over 2000s, this shows that re-planning can be done in well under 1/100th of the time needed to perform the mission. Computation times generally decrease as the mission progresses as the current state moves closer to the goal region and the planner does not have to explore as much of the space to find a feasible plan that reaches the goal. Figure 6.21 shows that the planner efficiently solves the online re-planning problems to perform

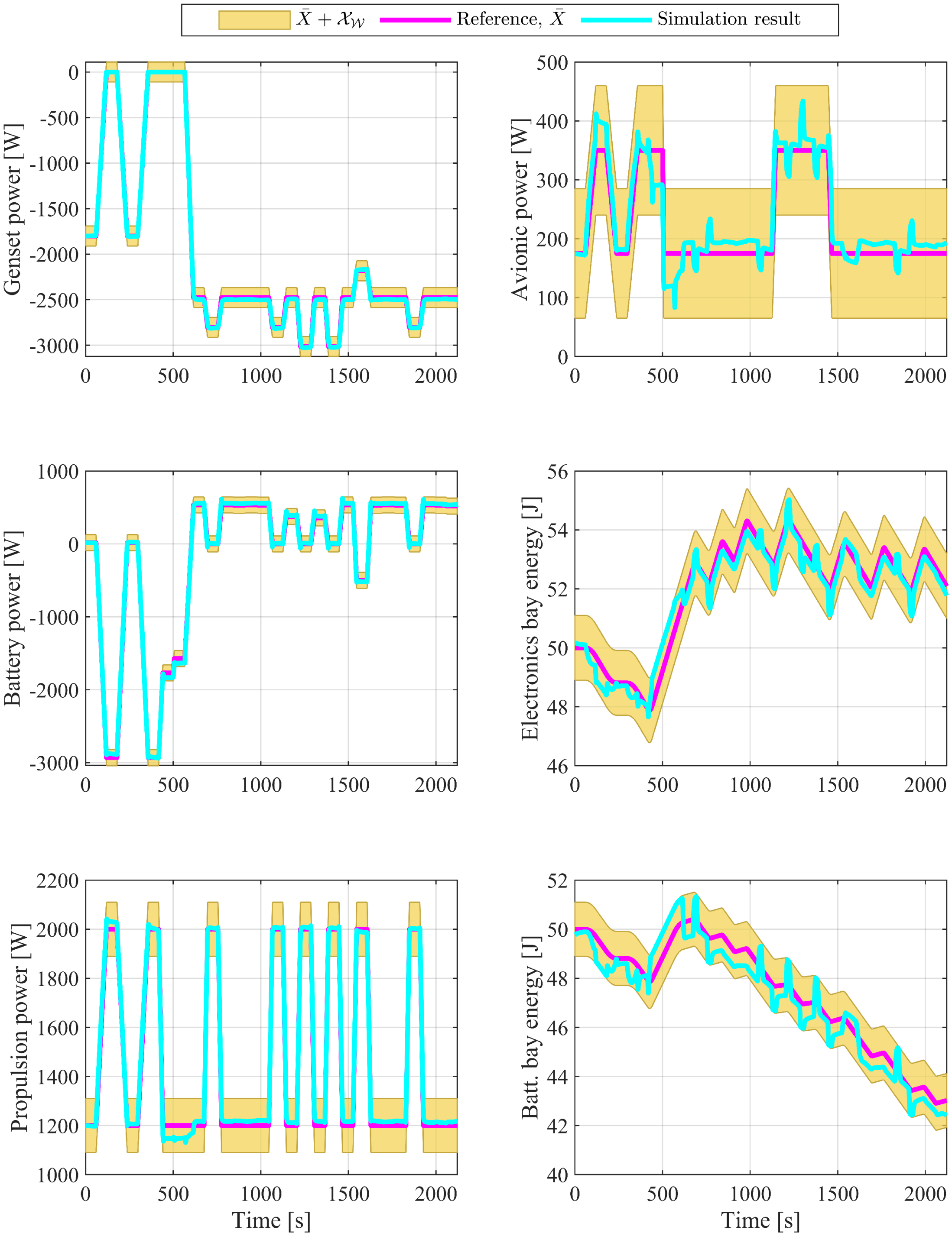


Figure 6.19: Power references, states, and error bounds for the hybrid UAV PPTS online re-planning scenario. RMPC ensures that the power states track the reference trajectory from the planner within the error bound.

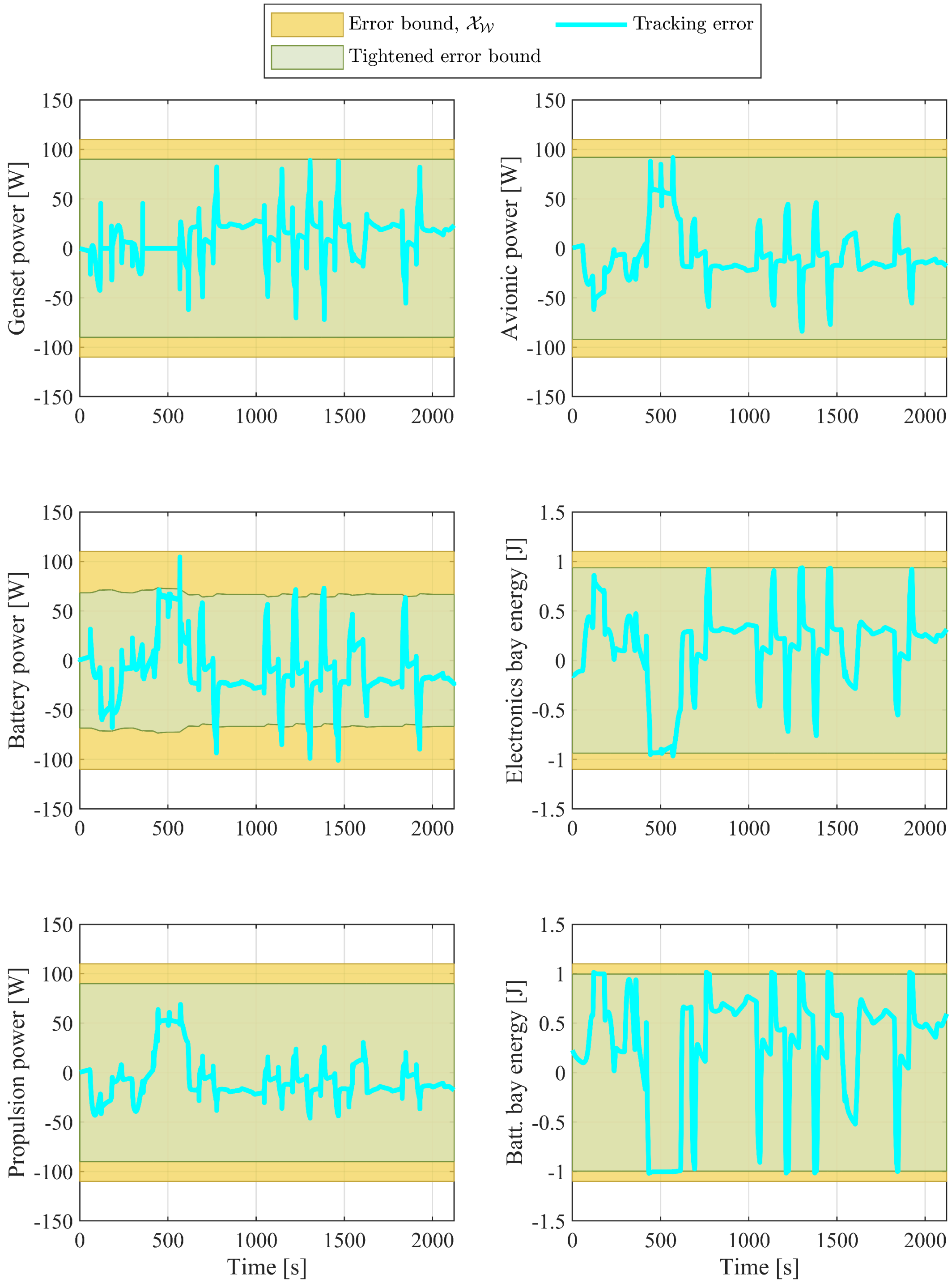


Figure 6.20: Simulated tracking error for hybrid UAV PPTS online re-planning scenario. RMPC enables satisfaction of error bound to ensure the hybrid UAV PPTS mission plan obtained via online re-planning is feasible by tightening the error bound.

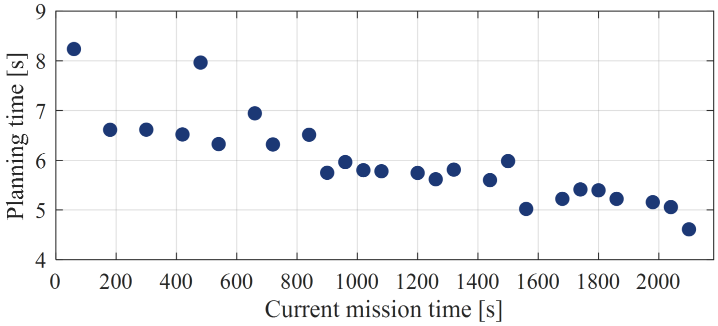


Figure 6.21: Planning computation time for hybrid UAV PPTS online re-planning. Plotted in terms of the current mission time (i.e., the time at which re-planning was initiated).

both energy and thermal management for the full mission at a rate >100X faster than the time taken to perform the mission.

6.3. HYBRID UAV POWERTRAIN EXPERIMENTAL CASE STUDY

In the final case study, the utility of this method for real-time operation of IPPT systems is demonstrated through experimental validation on a hybrid UAV powertrain testbed. Results of this case study demonstrate that, in both simulation and experiment, the two-stage approach employing sampling-based planning methods enables safe operation of the powertrain under complex constraints. Three scenarios are presented. The first two scenarios use the RRT algorithm for rapid offline planning of the powertrain and RMPC for tracking of planned trajectories. The third scenario demonstrates the ERRT algorithm for online re-planning in the presence of uncertain, varying constraints.

6.3.1. Testbed description

The series hybrid electric UAV powertrain experimental testbed was shown in Figure 3.2a). A detailed description of the testbed, including model validation, was presented in [60]. A brief overview is provided here for completeness. The five systems comprising the testbed, shown in Figure 6.22, are the battery, genset, brake, drivetrain, and DC-DC (avionic load). The battery, a

16S7P pack of Samsung 18650 cells with a rated capacity of 21Ah, provides electrical energy storage. The genset consists of a Briggs and Stratton 19N1 internal combustion engine, 16 pole outrunner brushless DC motor (BLDC) starter/generator (S/G), and an inverter. The drivetrain converts DC electrical power from the main bus to AC electrical power to drive the motor. This system consists of a 16-pole outrunner BLDC propulsion (referred to as prop) motor and an inverter. The braking system emulates torque demands of a propeller fixed to the motor shaft. This system consists of a 16-pole outrunner BLDC dynamometer (referred to as dyno or brake), active rectifier, filter box, high power load bank, DC-DC converter, low power (LP) battery pack, and power supply. The DC-DC system represents background electrical loads, avionics, and sensors onboard the aircraft. This system is composed of an electronic load, DC-DC converter, and filter box. The powertrain can source about 9-10kW of electrical power (6kW from the battery and 3kW from the engine), representative of a size class 2 UAV.

Three electronic speed controllers (ESCs) facilitate power conversion among the subsystems and control of the power flows. Although each ESC has the same circuit topology, they all have different operating modes and control strategies, able to act as inverters, active rectifiers, or DC-DC buck-boost converters. Each ESC has two three-phase bridge converter circuits, allowing a single ESC to perform multiple functions on separate branches. The DC-DC ESC acts as a buck-boost converter to control the current output to each of the three output branches. One branch of the hybrid ESC acts as an inverter to control the speed of the propeller motor. Another branch of the hybrid ESC controls the generator motor to start the engine or control the motor torque if the engine is on. One branch of the brake ESC, acting as an active rectifier, controls the torque of the dynamometer. The other branch of the brake ESC acts as a DC-DC converter to ensure the DC current from the rectifier is dissipated by the load bank. Each ESC provides measurements or

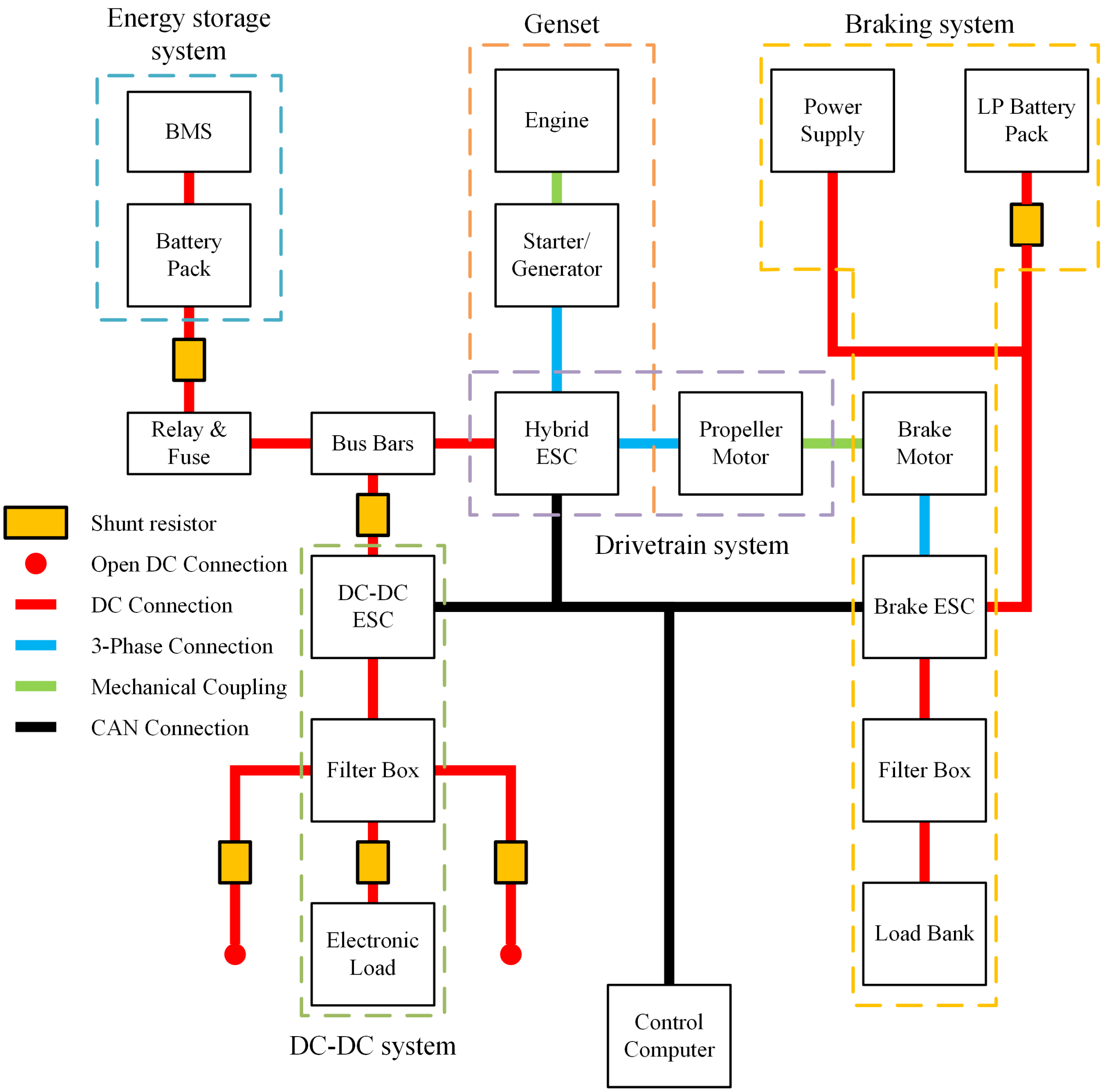


Figure 6.22: Hybrid electric UAV powertrain testbed layout. From [60].

estimates of internal states (e.g. voltages, currents, speeds, etc.). Shunt resistor measurements are obtained using a National Instruments CompactDAQ (cDAQ) with NI-9205 and NI-9403 cards.

A control computer communicates with the testbed to send and receive CAN signals and record shunt resistor measurements. This computer is a desktop PC with a 4.2GHz Intel i7

Processor and 16GB of RAM. A custom LabVIEW application, which communicates with the hardware at 10Hz, provides real-time feedback to safely monitor system operation. This application allows the user to control the testbed using a Simulink file to evaluate control strategies in a hardware-in-the-loop configuration.

To actuate the hardware, the LabVIEW drivetrain application provides command signals to the ESCs and electronic load as listed in Table 6.10. The propulsion motor speed, electronic load current, and genset current commands are decided by the energy management strategy. The electronic load voltage, treated as a disturbance in the graph model shown in Figure 3.3, is constant in this work. The torque command of the dynamometer is obtained from a model of the vehicle dynamics, described in [60], which is also used to simulate the vehicle speed. Measurements from the hardware provide information regarding a select few states of the system model. Relevant measurements from the testbed are described by Table 6.11, including the corresponding vertices of the graph model shown in Figure 3.3. A moving horizon estimator (MHE) [98] is used to estimate the full-order system state based on the measurements. The MHE is described in Appendix C.1.

Energy management controllers run in Simulink, sending commands to LabVIEW over a UDP connection. To limit the computational burden on the control computer and prevent communication delays, intensive calculations are performed on other machines and communicated

Table 6.10: List of commanded signals for hybrid UAV powertrain hardware-in-the-loop testbed.

Commanded signal description [unit]	*Graph model vertex*	*Component*
Propulsion motor speed [rad/s]	5	Hybrid ESC
Electronic load current [A]	6	DC-DC ESC
Genset current [A]	21	Hybrid ESC
Electronic load voltage [V]	28	Electronic load
Dyno torque [A]	-	Brake ESC

over UDP to the control computer. In this work, this is achieved by running each computationally intensive routine, including the RMPC, MHE, and ERRT, in a separate MATLAB/Simulink instance as shown in Figure 6.23, allowing all routines to run concurrently. Each routine communicates over a UDP connection to a central Simulink file, denoted as ‘Comms’ in Figure 6.23, which coordinates communications between the routines running in separate MATLAB instances as well as LabVIEW.

Table 6.11: List of relevant outputs from hybrid UAV powertrain hardware-in-the-loop testbed.

Measurement description [unit]	*Graph model vertex*
Propulsion motor speed [rad/s]	5
Vehicle speed [m/s]	4
Genset DC current [A]	21
Battery current [A]	8
Bus voltage [V]	7
Inverter DC current [A]	9
DC-DC converter input current [A]	16
Electronic load current [A]	6

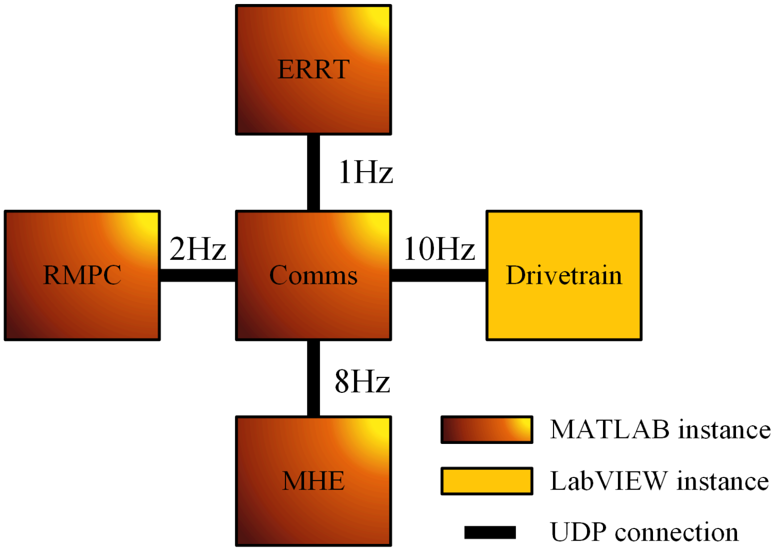


Figure 6.23: Software configuration for hybrid UAV powertrain experimental case studies.

6.3.2. Sampling-based planning problem setup

A brief overview of the setup for the hybrid UAV powertrain sampling-based planning problems is given below. Additional parameters used to solve these planning problems can be found in Appendix C.2.

6.3.2.1. Reduced order model

The four subsystems of the hybrid UAV powertrain reduced order model are listed in Table 6.12, including the edges of the graph model shown in Figure 3.5 corresponding to the power state of each subsystem. In Table 6.12, an asterisk is used to denote those subsystems for which the power state polarity is opposite to the corresponding graph model power flow. As this testbed focuses on the electro-mechanical dynamics, all subsystems are of type 1. The planning model for this system consists of four energy states and four power states corresponding to an eight-dimensional planning problem.

6.3.2.2. Operational constraints

Operational constraints for the hybrid UAV powertrain ensure safe operation of the battery pack and engine energy states. Constraints on the power states of these components, such as maximum genset power and battery charge rate, are enforced through the design of energy primitives so these are omitted from the planning problem formulation. The following constraint

Table 6.12: Description of hybrid UAV powertrain subsystems.

Subsystem number	*Subsystem description*	*Subsystem type*	*Edge(s) corresponding to subsystem power state*
1	Genset	1	11
2	Battery pack	1	1*
3	Propulsion	1	19, 21
4	Avionic load	1	28

ensures that the energy state of the genset remains positive, such that the engine does not run out of fuel.

$$\mathcal{C}_1 = \{X : X_1 \leq 0\text{kJ}\} \tag{6.25}$$

The following operational constraint provides a lower bound for the battery pack's energy state, such that the battery pack is not discharged beyond its capacity.

$$\mathcal{C}_2 = \{X : X_2 \leq 3600 Q V_{ocv}(0.3)\} \tag{6.26}$$

See (6.2) for definitions of Q and V_{ocv}. Constraint (6.26) corresponds to a lower SOC bound of 30%. The following operational constraint provides an upper bound for the battery pack's energy state, which corresponds to 80% SOC.

$$\mathcal{C}_3 = \{X : 3600 Q V_{ocv}(0.8) \leq X_2\} \tag{6.27}$$

6.3.2.3. Trim trajectory parameterization

Six trim trajectories, obtained following the process outlined in Section 4.4.1, were used to construct energy primitives here. These trim trajectories are listed in Table 6.13. The first three trim trajectories charged, sustained charge, and discharged the battery at the same high avionic power and low propulsion power state. Likewise, the second three trim trajectories charged, sustained charge, and discharged the battery at a higher overall power level with low avionic power and high propulsion power. The genset power states of trim trajectories 1 and 5 were selected to

Table 6.13: Trim trajectories for hybrid UAV powertrain case studies.

	Subsystem power state [W]			
Trim trajectory number	*Genset*	*Battery*	*Propulsion*	*Avionic*
1	-2475	500	1100	525
2	-1975	0	1100	525
3	0	-1975	1100	525
4	-2975	500	1700	300
5	-2475	0	1700	300
6	0	-2475	1700	300

optimize fuel efficiency [60]. Zero genset power states of trim trajectories 3 and 6 were achieved by turning off the engine and generator.

6.3.3. RMPC problem setup

The RMPC nominal model of the hybrid UAV powertrain is obtained by linearizing and discretizing the graph-based model shown in Figure 3.3, with a few notable distinctions. First, note that the hybrid ESC of the experimental system controls the generator to track a DC current reference. Without knowledge of the structure and parameters of this control strategy, this dissertation adopts the approach of [60] to model the genset. The genset current state is modeled outside of the graph-based modeling framework using the following first-order linear dynamic

$$\tau \dot{I}_{gen} = -I_{gen} + K_{gen} u_3 \tag{6.28}$$

where τ the time constant and K_{gen} the input gain were identified in [60], and u_3 is an input signal referred to as the genset throttle. The nominal RMPC model combines (6.28) with the linear graph-based model.

Feedback controllers of the hybrid UAV powertrain track commanded signals for the propulsion motor speed, electronic load current, and genset current as described in Table 6.10. In [60], grey-box model identification was performed to identify proportional-integral control parameters that matched the tracking performance of the propulsion motor speed and electronic load current controllers. These identified parameters, listed in Table 6.14, are assumed to define the feedback control strategy for the inverter and DC-DC converter duty cycles in the calculation of linearization error reachable sets. Linear quadratic regulator (LQR) theory is used to define a feedback control gain for the genset throttle in the calculation of linearization error reachable sets.

Reachable sets describing error between the nominal model and true system under the action of this feedback strategy are obtained following the process described in Sections 5.2 and 5.3.

Each time the RMPC problem is solved, the optimal propulsion motor speed and electronic load current states are passed as commanded signals to the respective ESCs. The genset throttle input is modified from the nominal input as in (5.9) using the optimal LQR gain. To command the hybrid ESC, the modified genset throttle input is used to determine a genset current command as follows.

$$I_{ref} = K_{gen} u_{gen} \tag{6.29}$$

To ensure safe operation, RMPC applies constraints to certain states of the full-order model. These constraints are summarized in Table 6.15. Additional RMPC parameters can be found in Appendix C.3.

Table 6.14: Parameters of feedback control assumed to calculate error sets in RMPC for hybrid UAV powertrain.

Input description	*Regulated state*	*Gains*	*[Upper bound, lower bound]*
Inverter duty cycle	Prop speed	K_P=1.9E-3, K_I=3.0E-3	[0.3,0.7]
Electronic load DC-DC converter duty cycle	Electronic load current	K_P=3.9E-5, K_I=2.1E-1	[0.01,0.99]
Genset throttle	-	Determined by LQR	[0,1]

Table 6.15: RMPC state constraints for hybrid UAV powertrain.

State description [unit]	*Slack variable weight*	*Lower bound*	*Upper bound*
Battery state of charge [-]	1E1	0.3	0.9
Propulsion motor speed [rpm]	1E-5	1500	5000
Vehicle speed [m/s]	1E-3	1	100

6.3.4. Offline planning results

Two sets of experimental results are presented here to demonstrate offline planning of the hybrid UAV powertrain using the two-stage approach in different scenarios. The first scenario focused on rapid generation of long-term mission plans when few constraints are active. The second scenario demonstrated the ability to plan feasible mission plans considering more complex constraint regions. In each experiment, a unique set of task-specific constraints was considered in addition to the operational constraints listed in Section 6.3.2.2. Similar to the hybrid UAV PPTS case study, the task-specific constraints restrict the operation of the genset and avionic subsystems when the propulsion energy state falls within a particular range. Both experiments used the RRT algorithm to generate a mission plan offline and used RMPC to track the mission plan. The experimental and simulation results demonstrate RMPC's ability to track the mission plan within the error bound. Efficient computation of long-term mission plans is demonstrated in each scenario, and the two scenarios' computational performances are compared.

6.3.4.1. Scenario 1: long-term planning

In the first offline planning scenario, a relatively simple mission is considered with only two task-specific constraints to demonstrate the computational efficiency of long-term planning using RRT. The task-specific constraints considered here restrict the operation of the genset and avionic subsystems during segments of the mission based on the vehicle location, as illustrated in Figure 1.2. These constraints consider the energy state X_3 of the propulsion subsystem as a proxy for the vehicle location. The first task-specific constraint is a quiet zone which forbids engine operation to reduce noise produced by the vehicle. The second task-specific constraint forbids low avionic power, allowing the avionic subsystem to support high power sensors. These task-specific constraints are summarized in Table 6.16. For this initial case study, transitions between trim trajectories consisted of multiple phases with a maximum ramp rate of 50W/s and the time duration of each trim trajectory was 60s.

The initial condition of the reduced order model, $X_0 = [E_0^T \quad P_0^T]^T$, is given below. The initial energy state E_0 corresponds to full engine fuel capacity, 60% battery pack SOC, and zero energy use by the propulsion and avionic subsystems. The initial power states P_0 correspond to low speed, high avionic power operation with the battery charging.

$$E_0 = [19.5\text{MJ} \quad 2.83\text{MJ} \quad 0 \quad 0]^T \tag{6.30}$$

$$P_0 = [-2.48 \quad 0.500 \quad 1.10 \quad 0.525]^T \text{kW} \tag{6.31}$$

The goal region is specified as an interval for each energy state as given below.

$$\begin{aligned} \mathcal{X}_G = \{ X : 10.2\text{MJ} \le X_1 \le 13.8\text{MJ}, 2.87\text{MJ} \le X_2 \le 3.90\text{MJ}, \ldots \\ 3.68\text{MJ} \le X_3 \le 4.73\text{MJ}, 0.935\text{MJ} \le X_4 \le 1.985\text{MJ} \} \end{aligned} \tag{6.32}$$

This region corresponds to 60-75% battery SOC. In this scenario the goal genset, propulsion, and avionic energy state regions are not intended to reflect any particular mission requirement. Parameterizing the goal region by genset, propulsion, and avionic energy states demonstrates that the goal region could be chosen to specify a desired final fuel capacity, vehicle travel distance, and/or energy provided to sensors or avionic equipment.

In less than one second, the RRT algorithm generated a mission plan, shown by the pink reference trajectory in Figure 6.24 and Figure 6.25, which ensured satisfaction of all constraints.

Table 6.16: Task-specific constraints for scenario 1 of offline planning case study for hybrid UAV powertrain.

Description	*Impacted states*	*Constraint set*
Quiet zone constraint	Genset power, propulsion energy	$\mathcal{C}_4 = \{ X : 1.00\text{MJ} \le X_3 \le 1.20\text{MJ},\ X_5 \le -0.05\text{kW} \}$
Sensing mode constraint	Avionic power, propulsion energy	$\mathcal{C}_8 = \{ X : 2.00\text{MJ} \le X_3 \le 2.80\text{MJ},\ X_8 \le 0.32\text{kW} \}$

Figure 6.24a) shows that the planned trajectory satisfies task-specific constraints $\mathcal{C}_4$ and $\mathcal{C}_5$ and lands in the goal region. Figure 6.24b) illustrates the high-level mission requirements imposed by these constraints. To satisfy the quiet zone constraint, the planner brings the genset power state to zero such that the UAV operates on purely electric power. This can be seen in Figure 6.25a), where the vertical portion of the trajectory shows the battery energy depleting while the genset energy remains constant. Despite this depletion of the battery charge, Figure 6.25a) shows that the battery

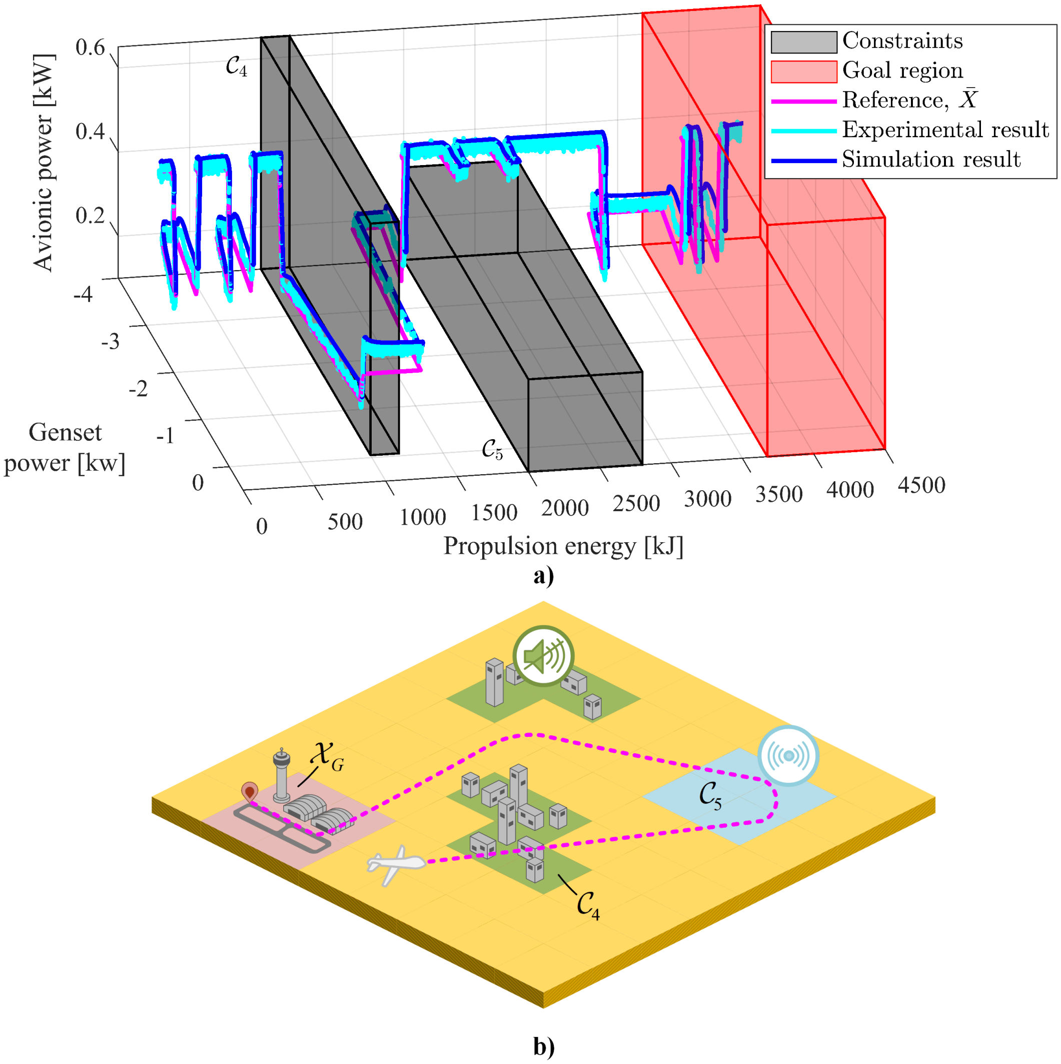


Figure 6.24: Task-specific constraints for hybrid UAV powertrain offline planning scenario 1. Subplot a) shows mission specifications and trajectories. Subplot b) shows the high-level mission requirements (quiet zone, sensing mode, goal region) corresponding to these mission specifications.

energy state does not violate operational constraints $\mathcal{C}_{2,3}$ such that the battery charge remains in a safe region. Moving to the left from the vertical trajectory segment in Figure 6.25a), the battery energy increases as the genset energy decreases, showing that the genset's energy is used to charge the battery to reach the goal region. Figure 6.24b) shows that the propulsion and avionic energy states reach the goal region. Both subplots of Figure 6.24 show the goal region at the beginning of the mission, as well as the 'shrunken' goal region which accounts for tracking error. The terminal point of the reference trajectory, shown as a cross, lands at the edge of the 'shrunken' goal region. The closed-loop results, shown in cyan for the experimental system and in blue for the simulation model, exhibit tracking error from the planned trajectory, but because the planner accounts for this tracking error, the closed-loop trajectories remain feasible in both simulation and experiment.

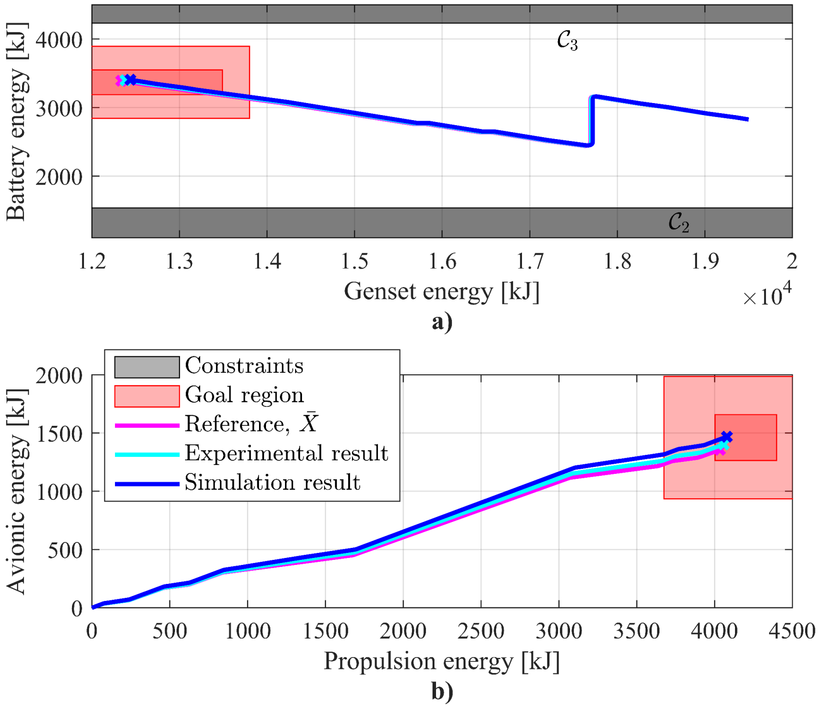


Figure 6.25: Operational constraints and goal region are satisfied in hybrid UAV powertrain offline planning scenario 1. Subplot a) shows that the genset and battery energy states satisfy the operational constraints and land in the goal region. Subplot b) shows that the avionic and propulsion energy states land in the goal region. Both subplots show the original goal region and 'shrunken' goal region, which accounts for tracking error over the full course of the mission.

Figure 6.24 and Figure 6.25 both show that tracking error in the simulation model and experimental system causes some deviation from the planned trajectory, but these closed-loop trajectories remain feasible, satisfying all constraints and landing correctly in the terminal region. RMPC ensures that this is the case by enforcing a tracking error bound on all of the power states. The tracking performance of RMPC is shown in Figure 6.26 and its satisfaction of the tracking error bound is shown in Figure 6.27. Figure 6.26, showing the time trajectories of all power states, demonstrates how the subsystem power states satisfy the mission specifications shown in Figure 6.24 and Figure 6.25. In particular, the genset power state in Figure 6.26a) is held at zero from 740-980s to satisfy constraint $\mathcal{C}_4$. Subsequently, the genset power remains nonzero to prevent depletion of the battery. The battery power in Figure 6.26b) remains nonnegative for the majority of the mission, other than the time period from 740-980s during which the genset power is zero to satisfy constraint $\mathcal{C}_4$. This corresponds to charging the battery to increase its SOC to reach the goal region shown in Figure 6.25a). The avionic power state remains above 0.4kW from 1210-2430 to satisfy constraint $\mathcal{C}_5$.

6.3.4.2. Scenario 2: complex constraint regions

To demonstrate the RRT algorithm's ability to navigate complex constraint regions efficiently and safely, this scenario considers a more taxing mission consisting of a variety of task-specific constraints in addition to the operational constraints listed in Section 6.3.2.2. To enable the planner to navigate these complex regions, the transitions of energy primitives are parameterized to enable quicker and more frequent changes in trim trajectories. Particularly, the maximum ramp rate during transitions is 95W/s and the time duration of each trim trajectory is 20s. The task-specific constraints include both location-based constraints parameterized by

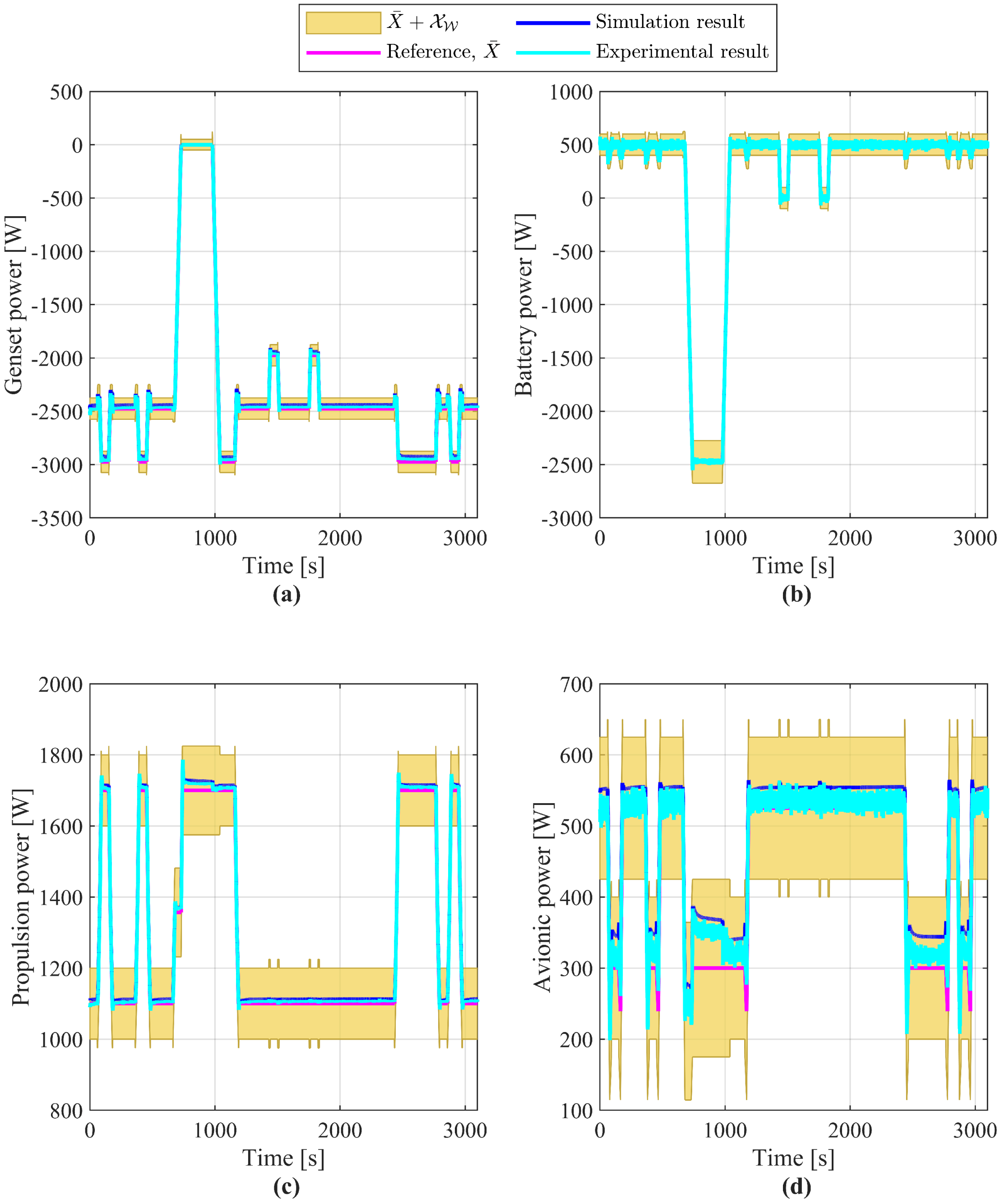


Figure 6.26: Power references, states, and error bounds for hybrid UAV powertrain offline planning scenario 1. RMPC ensures that the power states track the reference trajectory from the planner.

propulsion energy, similar to those considered in other case studies, and more generic constraints that are parameterized by other states in the energy domain. The full set of task-specific constraints is listed in Table 6.17.

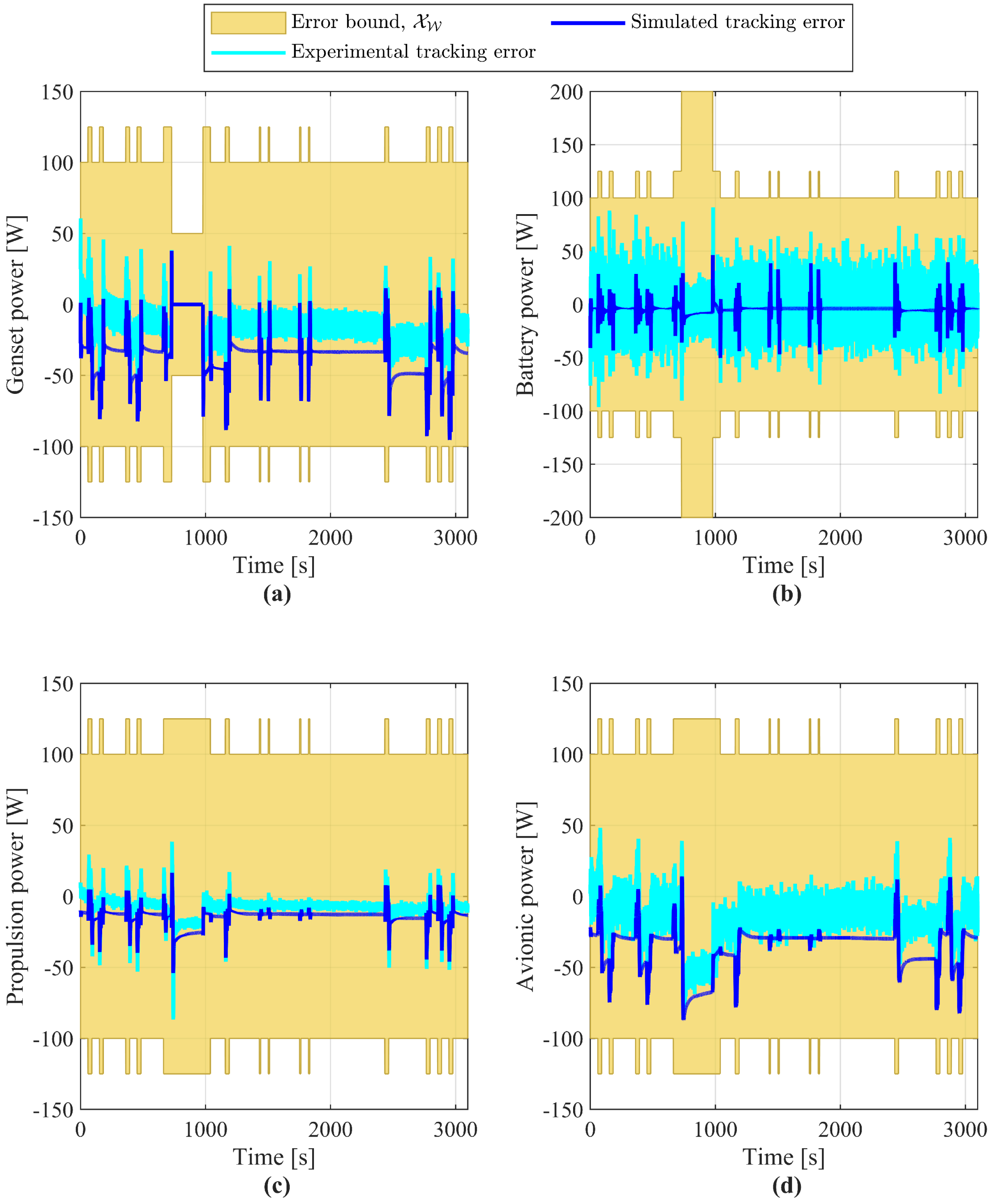


Figure 6.27: Experimental and simulated tracking error for hybrid UAV powertrain offline planning scenario 1. RMPC enables satisfaction of error bound to ensure the offline mission plan for hybrid UAV powertrain is feasible in Scenario 1.

Table 6.17: Task-specific constraints for scenario 2 of offline planning case study for hybrid UAV powertrain.

Impacted states	*Constraint set*
Genset energy, battery energy	$\mathcal{C}_4 = \{X : 18.3\text{MJ} \le X_1 \le 18.8\text{MJ},$ $2.16\text{MJ} \le X_2 \le 2.34\text{MJ}\}$
Propulsion energy, avionic energy	$\mathcal{C}_5 = \{X : 1.00\text{MJ} \le X_3 \le 1.15\text{MJ},$ $0.116\text{MJ} \le X_4 \le 0.266\text{MJ}\}$
Propulsion energy, genset power	$\mathcal{C}_6 = \{X : 0.500\text{MJ} \le X_3 \le 0.600\text{MJ},\ X_5 \le -0.05\text{kW}\}$
Propulsion energy, genset power	$\mathcal{C}_7 = \{X : 0.975\text{MJ} \le X_3 \le 1.18\text{MJ},\ X_5 \le -3.0\text{kW}\}$
Propulsion energy, avionic power	$\mathcal{C}_8 = \{X : 1.35\text{MJ} \le X_3 \le 1.45\text{MJ},\ X_8 \le 0.32\text{kW}\}$
Propulsion energy, avionic power	$\mathcal{C}_9 = \{X : 1.90\text{MJ} \le X_3 \le 2.00\text{MJ},\ 0.55\text{kW} \ge X_8\}$

The initial condition of the reduced order model, $X_0 = [E_0^T \quad P_0^T]^T$, is given by (6.30)-(6.31). The goal region is specified as an interval for each energy state as given below. The goal region corresponds to 50-60% battery SOC and a relatively high genset energy state in order to reserve a significant amount of fuel. In this scenario the goal propulsion and avionic energy state regions are not intended to reflect any particular mission requirement.

$$\begin{aligned} \mathcal{X}_G = \{X : 15.5\text{MJ} \le X_1 \le 17.2\text{MJ}, 2.17\text{MJ} \le X_2 \le 2.87\text{MJ}, \ldots \\ 2.30\text{MJ} \le X_3 \le 2.80\text{MJ}, 0.421\text{MJ} \le X_4 \le 1.02\text{MJ}\} \end{aligned} \tag{6.33}$$

In only 8.6s, the RRT algorithm generated a mission plan offline, shown by the pink reference trajectory in Figure 6.28 and Figure 6.29, which ensured satisfaction of all constraints. In Figure 6.28a), the battery energy state does not violate operational constraints $\mathcal{C}_{2,3}$ and the task-specific constraint $\mathcal{C}_4$ is avoided. Figure 6.28b) shows that the task-specific constraint $\mathcal{C}_5$ is avoided by planning the propulsion and avionic energy states appropriately. Both subplots of Figure 6.28 show the goal region at the beginning of the mission, as well as the ‘shrunken’ goal region which accounts for tracking error at the end of the mission. The terminal point of the reference trajectory, shown as a cross, lands at the edge of the 'shrunken' goal region. The closed-loop results, shown in cyan for the experimental system and in blue for the simulation model, exhibit tracking error from the planned trajectory, but because the planner accounts for this tracking error, the closed-loop trajectories remain feasible in both simulation and experiment. Figure 6.29 shows that the planned trajectory satisfies constraints for the propulsion energy, genset power, and avionic power states. Note that in Figure 6.29, constraints $\mathcal{C}_6$ and $\mathcal{C}_8$ are similar to the quiet zone and sensing mode constraints, respectively, considered in Section 6.2.1.2. Likewise, constraints $\mathcal{C}_7$ and $\mathcal{C}_9$ are similar to the genset and avionic limited constraints, respectively.

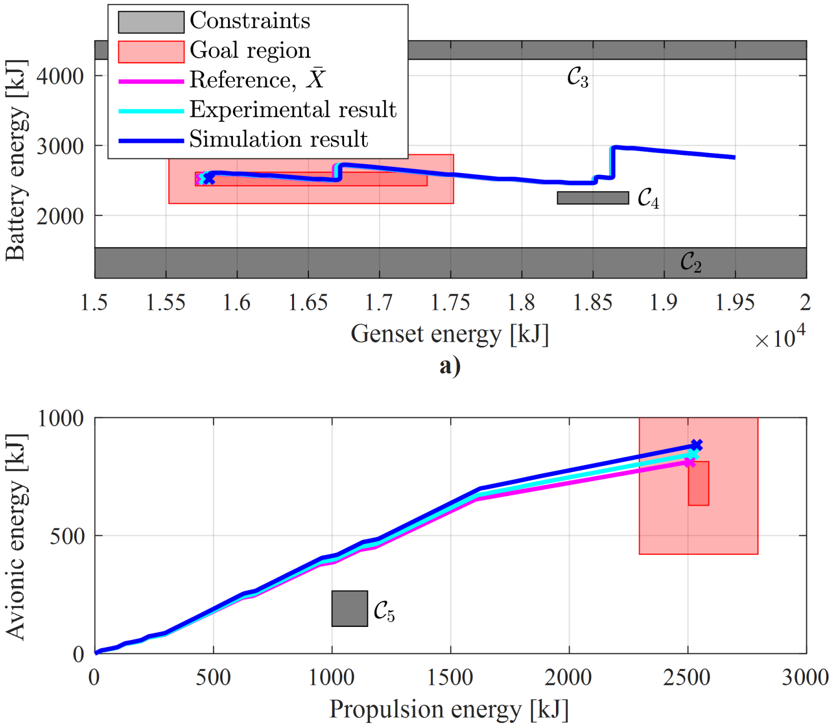


Figure 6.28: Operational constraints and goal region are satisfied in hybrid UAV powertrain offline planning scenario 2.. Subplot a) shows that the genset and battery energy states satisfy the operational constraints and task-specific constraint $\mathcal{C}_4$. Subplot b) shows that the avionic and propulsion energy states satisfy constraint $\mathcal{C}_5$. Both subplots show the original goal region and 'shrunken' goal region, which accounts for tracking error over the full course of the mission.

Figure 6.28 and Figure 6.29 both show that tracking error in the simulation model and experimental system causes some deviation from the planned trajectory, but these closed-loop trajectories remain feasible, satisfying all constraints and landing correctly in the terminal region. RMPC ensures that this is the case by enforcing a tracking error bound on all of the power states. The tracking performance of RMPC is shown in Figure 6.30 and its satisfaction of the tracking error bound is shown in Figure 6.31. Figure 6.30, showing the time trajectories of all power states, demonstrates how the genset and avionic power states satisfy the constraints shown in Figure 6.29. In particular, the genset power state is held at zero from 360-550s to satisfy constraint $\mathcal{C}_6$. Subsequently, the genset power remains well below -3kW until 1350s, which ensures satisfaction of constraint $\mathcal{C}_7$. The avionic power state remains above 0.4kW from 960-1350s to satisfy constraint $\mathcal{C}_8$ and remains below 0.4kW after 1370s to satisfy constraint $\mathcal{C}_9$.

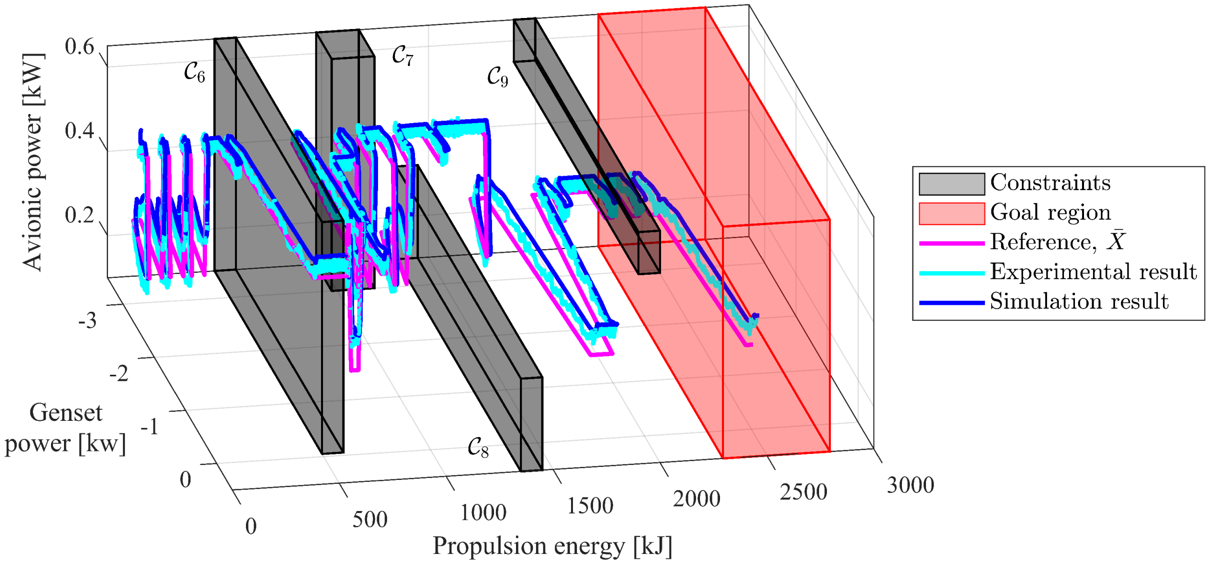


Figure 6.29: Task-specific constraints for hybrid UAV powertrain offline planning scenario 2.Offline planning and control results for hybrid UAV powertrain satisfy constraints for genset power and avionic power states and land in the goal region.

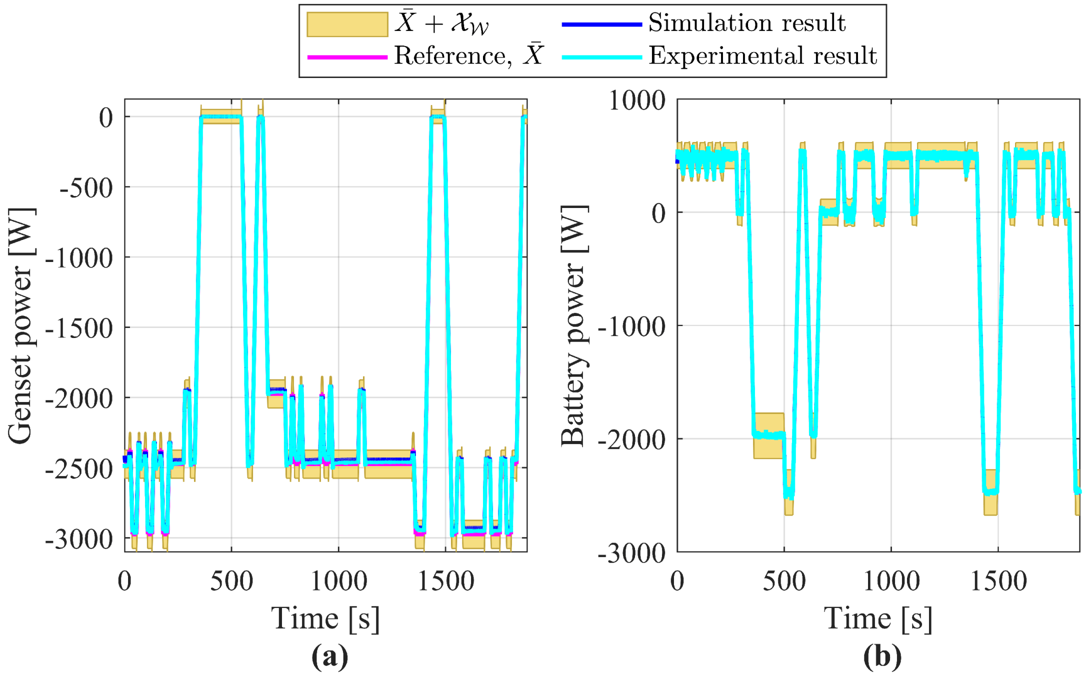


Figure 6.30: Power references, states, and error bounds for hybrid UAV powertrain offline planning scenario 2. RMPC ensures that the power states track the reference trajectory from the planner.

6.3.4.3. Computational performance comparison

The two scenarios described above demonstrate the utility of the proposed method for rapid, long-term planning in experimental validation, but the second scenario required more planning

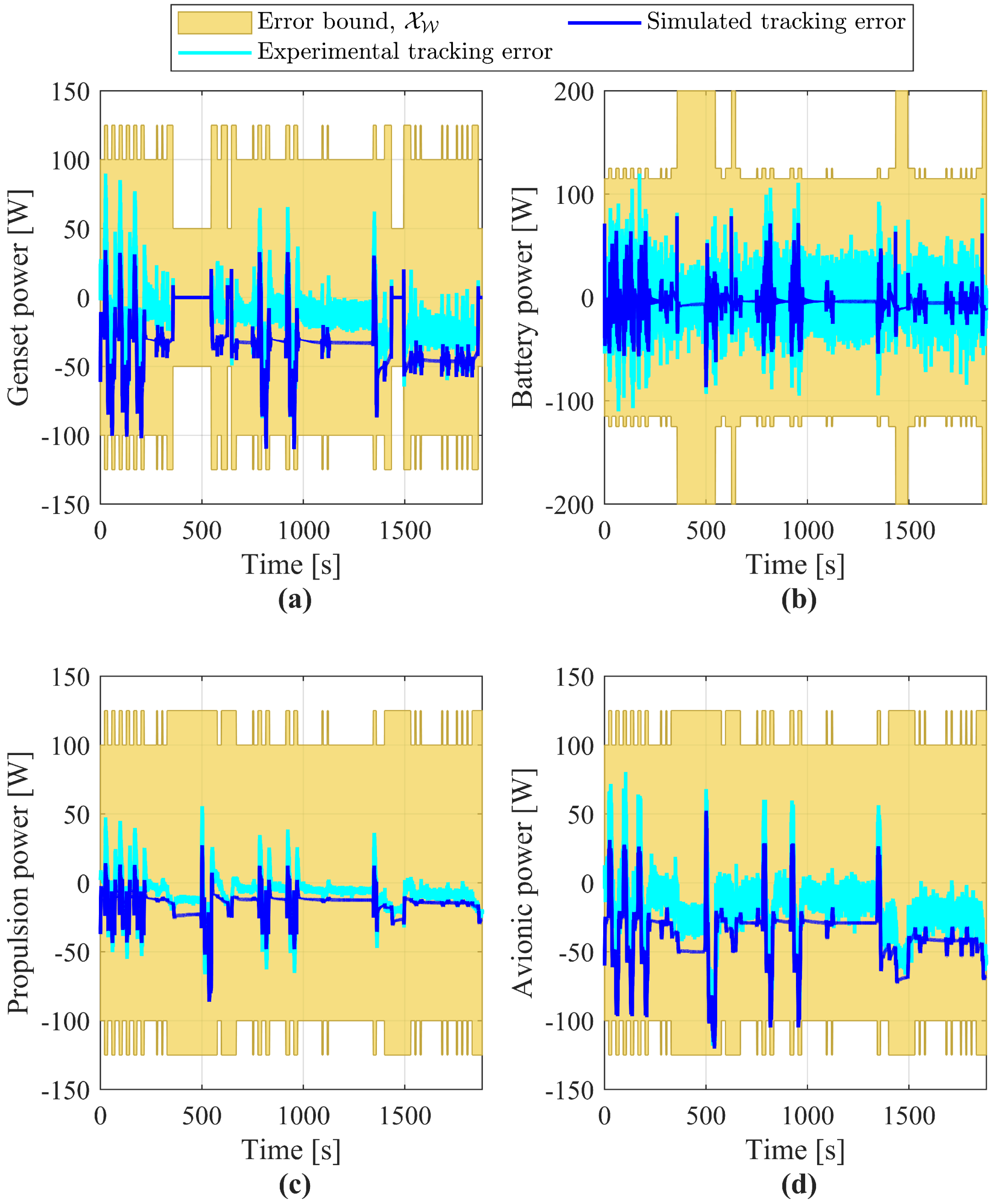


Figure 6.31: Experimental and simulated tracking error for hybrid UAV powertrain offline planning scenario 2. RMPC enables satisfaction of error bound to ensure the offline mission plan for hybrid UAV powertrain is feasible in Scenario 2.

time. Figure 6.32 provides additional statistics regarding the offline planning stage in these two scenarios. To account for the randomized nature of the RRT algorithm, results presented in Figure 6.32 summarize a series of 100 trials of each planning problem. While planning times for both scenarios are small in comparison to time durations of the missions, it is notable that computation times for scenario 2 are considerably higher than scenario 1. One reason for this increase in computation time stems from the fact that reducing the time duration of trim trajectories led to the inclusion of more energy primitives in the search tree for scenario 2. This is shown in Figure 6.32 as an increase in the number of nodes required to find a feasible path. Another reason for this increase in computation time is the increased number of constraint violation checks required for each trajectory segment. Note that the time durations of the missions shown in Section 6.3.4.1 for scenario 1 and in Section 6.3.4.2 for scenario 2 were >3000s and >1800s, respectively. Even the largest planning times reported in Figure 6.32 remain <1/1000th and <1/100th of the time taken to perform the missions corresponding to scenario 1 and scenario 2 respectively. These results are promising for implementation of the algorithm in a wide range of practical applications.

6.3.5. Online re-planning results

This section provides experimental and simulation-based demonstrations of the two-stage method for online re-planning of the hybrid UAV PPTS using the problem setup described above

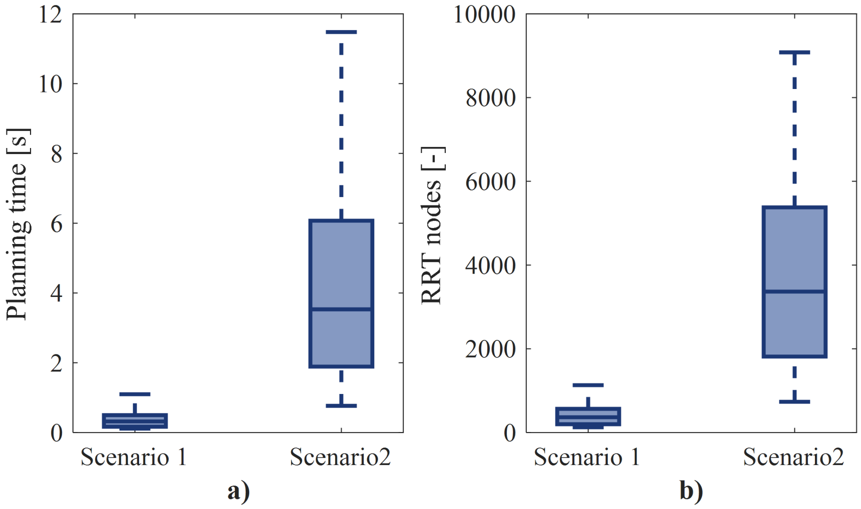


Figure 6.32: Computational statistics of RRT during offline planning scenarios 1 and 2 for hybrid UAV powertrain. Subplot a) shows computation time and b) number of RRT nodes required to find a feasible solution.

[99]. The ERRT strategy described in Section 4.3.2 was used for online re-planning using a custom MATLAB code. To solve the RMPC optimization problem described in Section 5.4, CORA [91] was used to calculate linearization error reachable sets, the YALMIP toolbox [96] was used to formulate the optimization problem, and Gurobi [95] was used to solve the optimization problem online. For the simulation demonstration, all computations were performed on an Intel i7 processor with 32GB RAM. For the experimental demonstration, ERRT, MHE, and RMPC were run in concurrent MATLAB instances on the same machine with an Intel i7 CPU and 16GB RAM.

The initial condition of the reduced order model, $X_0 = [E_0^T \quad P_0^T]^T$, is given below.

$$E_0 = [19.5\text{MJ} \quad 3.37\text{MJ} \quad 0 \quad 0]^T \tag{6.34}$$

$$P_0 = [-2.48 \quad 0.500 \quad 1.10 \quad 0.525]^T \,\text{kW} \tag{6.35}$$

The initial energy state E_0 corresponds to full engine fuel capacity, 70% battery pack SOC, and zero energy use by the propulsion and avionic subsystems. The initial power state P_0 corresponds to low speed, high avionic power operation with the battery charging.

In addition to the operational constraints described in Section 6.3.2.2 and initial condition described above, mission information for this scenario included task-specific constraints and a goal region that were parameterized based on the vehicle's location. Task-specific constraints in this study restricted various levels of genset power and avionic power to represent physically meaningful mission specifications, similar to the mission depicted in Figure 1.2. The five task-specific constraints considered in this case study are described in Table 6.18. Note that three constraints ($\mathcal{C}_{4-6}$) are initially known, while constraints $\mathcal{C}_{7,8}$ became known approximately 2 minutes and 19 minutes into the mission, respectively. Each of these constraints were active at precise lateral distances from the vehicle's initial location, and these distances were used to estimate the required propulsion energy state using a linear mapping $\hat{X}_3(x)$. The goal region, listed in Table 6.18, was also location-based, requiring 23.5 miles of travel and 40-60% battery SOC at the end of the mission.

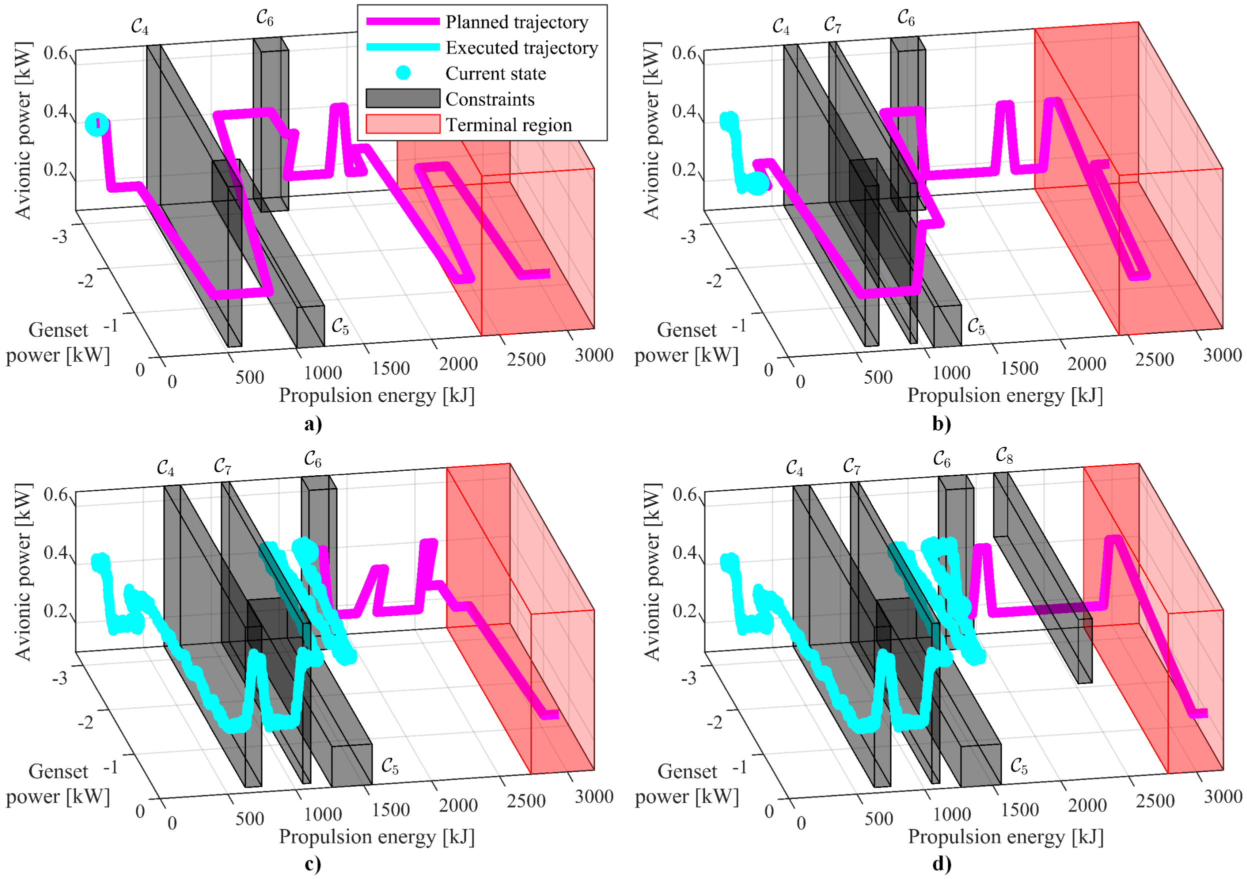
The ability to rapidly react to uncertain, varying mission specifications was demonstrated using the ERRT algorithm. Two sources of uncertainty considered were incomplete and changing information. Incomplete information took the form of initially unknown task-specific constraints that arose as the mission progressed. Mission information changed as the task-specific constraints and goal region shifted continuously during the mission. As the mission progressed, the distance travelled by the UAV was calculated by integrating its horizontal air speed. As in the hybrid UAV PPTS case study discussed in Section 6.2, horizontal air speed was obtained by subtracting wind speed, obtained from a local weather station [97], from the horizontal velocity calculated using the graph-based model. In addition to error incurred from the inaccurate linear mapping $\hat{X}_3(x)$,

Table 6.18: Summary of task-specific constraints for hybrid UAV powertrain online re-planning case study.

Description	*Vehicle distance [mi]*	*Detection time [s]*	*Constraint set*
Quiet zone constraint 1	5.2-6.2	0	$\mathcal{C}_4 = \left\{X : \hat{X}_3(5.2) \le X_3 \le \hat{X}_3(6.2), X_5 \le -0.12\text{kW}\right\}$
Sensing mode constraint	10-12	0	$\mathcal{C}_5 = \left\{X : \hat{X}_3(10) \le X_3 \le \hat{X}_3(12), X_8 \le 0.24\text{kW}\right\}$
Genset limited constraint	13-15	0	$\mathcal{C}_6 = \left\{X : \hat{X}_3(13) \le X_3 \le \hat{X}_3(15), X_5 \le -3.0\text{kW}\right\}$
Quiet zone constraint 2	8.5-9	129	$\mathcal{C}_7 = \left\{X : \hat{X}_3(8.5) \le X_3 \le \hat{X}_3(9), X_5 \le -0.12\text{kW}\right\}$
Avionic limited constraint	17-18	1129	$\mathcal{C}_8 = \left\{X : \hat{X}_3(17) \le X_3 \le \hat{X}_3(18), 0.43\text{kW} \le X_8\right\}$
Goal region	23.5	0	$\mathcal{X}_G = \{X : \hat{X}_3(23.5) \le X_3, 3600QV_{ocv}(0.4) \le X_2, \ldots$ $X_2 \le 3600QV_{ocv}(0.6)\}$

the wind speed disturbance shown in Figure 6.14 caused the mission specifications' propulsion energy state estimates to continuously shift.

An experimental demonstration of online re-planning to react to changing mission specifications is shown in Figure 6.33, where four subplots show the task-specific constraints and goal region, as well as the planned and executed trajectories, at four distinct time instances. Figure 6.33a) shows the initial condition and the initially planned trajectory, which was feasible with respect to initially known constraints $\mathcal{C}_{4-6}$. After two minutes passed, constraint $\mathcal{C}_7$ was detected. Figure 6.33b) shows the re-planned trajectory at 162s after constraint $\mathcal{C}_7$ was added to the mission specifications. Because constraint $\mathcal{C}_7$ limited operation of the genset, the re-planner avoided this



a)
b)
c)
d)

Figure 6.33: Task-specific constraints, goal region, and trajectories of hybrid UAV powertrain during re-planning experiment with ERRT. Subplots show planned and experimental trajectories at different time instances: a) t=0s, b) t=162s, c) t=1126s, d) t=1270s.

constraint to remain feasible by keeping the genset power state at zero. Figure 6.33c), taken after 18 minutes passed, shows the shifting of constraints and terminal region due to the wind disturbance and the updated trajectory to ensure feasibility despite these changes in mission specifications. Finally, Figure 6.33d), taken at 1270s after constraint $\mathcal{C}_8$ became known, shows the updates to ensure feasibility. Particularly, the planned avionic power state remained at a lower level to respect the final constraint.

These results show satisfaction of the constraints in terms of the planner which reasons only about the energy and power states, while the task-specific constraints are specified in terms of the distance traveled by the vehicle. Figure 6.34 shows the locations of the task-specific constraints and the final trajectory obtained by re-planning online using ERRT and tracking the re-planned trajectory using RMPC. Figure 6.34 shows that, despite uncertainty in vehicle location

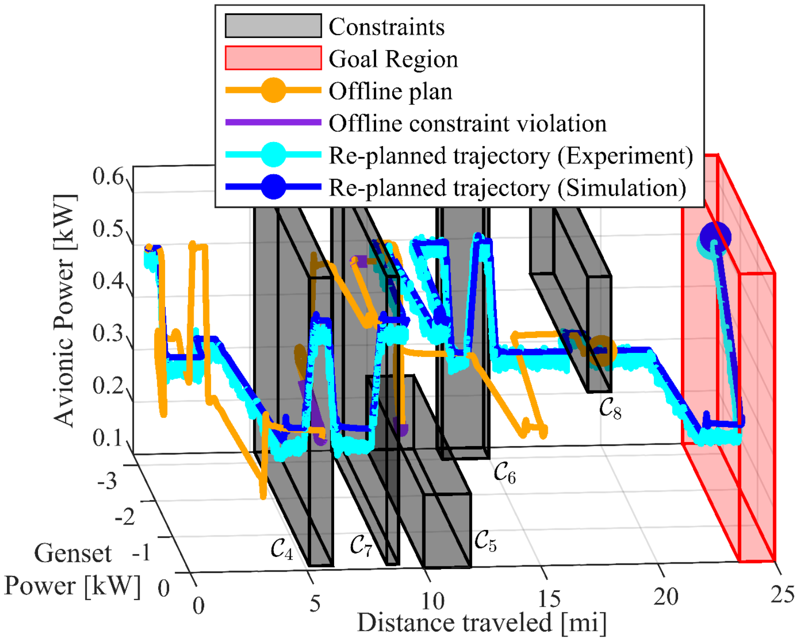


Figure 6.34: Location-based mission specifications for hybrid UAV powertrain are achieved via online re-planning, in both experiment and simulation. While the offline mission plan does not satisfy constraints $\mathcal{C}_{4,5,7}$ and does not land correctly in the goal region, online re-planning assures that the executed trajectory remains feasible with respect to all mission specifications.

and the inaccurate mapping of propulsion energy usage to vehicle location, re-planning online ensured that all task-specific constraints were satisfied. This is compared with a simulation in which RMPC tracked a trajectory planned offline using RRT based on the information known at the beginning of the mission. The offline-planned trajectory violated multiple constraints, including constraints $\mathcal{C}_4$ and $\mathcal{C}_5$ that were initially known, and did not land properly in the terminal region due to inaccurate, incomplete information. In addition to the task-specific constraints, re-planning online enabled satisfaction of the operational constraints, which is shown in Figure C.6 of Appendix C.4. Re-planning using ERRT enables rapid adjustments to changing mission specifications to safely perform the mission.

Figure 6.35 and Figure 6.36 show that, in both experiment and simulation, RMPC ensured satisfaction of the tracking error bound to guarantee feasibility. The genset power state shown in Figure 6.35a) decreased to 0W at 340s and remained there until 780s to respect constraints $\mathcal{C}_4$ and $\mathcal{C}_7$, which are "quiet zone" constraints. From then until 1690s, the magnitude of genset power remained well below its maximum of -3.3kW to respect constraint $\mathcal{C}_6$. The avionic power state increased at 670s to respect the high avionic power requirement imposed by constraint $\mathcal{C}_5$. Finally, the avionic power state decreased at 330s and remained at a low level to respect constraint $\mathcal{C}_8$ shown in Figure 6.33d). As with the offline planning scenarios, Figure 6.35 demonstrates good agreement between simulation model and experimental system in terms of dynamic behavior under the RMPC strategy. A slight offset in steady-state avionic power indicates that more losses are present in the experimental system than are captured by the simulation model. Figure 6.36 provides a magnified view of the tracking error and error bound to show that the error bound is satisfied in both simulation and experiment.

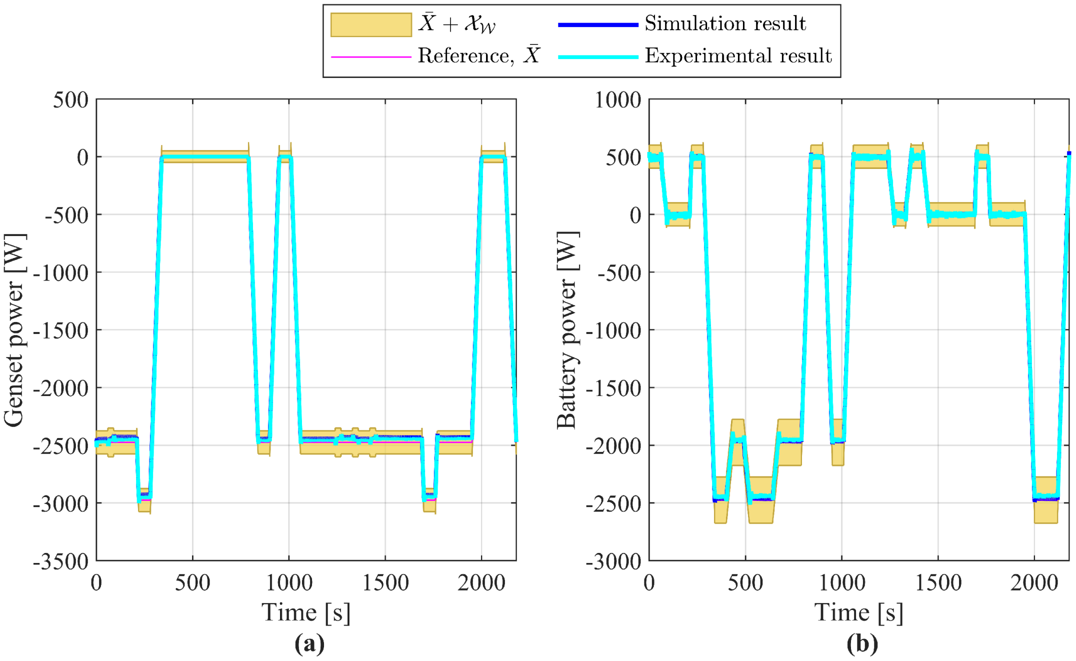
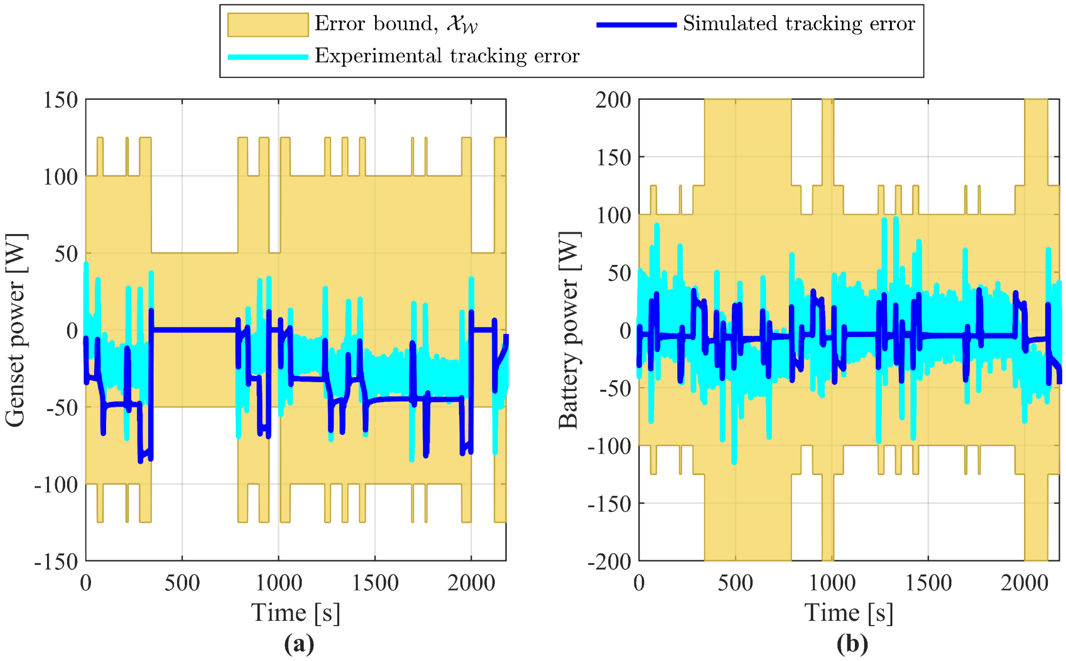


Figure 6.35: Power references, states, and error bounds for the hybrid UAV powertrain online re-planning scenario. RMPC ensures that, in both simulation and experiment, the power states track the reference trajectory from the planner within the error bound.

This re-planning online was made possible by the computationally efficient planner which consistently found feasible mission plans within 6 seconds. The computational performance for this case study is shown by the computation times in Figure 6.37. Computation times for re-



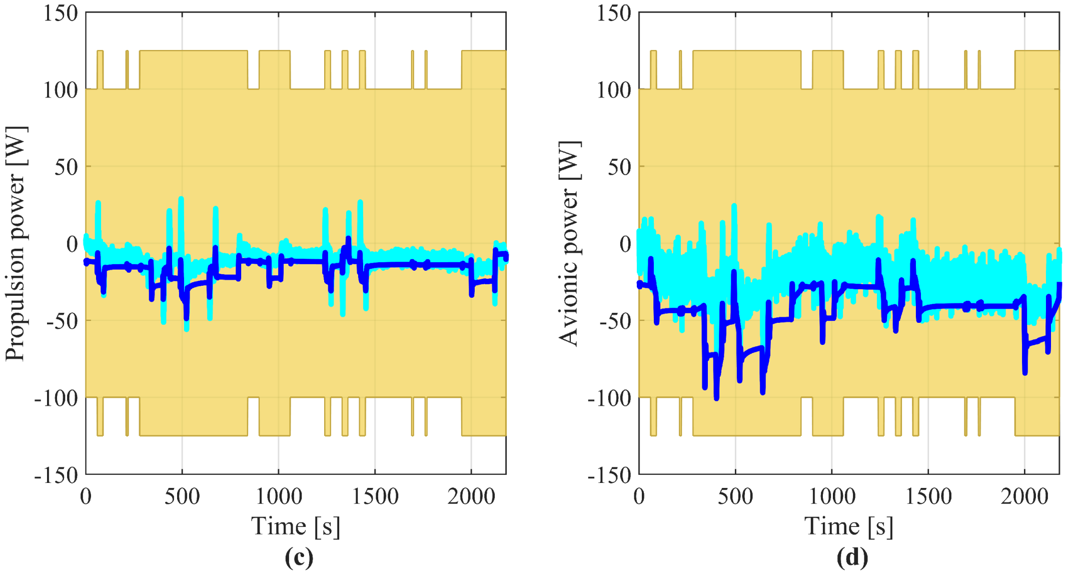


Figure 6.36: Experimental and simulated tracking error for hybrid UAV powertrain online re-planning scenario. RMPC enables satisfaction of error bound to ensure the hybrid UAV powertrain mission plan obtained via online re-planning is feasible.

planning decrease as the mission progresses as the ERRT planner uses waypoints from previous mission plans. To benchmark these computation times, the ERRT was compared to RRT and a genetic algorithm (GA). A genetic algorithm was chosen as the optimization algorithm because gradient-based alternatives cannot accommodate the discrete problem formulation of planning over a finite set of energy primitives, and gradient-based algorithms are ill-suited for these nonconvex problems. The GA was formulated to minimize constraint violations and fuel consumption using the same routines as RRT and ERRT for constraint violation checking, simulation of the planning model, and calculation of fuel consumption. Each of these three algorithms was used to plan a sequence of primitives to satisfy the initial mission scenario shown in Figure 6.33a). While the RRT and ERRT by design found only feasible trajectories, the GA converged to optimized solutions. To account for randomization of these algorithms, 100 trials each of ERRT and RRT and 20 GA trials were performed. The results of these trials are summarized in Table 6.19. ERRT and RRT exhibit similar computational performance, each enabling planning in well under 1/100th of the time required to complete the mission. Computation times are well within the 55s re-planning window to enable rapid reaction to changing constraints. ERRT yields a modest 3% reduction in fuel consumption compared to RRT. The GA, on the other hand, provides significantly lower fuel consumption but takes two minutes on average to find a feasible plan and ten minutes to converge to an optimum. This optimum is not guaranteed to be the global optimum, as the GA trials converged after a number of generations that "stalled" (i.e., did not improve the objective function). While convergence to a global optimum cannot be

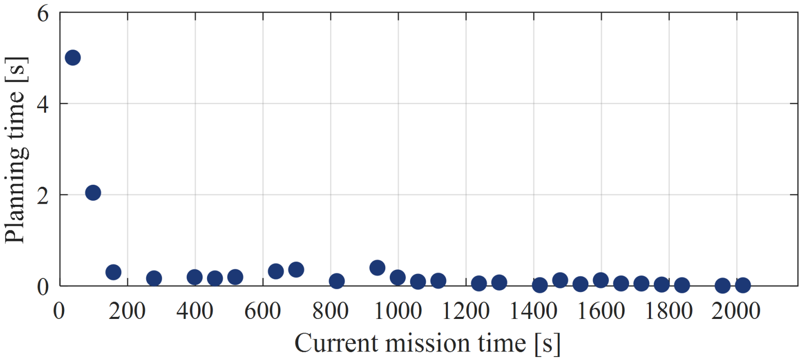


Figure 6.37: Planning computation time for hybrid UAV powertrain online re-planning. Plotted in terms of the current mission time (i.e., the time at which re-planning was initiated).

Table 6.19: Comparison of ERRT, RRT, and GA computation times and fuel consumption for offline planning of hybrid UAV powertrain.

Algorithm	*Average (max) computation times [s]*		*Average fuel consumption [g]*
	Feasible trajectory	*Optimal trajectory*	
ERRT	0.728 (1.85)	-	201
RRT	0.710 (2.82)	-	209
GA	118 (294)	598 (889)	187

guaranteed with this randomized algorithm, ensuring convergence to a global optimum with a high level of confidence would require a very large number of iterations [100]. With such slow computation times, the GA cannot quickly react to changing mission specifications to safely operate the powertrain.

6.4. DISCUSSION AND SUMMARY

In this chapter, three case studies were presented to demonstrate that this two-stage method enables efficient computation of long-term plans for IPPT systems that are assured to satisfy constraints. In the SPS case study of Section 6.1, the computationally efficient two-stage method for energy management has been shown to be scalable to highly complex powertrains with minimal computational increases. In the hybrid UAV PPTS case study of Section 6.2, the method has also been shown to enable simultaneous energy and thermal management for integrated power, propulsion, and thermal systems. Finally, in Section 6.3 this energy management method has been proven to be real-time applicable through experimental implementation on a hybrid UAV powertrain, including offline and online planning. Supplementary material for each of these case studies, including animations of online re-planning for the hybrid UAV systems, are in appendices.

In each case study, sampling-based planning methods efficiently planned missions in nonconvex feasible regions. It was demonstrated that high-level mission requirements, such as quiet operation, emissions-free operation, and low speed operation at certain locations, can be considered in these planning problems that reason only about the energy and power states of a system. Even in complex constraint regions, these sampling-based methods compute feasible mission plans in <1/100th and in some cases <1/1000th of the time taken to perform the mission. Compared to optimization-based planning, the sampling-based planners found feasible trajectories >100X faster in one case study. While it is important to acknowledge that the computational statistics reported in these case study may be dependent on the problem instance and computational resources, computational complexity analyses in the literature have proven the efficiency of these methods. For example, [101] shows that the time complexity of RRT is $\mathcal{O}(n \log n)$ for n samples. The rapid planning capability enables sampling-based planners' use for online re-planning to react to changing mission specifications. Each of the three case studies demonstrated online re-planning which ensured satisfaction of uncertain and changing constraints that offline planning failed to satisfy.

Importantly, the mission plans generated by these methods are safe, accounting for tracking error to assure that all constraints will be satisfied. While the planner manages slower dynamics, RMPC manages fast dynamics by solving an optimization problem online with a shorter time horizon and faster update rate than the planner. The nominal models in the RMPC formulations are generated using the graph-based modeling framework, which facilitates control design for multi-domain systems. RMPC is shown to track the planned trajectories and satisfy the tracking error bound for these case studies. Satisfaction of the error bound is achieved by enforcing a tightened error bound in RMPC. It is important to acknowledge that, while the case studies shown here demonstrated satisfaction of the error bound in all cases, a significant amount of conservatism is used in the calculation of linearization error sets and parameterization of the error bound to ensure that this is the case. In application of these methods to experimental hardware and other physical systems, no amount of constraint tightening can fully guarantee robustness to all modeling error, experimental noise, and other sources of uncertainty.

CHAPTER 7: CONCLUSION

7.1. SUMMARY OF RESEARCH CONTRIBUTIONS

Electrified vehicles' tight integration of power, propulsion, and thermal systems requires energy and thermal management strategies to ensure safety. Multi-domain interactions, multi-timescale dynamics, nonconvex feasible regions and uncertain external conditions are a few of the challenges making safe operation difficult to guarantee. This work provides a two-stage approach to address these challenges.

In Chapter 2, a brief overview of the two-stage approach is presented. To manage the multi-timescale dynamics of IPPT systems, this approach decomposes the energy and thermal management problems into a long-term mission planning stage and short-term tracking stage. Problem formulations for each stage are provided. A novel aspect of the two-stage approach is its use of sampling-based planning methods for long-term mission planning.

Both stages of the two-stage approach rely on an existing graph-based modeling approach, for which a brief overview is provided in Chapter 3. This overview includes a summary of the modeling fundamentals and linearization approaches. The model reduction method employed to develop planning models is presented. Additionally, examples of graph-based system models are presented that are later used in the case studies of Chapter 6.

Chapter 4 presents the sampling-based methods used for long-term planning of IPPT systems. A brief overview of sampling-based methods, which are commonly used in motion planning/robotics applications, is provided. Compared to optimization-based approaches for long-term planning, sampling-based methods have the advantage of increased efficiency in the generation of feasible mission plans and the ability to consider more geometrically complex constraint regions. An RRT algorithm for generating long-term mission plans for IPPT systems is presented. Using a finite set of energy primitives and random sampling, the RRT algorithm generates missions that are dynamically feasible for reduced order models of IPPT systems. This algorithm accounts for bounded error in the reduced-order model due to imperfect tracking control. To enable consideration of changing mission specifications, approaches for online re-planning are discussed which leverage the structure of the energy primitives. An ERRT algorithm for online re-planning of IPPT systems is presented that considers fuel consumption in its generation of mission plans. To enable these primitive-based planning algorithms, a procedure for parameterization of energy primitives is presented.

Chapter 5 presents a robust model predictive control formulation to enable tracking of the planned trajectories while satisfying the tracking error bound. Using a nominal, linearized model of the full order state dynamics, a numerical optimization problem is solved to determine optimal inputs to actuate the IPPT systems. This optimization problem minimizes tracking error between the reference trajectory generated by the planner and the reduced order states obtained using RMPC's nominal model. A filtering strategy is employed to reduce tracking offsets in the reduced order states. Constraint tightening approaches, relying on online computation of reachable sets, provide robustness to linearization error.

Finally, Chapter 6 presents case studies demonstrating the utility of this approach for safe operation of IPPT systems in aerospace and marine applications. First, the two-stage method is applied to enable safe operation of a shipboard power system in both offline and online planning scenarios. These 20-dimensional, nonconvex planning problems with emissions restrictions, speed restrictions, and time-varying constraints are solved using RRT in under 1/1000th of the time taken to perform the mission. Closed-loop control results show that the RMPC strategy effectively satisfies the tracking error bound to guarantee that the mission plan satisfies constraints. In the second case study, the two-stage approach employing RRT and RMPC is used to enable simultaneous energy and thermal management of a hybrid UAV PPTS. Simulation results demonstrate similarly fast computation times during online re-planning. Temperature constraints for the thermal system are shown to be satisfied. The final case study experimentally validates the two-stage approach for energy management of a hybrid UAV powertrain. Offline planning results for the experimental system demonstrate rapid, long-term planning and safe navigation of complex constraint regions. Online re-planning experimental results demonstrate the ability to maintain a feasible mission plan despite changing mission information. In each scenario, RMPC ensures that the experimental system operates safely without constraint violation. Collectively, these case studies demonstrate that the rapid, assured planning achieved using this approach is scalable to complex systems, flexible to manage multi-domain dynamics, and real-time capable.

7.2. FUTURE WORK

While this work provides an initial demonstration of the benefits of sampling-based planning for energy and thermal management of IPPT systems, there are a variety of directions in which this work can be extended to widen its range of application and formalize the theoretical properties. A brief list of potential extensions is provided below.

7.2.1. Theoretical properties of mission planning

The online re-planning case studies in this work demonstrated the ability to quickly re-compute feasible mission plans despite changing constraints. However, it may be possible that online re-planning pushes the system trajectory toward an infeasible region. Analytical studies of the recursive feasibility of these online, energy-primitive-based re-planning methods can increase confidence in this method’s ability safely perform a mission. Further, each case study considered only a finite set of energy primitives, which restricted the system's operational envelope and, consequently, the types of constraint regions that could be accommodated using this framework. It may prove useful to quantify the restrictiveness of planning over such a finite set of energy primitives. Investigating the number and distribution of energy primitives required to solve a given class of problems can widen the application of this method.

7.2.2. Solution quality improvements

Focusing foremost on computational efficiency, many sampling-based methods including the RRT and ERRT considered in this work yield suboptimal mission plans. With fuel efficiency and cost savings at the forefront of many energy and thermal management decisions, extensions of this work should consider methods of improving upon these metrics. This may include using asymptotically optimal sampling-based methods, such as RRT* [76], or using the solution provided by RRT or ERRT to warm-start an optimization algorithm. Additionally, the quality of solutions produced by the mission planner is limited by the conservatism required to guarantee satisfaction of the error bound. Relaxing this guarantee, for example, using a chance-constrained MPC approach [103], may allow increased quality of mission plans and the consideration of more densely packed constraint regions. Finally, solution quality can be improved through the optimized design of energy primitives.

7.2.3. Contingencies for mission infeasibility and plant changes

It may be the case that changing constraints lead to an infeasible planning problem, such that the sampling-based planners fail to find a mission plan within a reasonable timeframe. To assure safety in these situations, contingencies can be introduced that either surrender control of the system to a run-time assured, fail-safe control routine or change the mission information. Focusing on the latter option, if a mission is deemed infeasible, it may be possible to relax the problem to consider only a subset of the original constraints, similar to a load-shedding approach. In addition to reacting to changing mission specifications as considered in this work, safe operation of IPPT systems may require the ability to react to changes in the plant due to fault conditions or external disturbances. Extensions of this work may consider modifications to the planning algorithm to consider changes in the plant model, and to enable detection, isolation, and recovery from fault conditions.

7.2.4. Multi-agent planning problems

The energy and thermal management problems considered in this work focused on single vehicle applications, but in vehicle fleets the cooperative control of energy systems onboard multiple vehicles is an active area of research. Extensions of this work could apply the two-stage approach to enable rapid planning and robust control of the IPPT systems of vehicle fleets.

7.2.5. System design

This dissertation focuses on mission planning and control but another important aspect of IPPT system operation is the system design process. Control co-design for IPPT systems, which integrates control strategies into system design processes, is an active area of research. Incorporating this two-stage approach into control co-design studies can help ensure that the designed system can safely execute a mission.

REFERENCES

[1] M. Dwyer, “Electric vehicles and hybrids make up 16% of U.S. light-duty vehicle sales,” *Today in Energy*. US Energy Information Administration, 2023.

[2] O. Burkacky, J. Deichmann, M. Guggenheimer, and M. Kellner, “Outlook on the automotive software and electronics market through 2030.” McKinsey & Company, 2023.

[3] “FACT SHEET: Biden-Harris Administration Announces New Private and Public Sector Investments for Affordable Electric Vehicles.” The White House [Statements and releases], 2023.

[4] “Multi-Pollutant Emissions Standards for Model Years 2027 and Later Light-Duty and Medium-Duty Vehicles.” US Environmental Protection Agency, Office of Transportation and Air Quality, 2023.

[5] “Top 5 Reasons to Drive Electric.” Drive Clean CA - California Air Resources Board, 2023.

[6] “Electric aircraft market size, growth, trends, report.” Precedence Research, 2023.

[7] “Global electric construction vehicles market overview.” Market Research Future, 2023.

[8] “Electric ship market size.” Custom Market Insights, 2022.

[9] M. Sinnett, “Saving Fuel and Enhancing Operational Efficiencies,” *Aero Quarterly - Boeing Commercial Airplanes*, 2007.

[10] T. J. McCoy, “Trends in ship electric propulsion,” *IEEE Power Engineering Society Summer Meeting*, 2002.

[11] A. Allam, S. Onori, S. Marelli, and C. Taborelli, “Battery Health Management System for Automotive Applications: A retroactivity-based aging propagation study,” *American Control Conference*, 2015.

[12] J. Pollefliet, "Transistor Power Switches," in *Power Electronics*, Elsevier, 2018, pp. 3.1-3.60.

[13] A. Morozumi, K. Yamada, T. Miyasaka, S. Sumi, and Y. Seki, "Reliability of power cycling for IGBT power semiconductor modules," *IEEE Trans. Ind. Appl.*, vol. 39, no. 3, pp. 665–671, May 2003.

[14] H. Wang, M. Liserre, and F. Blaabjerg, "Toward reliable power electronics: Challenges, design tools, and opportunities," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 17–26, 2013.

[15] R. Bayles, "Aircraft Electric Power System Design, Control, and Protection," in *Electrified Aircraft Propulsion*, K. Haran, N. Madavan, and T. O'Connell, Eds. Cambridge University Press, 2022, pp. 29–48.

[16] S. G. Wirasingha and A. Emadi, "Classification and review of control strategies for plug-in hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 111–122, 2011.

[17] A. Biswas and A. Emadi, "Energy management systems for electrified powertrains: State-of-the-art review and future trends," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6453–6467, 2019.

[18] F. R. Salmasi, "Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5 I, pp. 2393–2404, 2007.

[19] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 20, pp. 82–102, 2013.

[20] R. Hiserote and F. Harmon, "Analysis of hybrid-electric propulsion system designs for small unmanned aircraft systems," *International Energy Conversion Engineering*

Conference, 2010.

[21] T. Marzougui, E. S. Saenz, and M. Bareille, “A rule-based energy management strategy for hybrid powered eVTOL,” *Innovation in Aviation & Space for opening New Horizons*, 2023.

[22] T. M. N. Bui, T. Q. Dinh, J. Marco, and C. Watts, “An Energy Management Strategy for DC Hybrid Electric Propulsion System of Marine Vessels,” *International Conference on Control, Decision, and Information Technologies*, 2018.

[23] K. Hein, Y. Xu, Y. Senthilkumar, W. Gary, and A. K. Gupta, “Rule-based operation task-aware energy management for ship power systems,” *IET Gener. Transm. Distrib.*, vol. 14, no. 25, pp. 6348–6358, 2020.

[24] J. P. S. P. Leite and M. Voskuijl, “Optimal energy management for hybrid electric aircraft,” *Aircr. Eng. Aerosp. Technol.*, vol. 92, no. 6, 2020.

[25] A. Sergent, M. Ramunno, M. D’Arpino, and M. Canova, “Optimal sizing and control of battery energy storage systems for hybrid turboelectric aircraft,” *SAE Int. J. Adv. Curr. Pract. Mobil.*, vol. 2, no. 3, 2020.

[26] F. D. Kanellos, G. J. Tsekouras, and N. D. Hatziargyriou, “Optimal demand-side management and power generation scheduling in an all-electric ship,” *Trans. Sustain. Energy*, vol. 5, no. 4, 2014.

[27] C. T. Aksland and A. G. Alleyne, “Hierarchical model-based predictive controller for a hybrid UAV powertrain,” *Control Eng. Pract.*, vol. 115, 2021.

[28] M. Doff-Sotta, M. Cannon, and M. Bacic, “Optimal energy management for hybrid electric aircraft,” *IFAC PapersOnLine*, 2020.

[29] S. Antonopoulos, K. Visser, M. Kalikatzarakis, and V. Reppa, “MPC Framework for the Energy Management of Hybrid Ships with an Energy Storage System,” *J. Mar. Sci. Eng.*,

vol. 9, no. 9, 2021.

[30] B. Liu, B. Xu, T. He, W. Yu, and F. Guo, “Hybrid Deep Reinforcement Learning Considering Discrete-Continuous Action Spaces for Real-Time Energy Management in More Electric Aircraft,” *Energies*, vol. 15, no. 17, 2022.

[31] P. Wu, J. Partridge, and R. Bucknall, “Cost-effective reinforcement learning energy management for plug-in hybrid fuel cell and battery ships,” *Appl. Energy*, vol. 275, 2020.

[32] D. Scott, S. G. Manyam, D. W. Casbeer, M. Kumar, M. J. Rothenberger, and I. E. Weintraub, “Power Management for Noise Aware Path Planning of Hybrid UAVs,” *American Control Conference*, 2022.

[33] S. G. Manyam, D. W. Casbeer, S. Darbha, I. E. Weintraub, and K. Kalyanam, “Path Planning and Energy Management of Hybrid Air Vehicles for Urban Air Mobility,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10176–10183, 2022.

[34] J. H. Jadischke, M. Wolff, J. Zumberge, B. Hencey, and A. Ngo, “Optimal Route Planning and Power Management for Hybrid UAV Using A* Algorithm,” *AIAA Aviation Forum*, 2023.

[35] A. Behbahani, A. Von Moll, R. Zeller, and J. Ordo, “Aircraft Integration Challenges and Opportunities for Distributed Intelligent Control, Power, Thermal Management, and Diagnostic and Prognostic Systems,” *Aerospace Systems and Technology Conference*, 2014.

[36] J. Doty *et al.*, “Dynamic Thermal Management for Aerospace Technology: Review and Outlook,” *J. Thermophys. Heat Transf.*, vol. 31, no. 1, pp. 86–98, Jan. 2017.

[37] 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.

[38] 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, pp. 1–16, 2018.

[39] T. O. Deppen, J. E. Hey, A. G. Alleyne, and T. S. Fisher, “A Model Predictive Framework for Thermal Management of Aircraft,” *Dynamic Systems and Control Conference*, 2015.

[40] D. T. Pollock, M. A. Williams, and B. M. Hencey, “Model predictive control of temperature-sensitive and transient loads in aircraft vapor compression systems,” *American Control Conference*, 2016.

[41] R. Scattolini, “Architectures for distributed and hierarchical Model Predictive Control - A review,” *J. Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[42] C. T. Aksland, P. J. Tannous, M. J. Wagenmaker, H. C. Pangborn, and A. G. Alleyne, “Hierarchical Predictive Control of an Unmanned Aerial Vehicle Integrated Power, Propulsion, and Thermal Management System,” *Trans. Control Syst. Technol.*, vol. 31, no. 3, pp. 1280–1295, May 2022.

[43] Z. Jin, L. Meng, J. M. Guerrero, and R. Han, “Hierarchical control design for a shipboard power system with DC distribution and energy storage aboard future more-electric ships,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 2, pp. 703–714, 2018.

[44] G. Seenumani, J. Sun, and H. Peng, “A hierarchical optimal control strategy for power management of hybrid power systems in all electric ships applications,” *IEEE Conference on Decision and Control*, 2010.

[45] H. Liu, A. Fan, Y. Li, R. Bucknall, and L. Chen, “Hierarchical distributed MPC method for hybrid energy management: A case study of ship with variable operating conditions,” *Renew. Sustain. Energy Rev.*, vol. 189, 2024.

[46] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[47] R. D. Smith, B. M. Hencey, A. C. Parry, and A. G. Alleyne, "Safe Learning of Energy Primitives for Electrified Aircraft," *American Control Conference*, 2024.

[48] "Simscape version: 4.7 (R2019b)." The MathWorks Inc., 2019.

[49] "Modelica Standard Library - Version: 4.0.0." Modelica Association Project - Libraries, 2020.

[50] K. Mahmud and G. E. Town, "A review of computer tools for modeling electric vehicle energy requirements and their impact on power distribution networks," *Appl. Energy*, vol. 172, pp. 337–359, 2016.

[51] P. T. McCarthy *et al.*, "A Multi-Domain Component Based Modeling Toolset for Dynamic Integrated Power and Thermal System Modeling," *SAE Tech. Pap.*, 2019.

[52] K. McCarthy *et al.*, "Dynamic Thermal Management System Modeling of a More Electric Aircraft," *Power Systems Conference*, 2008.

[53] M. Williams *et al.*, "PowerFlow: A Toolbox for Modeling and Simulation of Aircraft Systems," *SAE Tech. Pap.*, 2015.

[54] J. A. Tenreiro Machado and V. M. R. Cunha, *An Introduction to Bond Graph Modeling with Applications*. Boca Raton, FL: CRC Press, 2021.

[55] J. Jalving, Y. Cao, and V. M. Zavala, "Graph-based modeling and simulation of complex systems," *Comput. Chem. Eng.*, vol. 125, pp. 134–154, 2019.

[56] C. E. Laird and A. G. Alleyne, "A hybrid electro-thermal energy storage system for high ramp rate power applications," *Dynamic Systems and Control Conference*, 2019.

[57] C. Laird, Z. Kang, K. A. James, and A. G. Alleyne, "Framework for integrated plant and control optimization of electro-thermal systems: An energy storage system case study,"

Energy, vol. 258, Nov. 2022.

[58] D. J. Docimo and A. G. Alleyne, “Electro-Thermal Graph-Based Modeling for Hierarchical Control with Application to an Electric Vehicle,” *IEEE Conference on Control Technology and Applications*, 2018.

[59] 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.

[60] C. T. Aksland, “Modular Modeling And Control of A Hybrid Unmanned Aerial Vehicle’s Powertrain,” MS Thesis, University of Illinois at Urbana-Champaign, 2019.

[61] M. A. Williams, J. P. Koeln, H. C. Pangborn, and A. G. Allenye, “Dynamical Graph Models of Aircraft Electrical, Thermal, and Turbomachinery Components,” *J. Dyn. Syst. Meas. Control*, vol. 140, no. 4, pp. 1–17, 2018.

[62] C. Laird, “Modeling, control, and design of hybrid electrical and thermal energy storage systems,” MS Thesis, University of Illinois at Urbana-Champaign, 2021.

[63] J. P. Koeln, M. A. Williams, H. C. Pangborn, and A. G. Alleyne, “Experimental validation of graph-based modeling for thermal fluid power flow systems,” *Dynamic Systems and Control Conference*, 2016.

[64] P. Renkert, “Component-based design optimization of multirotor aircraft,” University of Illinois at Urbana-Champaign, 2022.

[65] C. T. Aksland, T. W. Bixel, L. C. Raymond, M. A. Rottmayer, and A. G. Alleyne, “Graph-based electro-mechanical modeling of a hybrid unmanned aerial vehicle for real-time applications,” *American Control Conference*, 2019.

[66] R. D. Smith and A. G. Alleyne, “Dynamical Graph-Based Models of Brayton Cycle

Systems," *American Control Conference*, 2022.

[67] 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.*, pp. 1–15, 2019.

[68] J. P. Koeln and A. G. Alleyne, "Robust hierarchical model predictive control of graph-based power flow systems," *Automatica*, vol. 96, pp. 127–133, 2018.

[69] J. P. Koeln, "Hierarchical Power Management in Vehicle Systems," PhD Dissertation, University of Illinois at Urbana-Champaign, 2016.

[70] P. J. Tannous, D. J. Docimo, H. C. Pangborn, and A. G. Alleyne, "Hierarchical estimation for complex multi-domain dynamical systems," *American Control Conference*, 2019.

[71] 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.

[72] O. B. Inal, J. F. Charpentier, and C. Deniz, "Hybrid power and propulsion systems for ships: Current status and future challenges," *Renew. Sustain. Energy Rev.*, vol. 156, no. December 2021, 2022.

[73] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[74] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numer. Math.*, vol. 1, pp. 269–271, 1959.

[75] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[76] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int.*

J. Rob. Res., vol. 30, no. 7, pp. 846–894, 2011.

[77] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, “Real-time motion planning with applications to autonomous urban driving,” *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, 2009.

[78] K. Yang, S. Keat Gan, and S. Sukkarieh, “A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV,” *Adv. Robot.*, vol. 27, no. 6, pp. 431–443, 2013.

[79] S. M. Lavalle and J. J. Kuffner, “Randomized Kinodynamic Planning,” *Int. J. Rob. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[80] P. Cheng and S. M. La Valle, “Reducing metric sensitivity in randomized trajectory design,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 1, pp. 43–48, 2001.

[81] C. Fan, K. Miller, and S. Mitra, “Fast and Guaranteed Safe Controller Synthesis for Nonlinear Vehicle Models,” *International Conference on Computer Aided Verification*, 2020.

[82] J. Bruce and M. M. Veloso, “Real-time randomized path planning for robot navigation,” *International Conference on Intelligent Robots and Systems*, 2002.

[83] K. Y. Tseng, J. S. Shamma, and G. E. Dullerud, “Low-Fidelity Gradient Updates for High-Fidelity Reprogrammable Iterative Learning Control,” *American Control Conference*, 2022.

[84] J. Seok, I. Kolmanovsky, and A. Girard, “Coordinated Model Predictive Control of Aircraft Gas Turbine Engine and Power System,” *J. Guid. Control. Dyn.*, vol. 40, no. 10, pp. 2538–2555, Oct. 2017.

[85] M. Salah, T. Mitchel, J. Wagner, and D. Dawson, “Adaptive and Robust Control for

Thermal Management Systems," 2006.

[86] H. C. Pangborn, "Hierarchical control for multi-domain coordination of vehicle energy systems with switched dynamics," PhD Dissertation, University of Illinois at Urbana-Champaign, 2019.

[87] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[88] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering*, vol. 4, no. 4. pp. 317–367, 1998.

[89] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject to bounded disturbances," University of Cambridge, 2004.

[90] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *Int. J. Control*, vol. 84, no. 6, pp. 1157–1170, 2011.

[91] M. Althoff, "An Introduction to CORA 2015," *1st and 2nd Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.

[92] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE J.*, vol. 49, no. 2, pp. 426–437, Feb. 2003.

[93] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *J. Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[94] MATLAB, *version 9.7.0.1737446 (R2019b)*. Natick, Massachusetts: The MathWorks Inc., 2019.

[95] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual.” 2022, [Online]. Available: https://www.gurobi.com.

[96] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, 2004.

[97] Iowa Environmental Mesonet, “15 December 2022 Observation History, [CMI] CHAMPAIGN/URBANA.” Iowa State University, 2023.

[98] F. Allgower *et al.*, “Nonlinear Predictive Control and Moving Horizon Estimation — An Introductory Overview,” *Advances in Control*, 1999.

[99] C. L. Butler, R. D. Smith, and A. G. Alleyne, “Sampling-based planning for guaranteed safe energy management of hybrid UAV powertrain under complex, uncertain constraints,” *(submitted to) Trans. Control Syst. Technol.*, 2024.

[100] D. Greenhalgh and S. Marshall, “Convergence criteria for genetic algorithms,” *SIAM J. Comput.*, vol. 30, no. 1, 2000.

[101] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” *Robot. Sci. Syst.*, vol. 6, pp. 267–274, 2011.

[102] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[103] A. T. Schwarm and M. Nikolaou, “Chance-Constrained Model Predictive Control,” *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, 1999.

[104] B. Zahedi and L. E. Norum, “Modeling and simulation of all-electric ships with low-voltage

DC hybrid power systems,” *IEEE Trans. Power Electron.*, vol. 28, no. 10, pp. 4525–4537, 2013.

APPENDICES

APPENDIX A: SHIPBOARD POWER SYSTEM CASE STUDY SUPPLEMENTARY MATERIAL

A.1: SPS simulation model parameters and validation

The parameters of the SPS model are shown in Table A.1. The reader is referred to [60] for a detailed description of the graph-based component models and parameters. The battery pack model is based on a pack of lithium-ion 18650 cells. The open circuit voltage curve, V_{ocv}, is plotted in Figure A.1 as a 7th-order polynomial function of the battery SOC. This is modified from the battery open circuit voltage curves obtained experimentally in [60].

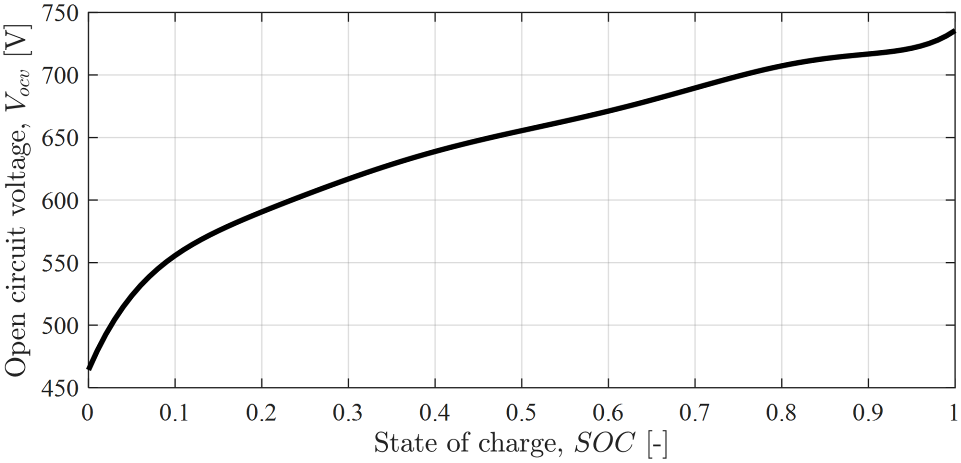


Figure A.1: SPS battery pack open circuit voltage curve.

Table A.1: SPS model parameters.

Component	*Parameter, symbol [unit]*	*Value*	*Component*	*Parameter, symbol [unit]*	*Value*
Battery pack	Capacity, Q [Ah]	1.07E3	Bus	Resistance, R [Ω]	5.00E-4
	Resistance 1, R_1 [Ω]	1.40E-2		Inductance, L [H]	1.00E-4
	Resistance 2, R_2 [Ω]	1.72E-2		Capacitance, C [F]	1.00E-1
	Capacitance 1, C_1 [F]	3.92E3	Switch	Resistance, R [Ω]	5.00E-4
	Capacitance 2, C_2 [F]	1.05E5		Inductance, L [H]	1.00E-6
	Internal resistance, R_s [Ω]	1.05E-2		Capacitance, C [F]	1.00E-3
Propulsion motor	Resistance, R [Ω]	2.92E-2	Inverter	Resistance, R [Ω]	1.47E-2
	Inductance, L [H]	4.87E-5	Vehicle	Fluid density, ρ [kg/m^3]	1.01E3
	Inertia, J [kg m^2]	2.73E-2		Prop diameter, D [m]	1.60
	Motor constant, K_v [Vs/rad]	5.00E-1		Drag coeff., C_D [-]	2.7E-2
	Viscous friction, b [Nms/rad]	3.60E-4		Frontal area, A [m^2]	5.50E2
	Static friction, c [Nms/rad]	2.28E-1		Mass, m [kg]	5.00E4
DC-DC Converter	Diode voltage, V_D [V]	3.97E-1		Thrust coeff., C_T [-]	3.30E-2
	Diode loss, R_D [Ω]	0.00		Torque coefficient, C_τ [-]	2.10E-2
	Switch loss, R_s [Ω]	2.42E-2		Prop. efficiency, η [-]	95.0%
	Coil loss, R_L [Ω]	2.39E-2	Hotel load	Inductance, L [H]	1.00E-3
Main genset	Max current, I_{max} [A]	2.80E3	Aux genset	Max current, I_{max} [A]	1.00E3
	Time constant, τ [s]	1		Time constant, τ [s]	5.0E-1

This parameter set was validated through comparison with simulation results for a similar SPS found in the literature [104]. These validation efforts used a model of the architecture shown in

Figure A.2, including two energy storage systems (ESS). Note that in this architecture used for model validation, both ESSs are battery packs, while the ESSs of the model in [104] included one battery pack and one fuel cell.

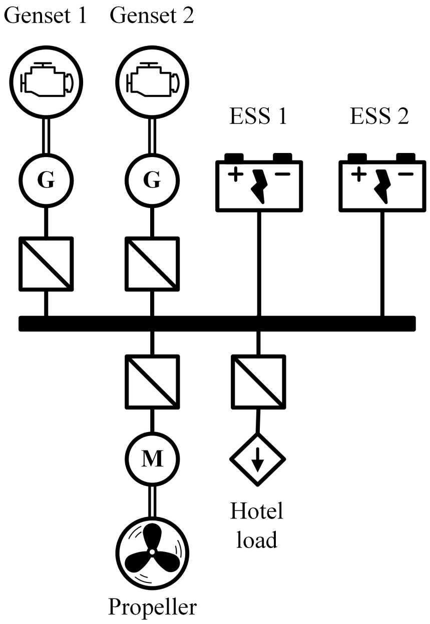


Figure A.2: SPS validation model architecture.

Feedback controllers were used to track references for the propulsion motor speed, genset currents, and hotel loads. For model validation, a mission with step changes in references for these components was considered. The simulation results, shown in Figure A.3, demonstrate good agreement with the validation data [104], with some discrepancies in the ESS current which can be attributed to the architectural differences between the model in Figure A.2 and the one presented in [104].

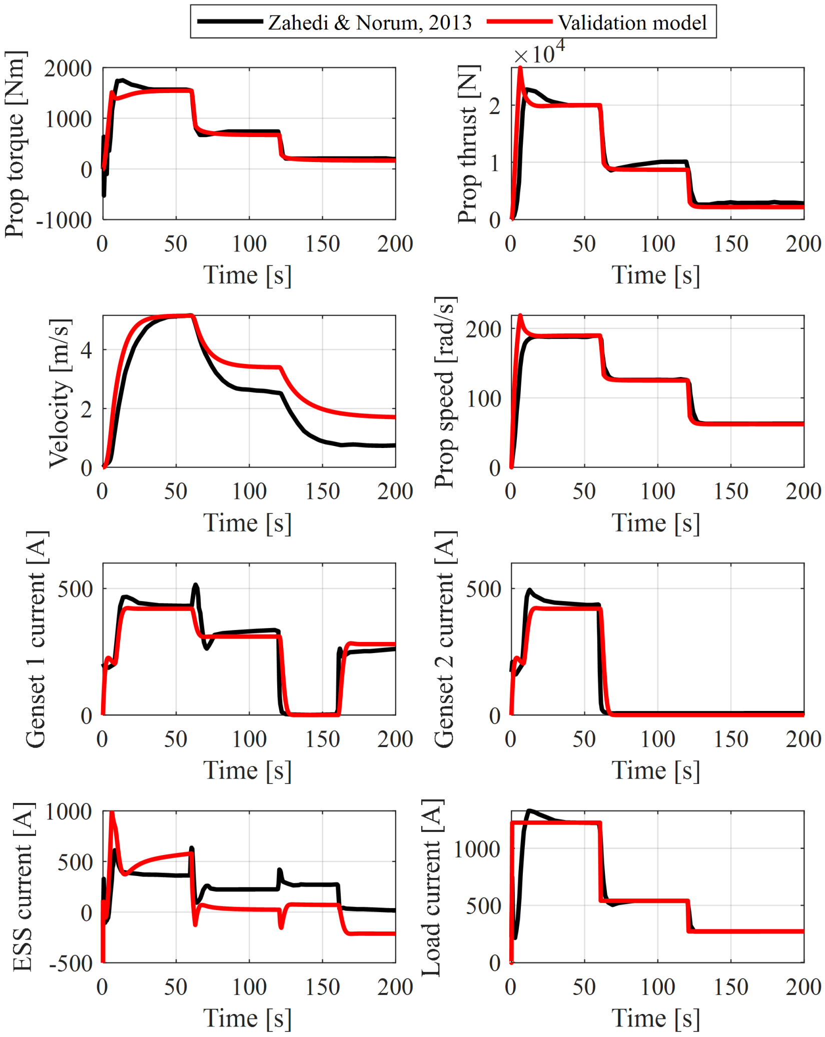


Figure A.3: SPS validation model simulation results and validation data from [104].

A.2: SPS RRT/ERRT parameters

Trim trajectories for the shipboard power system are provided in Table A.2. Symmetric operating points are those numbered 1-20, while asymmetric operating points are numbered 21-51. In each trim, an asterisk denotes the subsystem whose power state is 'free' (see Section 4.4.1).

Table A.2: Trim trajectories for shipboard power system.

	Subsystem operating point [kW]									
Trim number	*Main genset 1*	*Aux genset 1*	*Battery pack 1*	*Prop 1*	*Hotel 1*	*Main genset 2*	*Aux genset 2*	*Battery pack 2*	*Prop 2*	*Hotel 2*
1	0	0	-48.0*	25.2	10.0	0	0	-48.0*	25.2	10.0
2	-1550	0	-486*	683	10.0	-1550	0	-486*	683	10.0
3	0	0	-64.4*	25.2	25.0	0	0	-64.4*	25.2	25.0
4	-1160	0	-951*	683	25.0	-1160	0	-951*	683	25.0
5	0	0	-26.6*	0	25.0	0	0	-26.6*	0	25.0
6	-1550	0	-537*	683	50.0	-1550	0	-537*	683	50.0
7	-1160	-580	-323*	683	50	-1160	-580	-323*	683	50
8	0	0	-56.2*	0	50	0	0	-56.2*	0	50
9	-47.9*	0	0	25.2	10.0	-47.9*	0	0	25.2	10.0
10	-75.9*	0	28.0	25.2	10.0	-75.9*	0	28.0	25.2	10.0
11	-38.3*	0	28.0	0	10.0	-38.3*	0	28.0	0	10.0
12	-64.1*	0	0	25.2	25.0	-64.1*	0	0	25.2	25.0
13	-92.2*	0	28.0	25.2	25.0	-92.2*	0	28.0	25.2	25.0
14	-477*	0	28.0	202	25.0	-477*	0	28.0	202	25.0
15	-54.6*	0	28.0	0	25.0	-54.6*	0	28.0	0	25.0
16	-122*	0	28.0	25.2	50.0	-122*	0	28.0	25.2	50.0
17	-507*	0	28.0	202	50.0	-507*	0	28.0	202	50.0
18	-1470*	-580	0	683	50.0	-1470*	-580	0	683	50.0
19	-1490*	-580	28.0	683	50.0	-1490*	-580	28.0	683	50.0
20	-84.1*	0	28.0	0	50.0	-84.1*	0	28.0	0	50.0
21	0	-580	-145*	202	10.0	0	0	-145*	202	10.0
22	0	0	-145*	202	10.0	0	-580	-145	202	10.0
23	0	0	-192*	202	50.0	0	-580	-192*	202	50.0
24	-1160	0	-411*	683	10.0	0	0	-411*	25.2	10.0
25	-1160	0	-461*	683	50.0	0	0	-461*	25.2	50.0
26	-1160	0	-671*	683	25.0	0	0	-671*	202	25.0
27	-1160	0	-400*	683	25.0	0	0	-400*	0	25.0
28	0	0	-28.6*	0	10.0	0	0	-28.6*	25.2	10.0
29	0	0	-262*	0	50.0	0	0	-262*	202	50.0
30	0	0	-383*	0	10.0	-1160	0	-383*	683	10.0
31	0	0	-400*	0	25.0	-1160	0	-400*	683	25.0
32	-56.6*	0	28.0	0	10.0	-56.6*	0	28	25.2	10.0
33	-44.8*	0	0	0	25.0	-44.8*	0	0	25.2	25.0
34	-102*	0	28.0	0	50.0	-102*	0	28	25.2	50.0
35	-212*	0	0	0	10.0	-212*	0	0	202	10.0
36	-240*	0	28.0	0	10.0	-240*	0	28	202	10.0
37	0	-580	-192*	202	50.0	0	0	-192*	202	50.0
38	0	0	-411*	25.2	10.0	-1160	0	-411*	683	10.0
39	0	0	-653*	202	10.0	-1160	0	-653*	683	10.0
40	-1160	0	-37.4*	202	10.0	-1160	0	-37.4*	683	10.0
41	0	0	-432*	0	50.0	-1160	0	-432*	683	50.0
42	0	0	-28.6*	25.2	10.0	0	0	-28.6*	0	10.0
43	0	0	-44.9*	25.2	25.0	0	0	-44.9*	0	25.0
44	0	0	-74.7*	25.2	50.0	0	0	-74.7*	0	50.0
45	0	0	-215*	202	10.0	0	0	-215*	0	10.0
46	0	0	-232*	202	25.0	0	0	-232*	0	25.0
47	-1160	0	-432*	683	50.0	0	0	-432*	0	50.0
48	-28.5*	0	0	25.2	10.0	-28.5*	0	0	0	10.0
49	-56.6*	0	28.0	25.2	10.0	-56.6*	0	28.0	0	10.0
50	-212*	0	0	202	10.0	-212*	0	0	0	10.0
51	-240*	0	28.0	202	10.0	-240*	0	28.0	0	10.0

Tracking error bounds for the power states of the shipboard power system are given as follows for the trim trajectory corresponding to phase N_R of primitive i. Two distinct sets of error bounds are used depending on whether the gensets are on or off.

$$L^i_{N_R-1} = \begin{cases} [25,25,16,11,5.1,25,25,16,11,5.1]^T \text{ kW} & \text{if } \overline{P}^i_{N^i_R}(j)=0,\, j\in\{1,2,6,7\} \\ [51,51,13,11,5.1,51,51,13,11,5.1]^T \text{ kW} & \text{otherwise} \end{cases}$$

Likewise, two distinct sets of tracking error bounds during transitions of primitive i are given below, depending on whether the gensets are on or off.

$$L^i_k = \begin{cases} [25,25,2,12,5.2,25,25,2,12,5.2]^T \text{ kW} & \text{if } \overline{P}^i_k(j)=\overline{P}^i_{k-1}(j)=0,\, j\in\{1,2,6,7\} \\ [53,53,2,12,5.2,53,53,2,12,5.2]^T \text{ kW} & \text{otherwise} \end{cases}$$

A normal distribution was used to select random states. The parameters of the random state selection routine included the covariance $\mathbf{\Sigma}$ of this normal distribution and the weighting matrix Γ_X of the distance metric (4.6). These parameters were tuned using a genetic algorithm to minimize the time required to find a feasible solution. The tuned parameter set is given below.

$$\mathbf{\Sigma} = 10^{19} \times diag\big([0.379,0.637,1.79,1.48,0.635,0.459,1.75,0.375,0.955,0.338,\ldots \\ 0.347,0.296,0.483,1.35,1.55,0.847,0.229,0.643,1.58,1.06]\big)$$

$$\Gamma_X = diag\big([5.61,8.36,1.38,6.57,0.769,3.95,6.91,0.749,0.769,7.62,\ldots \\ 9.19,1.14,8.76,5.49,5.45,9.20,7.90,3.29,6.52,0.718]\big)$$

For the online re-planning case studies of Section 6.1.4, waypoint probability $p_w = 0.6$ was used.

A.3: SPS RMPC parameters

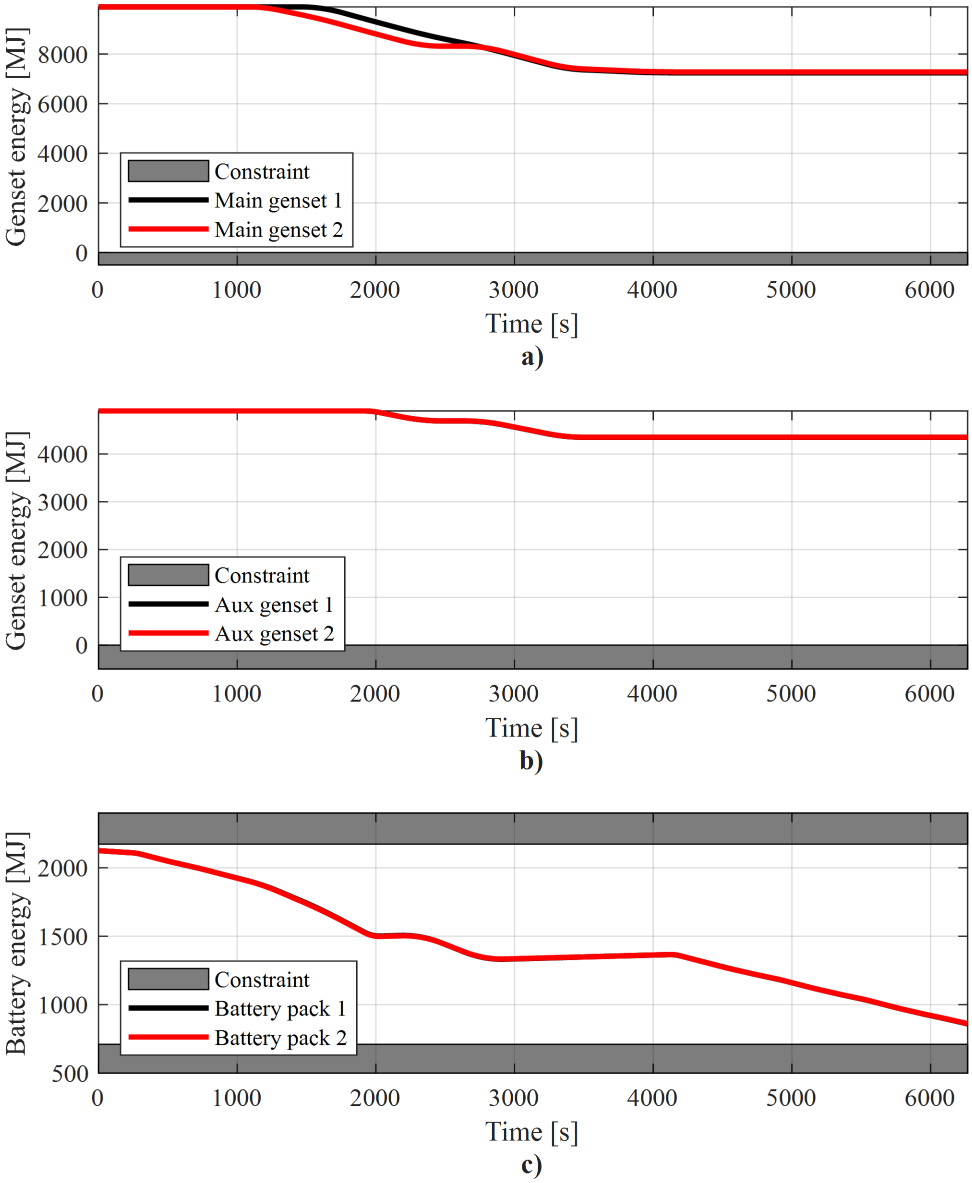
Parameters of RMPC include objective function weights, update rate, and prediction horizon length. Values of these parameters for the SPS case study are provided in Table A.3.

Table A.3: RMPC parameters for SPS case study.

Parameter	*Value*
Tracking error weight	$\Lambda_X = 10^{-5} \times diag([0^{1\times N}, 1, 1.6, 7.1, 3.8, 3.6, 1, 1.6, 7.1, 3.8, 3.6])$
Reduced order state slack weight	$\Lambda_{s^X} = 10^{-4} \times diag([0^{1\times N}, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3])$
Full order state slack weight	$\Lambda_s = diag(\lambda_s), \lambda_s(i) = \begin{cases} 10^{-4} & \text{if } i \in \{1, 7\} \\ 0 & \text{otherwise} \end{cases}$
State derivative weight	$\Lambda_{dx} = diag(\lambda_{dx}), \lambda_{dx}(i) = \begin{cases} 10^{-4} & \text{if } i \in \{4, 5, 6, 10\} \\ 0 & \text{otherwise} \end{cases}$
Input derivative weight	$\Lambda_{du} = diag([2, 1, 2, 1, 1, 1, 1, 1])$
Input deviation amount	$\delta_u = 10^{-1} \times [0.5, 0.5, 0.5, 0.5, 0.5, 2, 0.5, 2]$
Horizon length	$N_p = 10$
Update rate	$\Delta T = 0.5\text{s}$
Power state nonlinearity estimator gains	$K_{v1} = 0.248, K_{v2} = 0.148$

A.4: Additional simulation results for SPS case study

Figure A.4 shows that the gensets and battery packs are operated safely to assure satisfaction of the operational constraints $\mathcal{C}_{1-8}$ for the offline planning scenario of Section 6.1.3. Figure A.5 shows that this is also the case for online re-planning scenario of Section 6.1.4.

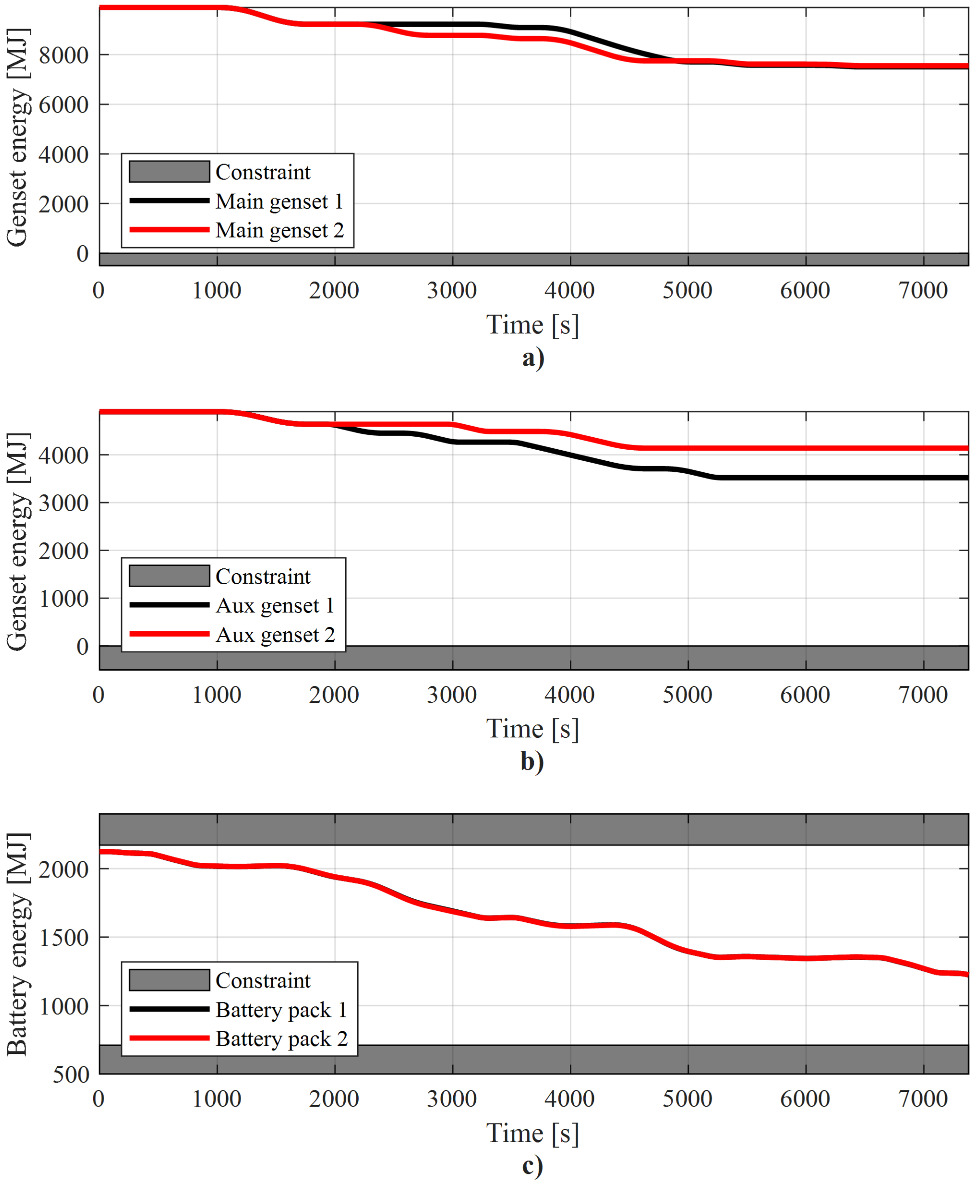


a)

b)

c)

Figure A.4: Operational constraints for SPS are satisfied during offline planning to assure safety. Subplot a) shows that main gensets satisfy constraint $\mathcal{C}_1$ *and* $\mathcal{C}_2$*, subplot b) shows that aux gensets satisfy constraints* $\mathcal{C}_3$ *and* $\mathcal{C}_4$*, and subplot c) shows that battery packs satisfy constraints* $\mathcal{C}_{5-8}$.



a)

b)

c)

Figure A.5: Operational constraints for SPS are satisfied during online re-planning to assure safety. Subplot a) shows that main gensets satisfy constraint $\mathcal{C}_1$ and $\mathcal{C}_2$, subplot b) shows that aux gensets satisfy constraints $\mathcal{C}_3$ and $\mathcal{C}_4$, and subplot c) shows that battery packs satisfy constraints $\mathcal{C}_{5-8}$.

APPENDIX B: HYBRID UAV PPTS CASE STUDY SUPPLEMENTARY MATERIAL

B.1: Hybrid UAV PPTS simulation model parameters

The powertrain of the hybrid UAV PPTS is based on that of [60], to which the reader is referred for detailed descriptions of the electro-mechanical component models. The thermal dynamics of this system include air bay temperatures. Air bays are modeled as thermal masses with constant heat capacity. Fans cool air bays according to the simplified graph-based model shown in Figure B.1.

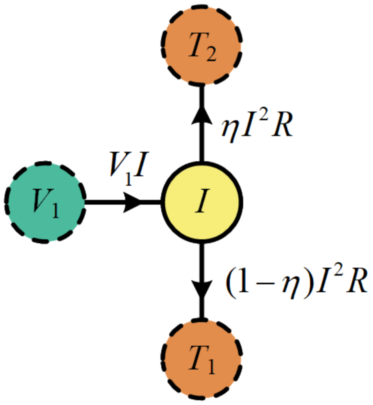


Figure B.1: Graph-based model of fans used in hybrid UAV PPTS model.

In this graph-based model, the fans are treated as constant-resistance loads. The amount of cooling provided by the fan to the air bay is given by $(1-\eta)I^2R$, where $\eta > 1$ is a constant parameter describing how efficiently the fan converts electrical input power to cooling. The vertex T_1 of Figure B.1 corresponds to the temperature of the air bay. Note that this simplified treatment of thermal dynamics may be subject to inaccuracies, but validation of this model is beyond the scope of this dissertation.

The electro-mechanical parameters of this model are largely unmodified from those listed in [60]. Thermal parameters of this model are listed in Table B.1 below.

Table B.1: Thermal parameters of hybrid UAV PPTS simulation model.

Component	*Parameter, symbol [unit]*	*Value*	*Component*	*Parameter, symbol [unit]*	*Value*
Battery pack	Heat capacity, C_p [J/K]	7.53E3	Rectifier	Heat capacity, C_p [J/K]	3.93E2
	Cooling resistance, R_u [K/W]	6.93E-2		Cooling resistance, R_u [K/W]	7.50E-1
DC-DC converters	Heat capacity, C_p [J/K]	3.93E2		Resistance, R [Ω]	9.20E-3
	Cooling resistance, R_u [K/W]	7.50E-1	Fan 1	Resistance, R [Ω]	1.00E1
Inverter	Heat capacity, C_p [J/K]	3.93E2		Cooling efficiency, η [-]	1.70E0
	Cooling resistance, R_u [K/W]	7.50E-1	Fan 2	Resistance, R [Ω]	2.00E1
	Resistance, R [Ω]	1.47E-2		Cooling efficiency, η [-]	1.70E0
Electronics cooling bay	Heat capacity, C_p [J/K]	2.00E0	Battery cooling bay	Heat capacity, C_p [J/K]	2.00E0

B.2: Hybrid UAV PPTS RRT parameters

Trim trajectories for the shipboard power system are provided in Table B.2. The battery bay operating point is determined as the following function of battery pack power.

$$\hat{P}_6(P_2) = 7.17\times10^{-9} P_2^2 + 2.41\times10^{-7} P_2 + 5.09\times10^{-5}$$

The coefficients of $\hat{P}_6$ were determined by applying a quadratic fit to simulation results for the steady state battery bay power and battery pack power with the battery bay fan completely off. These simulation results and the quadratic fit are shown in Figure B.2.

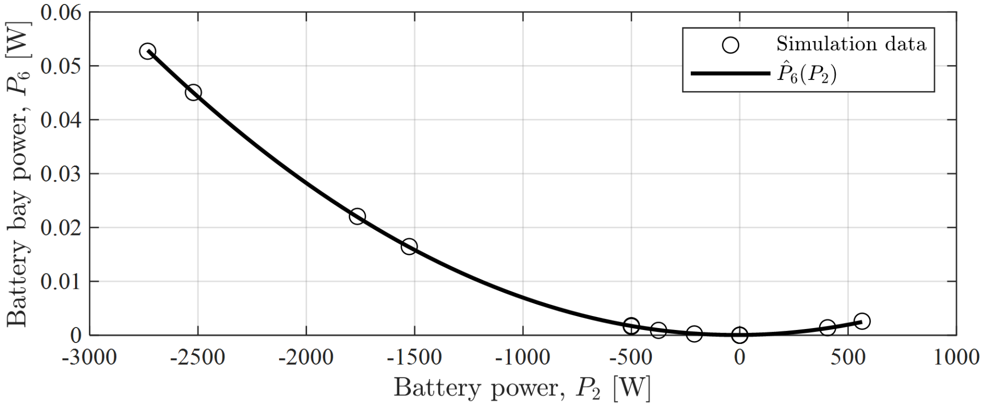


Figure B.2: Quadratic fit of hybrid UAV PPTS battery bay power as a function of battery pack power.

Table B.2: Trim trajectories for hybrid UAV PPTS.

	Subsystem operating point [W]					
Trim number	*Genset*	*Battery pack*	*Propulsion system*	*Avionic load*	*Power electronics bay*	*Battery bay*
1	0	-1570*	1200	175	0.02	0.015
2	-2480	535*	1200	175	0.02	0.0021
3	0	-2630*	2000	175	0.02	0.015
4	-2480	-232*	2000	175	0.02	0.00039
5	0	-1770*	1200	350	0.02	0.015
6	-2480	376*	1200	350	0.02	0.001
7	0	-2870*	2000	350	0.02	0.015
8	-2480	-404*	2000	350	0.02	0.0012
9	-1770*	0	1200	175	0.02	0
10	-1190*	-500	1200	175	0.02	0.0018
11	-2770*	0	2000	175	0.02	0
12	-2150*	-500	2000	175	0.02	0.0018
13	-1990*	0	1200	350	0.02	0
14	-1390*	-500	1200	350	0.02	0.0018
15	-2970*	0	2000	350	0.02	0
16	-2350*	-500	2000	350	0.02	0.0018
17	0	-1620*	1200	175	-0.01	-0.01
18	-2480	530*	1200	175	-0.01	-0.01
19	0	-2690*	2000	175	-0.01	-0.01
20	-2480	-262*	2000	175	-0.01	-0.01
21	0	-1830*	1200	350	-0.01	-0.01
22	-2480	349*	1200	350	-0.01	-0.01
23	0	-2930*	2000	350	-0.01	-0.01
24	-2480	-431*	2000	350	-0.01	-0.01
25	-1810*	0	1200	175	-0.01	-0.01
26	-1220*	-500	1200	175	-0.01	-0.01
27	-2800*	0	2000	175	-0.01	-0.01
28	-2180*	-500	2000	175	-0.01	-0.01
29	-2010*	0	1200	350	-0.01	-0.01
30	-1410*	-500	1200	350	-0.01	-0.01
31	-3010*	0	2000	350	-0.01	-0.01
32	-2380*	-500	2000	350	-0.01	-0.01
33	0	-1590*	1200	175	-0.01	0.015
34	-2480	518*	1200	175	-0.01	0.0019
35	0	-2660*	2000	175	-0.01	0.015

Table B.2 (Cont'd)

36	-2480	-262*	2000	175	-0.01	0.00049
37	0	-1800*	1200	350	-0.01	0.015
38	-2480	358*	1200	350	-0.01	0.00092
39	0	-2890*	2000	350	-0.01	0.015
40	-2480	-416*	2000	350	-0.01	0.0012
41	-1800*	0	1200	175	-0.01	0
42	-1210*	-500	1200	175	-0.01	0.0018
43	-2790*	0	2000	175	-0.01	0
44	-2170*	-500	2000	175	-0.01	0.0018
45	-201*	0	1200	350	-0.01	0
46	-1400*	-500	1200	350	-0.01	0.0018
47	-3000*	0	2000	350	-0.01	0
48	-2370*	-500	2000	350	-0.01	0.0018
49	0	-1600*	1200	175	0.02	-0.01
50	-2480	525*	1200	175	0.02	-0.01
51	0	-2670*	2000	175	0.02	-0.01
52	-2480	-246*	2000	175	0.02	-0.01
53	0	-1810*	1200	350	0.02	-0.01
54	-2480	367*	1200	350	0.02	-0.01
55	0	-2910*	2000	350	0.02	-0.01
56	-2480	-411*	2000	350	0.02	-0.01
57	-1790*	0	1200	175	0.02	-0.01
58	-1200*	-500	1200	175	0.02	-0.01
59	-2780*	0	2000	175	0.02	-0.01
60	-2160*	-500	2000	175	0.02	-0.01
61	-2000*	0	1200	350	0.02	-0.01
62	-1400*	-500	1200	350	0.02	-0.01
63	-2990*	0	2000	350	0.02	-0.01
64	-2360*	-500	2000	350	0.02	-0.01

Tracking error bounds for the power states of the hybrid UAV PPTS are given as follows for phase k of primitive i. Note that subsystems 1-4, corresponding to the electro-mechanical subsystems, are of type 1 so the tracking error bound applies to the power states of these subsystems. In contrast, subsystems 5-6, which correspond to the thermal dynamics of the air cooling bays, are of type 2 such that the tracking error bound applies to the energy states of these subsystems. In this case the same error bound is used for all trim trajectories and transitions.

$$L_k^i(j) = \begin{cases} 110\text{W} & \text{if } j \in \{1,2,3,4\} \\ 1.10\text{J} & \text{otherwise} \end{cases}$$

A normal distribution was used to select random states. The parameters of the random state selection routine included the covariance $\mathbf{\Sigma}$ of this normal distribution and the weighting matrix Γ_X of the distance metric (4.6). The parameter set used for this case study is given below.

$$\begin{aligned}\mathbf{\Sigma} = 10^{12} \times diag\big([&1.80,0.420,0.750,0.0600,0.0150,0.0150,...\\ &0.0010,0.0010,0.0150,0.0150,0.0150,0.0150]\big)\end{aligned}$$

$$\Gamma_X = diag\big([0.10, 20, 8.0, 20, 0.010, 0.010, 0, 0, 0, 0, 0, 0]\big)$$

B.3: Hybrid UAV PPTS RMPC parameters

Parameters of RMPC include objective function weights, update rate, and prediction horizon length. Values of these parameters for the hybrid UAV PPTS case study are provided in Table B.3.

Table B.3: RMPC parameters for hybrid UAV PPTS case study.

Parameter	*Value*
Tracking error weight	$\Lambda_X = diag(\lambda_X), \lambda_X(i) = \begin{cases} 1.5\times10^{-2} & \text{if } i \in \{5,6\} \\ 9\times10^{-4} & \text{if } i \in \{7,8,10\} \\ 1.5\times10^{-3} & \text{if } i = 9 \\ 0 & \text{otherwise} \end{cases}$
Reduced order state slack weight	$\Lambda_{s^X} = diag(\lambda_{s^X}), \lambda_{s^X}(i) = \begin{cases} 5\times10^{-3} & \text{if } i \in \{7,8,9,10\} \\ 1\times10^{-1} & \text{if } i = 5 \\ 1 & \text{if } i = 6 \\ 0 & \text{otherwise} \end{cases}$
State derivative weight	$\Lambda_{dx} = diag(\lambda_{dx}), \lambda_{dx}(i) = \begin{cases} 10^{-2} & \text{if } i \in \{6,17\} \\ 5\times10^{-1} & \text{if } i \in \{12,13\} \\ 10^{-1} & \text{if } i = 8 \\ 0 & \text{otherwise} \end{cases}$
Input derivative weight	$\Lambda_{du} = 0$
Input deviation amount	$\delta_u = 10^{-2} \times [5,5,5,5,5]$
Horizon length	$N_p = 10$
Update rate	$\Delta T = 0.5\text{s}$
Power state nonlinearity estimator gains	$K_{v1} = 0.345, K_{v2} = 0.221$

B.4: Additional simulation results for hybrid UAV PPTS case study

Satisfaction of operational constraints and the goal region in the hybrid UAV PPTS online re-planning case study is shown in Figure B.3, Figure B.4, and Figure B.5. An animation of online re-planning can be found at https://youtu.be/98X5xlNr4WU.

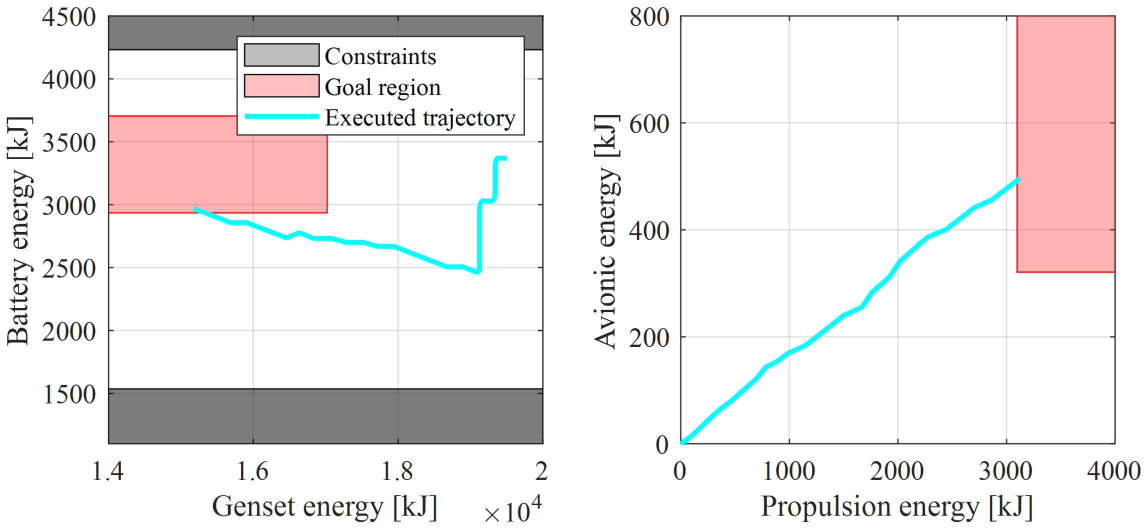


Figure B.3: Powertrain energy states, constraints, and goal region for hybrid UAV PPTS during online re-planning. Left subplot shows battery and genset energy states, including constraints $\mathcal{C}_{3-4}$ *which are shown to be satisfied and the goal region which is achieved. Right subplot shows avionic and propulsion energy states, where the propulsion energy state reaches the goal region at the end of the mission.*

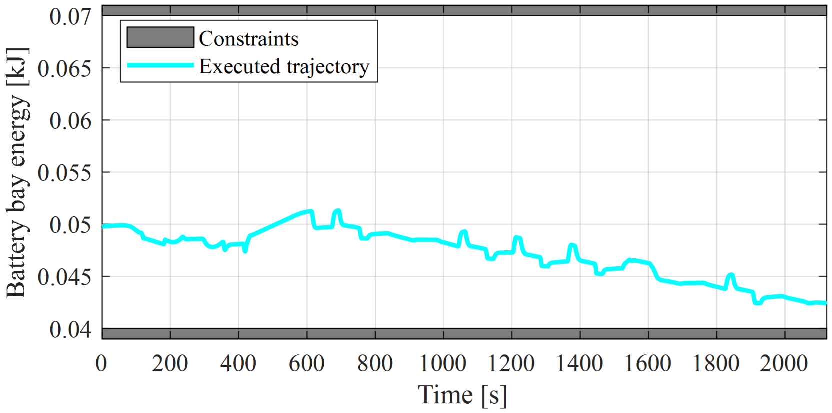


Figure B.4: During online re-planning, hybrid UAV PPTS battery bay energy state satisfies operational constraints $\mathcal{C}_{8,10}$.

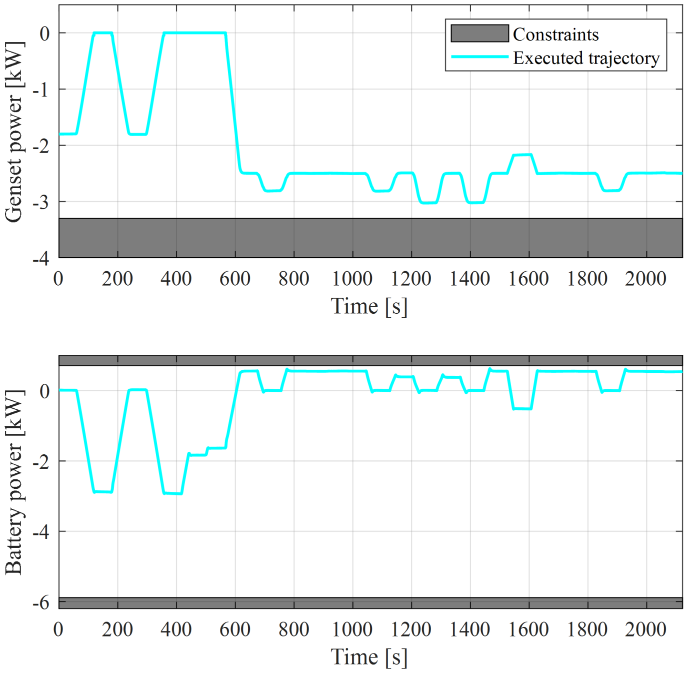


Figure B.5: Hybrid UAV PPTS genset and battery power states satisfy operational constraints during online re-planning. Top subplot shows that the genset power satisfies constraint $\mathcal{C}_2$*. Bottom subplot shows that the battery power satisfies constraints* $\mathcal{C}_{3,4}$.

APPENDIX C: HYBRID UAV POWERTRAIN CASE STUDY SUPPLEMENTARY MATERIAL

C.1: Moving horizon estimation for hybrid UAV powertrain testbed

The RMPC formulations used in this work rely on full state information for the graph-based model. However, only a select number of states of the hybrid UAV powertrain model are measurable in the hybrid UAV powertrain testbed as currently configured. To estimate unmeasured states while smoothing noise from experimental data, a moving horizon estimator [98] was used. The MHE formulation used past measurements to estimate the system, solving the following optimization problem at each time k using a horizon of length m :

$$
\begin{aligned}
&\min_{x,w}\left(\sum_{i=k-m}^{k-1}\left(\left\|w_i\right\|_{Q^{-1}}^2+\left\|v_{i+1}\right\|_{R^{-1}}^2\right)+\left\|x_{k-m}-\hat{x}_{k-m}^*\right\|_{\left(P_{k-m}^*\right)^{-1}}^2\right)\\
&\text{s.t.}\quad x_{i+1}=A_d x_i+A_t x_i^t+B_1 u_i+B_2+w_i,\ \ i=k-m,\dots,k-1\\
&\qquad\ \ x_i^a=E_d x_i+E_t x_i^t+F_1 u_i+F_2,\ \ i=k-m,\dots,k\\
&\qquad\ \ v_i=y_i-g(x_i,x_i^a),\ \ i=k-m+1,\dots,k\\
&\qquad\ \ [x_i;x_i^a]\subseteq \mathrm{X},\ \ i=k-m,\dots,k\\
&\qquad\ \ w_i\subseteq \mathrm{W},\ \ i=k-m,\dots,k-1
\end{aligned}
$$

The first term in the cost function penalizes dynamic state error, w_i. The second term penalizes error between the past m measurements y_i and the predicted outputs which are given by $g(x_i, x_i^a)$. The third term penalizes deviation of the first state estimate (m timesteps in the past) from the previous estimate $\hat{x}_{k-m}^*$. The first two constraints govern the linear state dynamics and algebraic state model. The third constraint defines measurement error. The final two constraints bound the states and state error values. The weighting matrices Q, R , and P_{k-m}^* represent, respectively, the model noise covariance, output noise covariance, and estimate covariance for step $k-m$.

The battery current, measured with a shunt resistor, is one of the experimental measurements that is subject to noise that can lead to noisy power flows in the graph model. To determine the performance of the observer in smoothing this noise, Figure C.1a) compares the battery current estimated from experimental measurements using the MHE and an extended Kalman filter (EKF). These estimates were analyzed in two different time ranges using the fast Fourier transform (FFT). Time range 1 corresponded to experimental data obtained with the engine off. Time range 2 corresponded to experimental data obtained with the engine on. Figure C.1b) shows the amplitude of the FFT signal at different frequencies during time range 1. Figure C.1b) shows that MHE and EKF yield similar noise reduction at low frequencies <1Hz, but MHE provides more smoothing at higher frequencies. Figure C.1c) shows the amplitude of the FFT signal at different frequencies during time range 2. Figure C.1c) shows that, in time range 2, the experimental data exhibits increased noise, but MHE provides improved smoothing of this noise compared to EKF. Quantitatively, Table C.1 shows that the power spectral density of the FFT signal corresponding to the MHE is lower in both time ranges, demonstrating improved noise reduction compared to the EKF. Figure C.2 shows estimation results for both observers for the full set of measurements in Table 6.11, demonstrating smoothing of noisy experimental data.

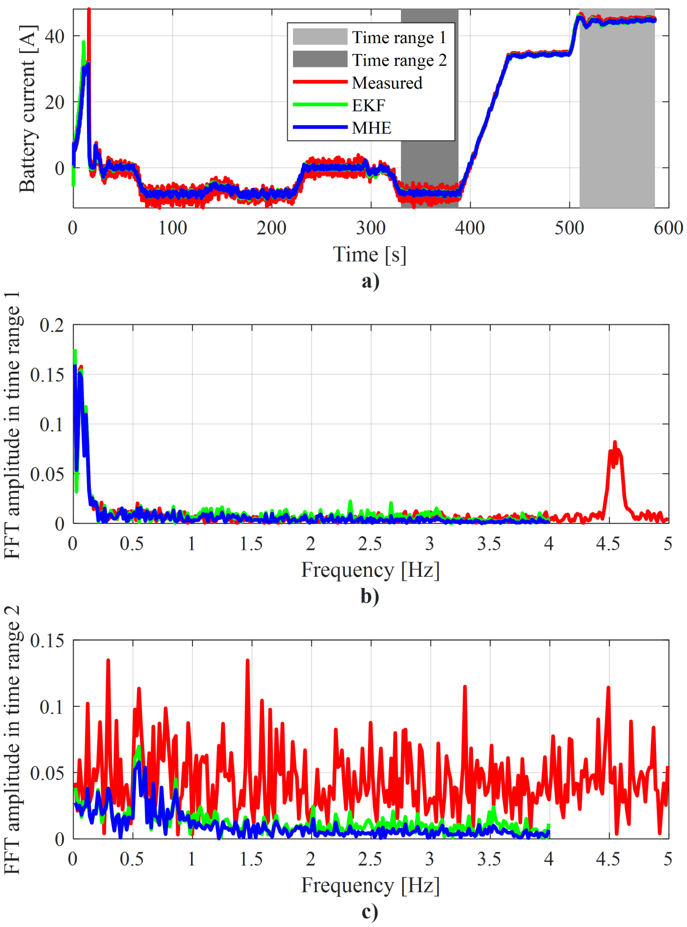


Figure C.1: Comparison of MHE and EKF for smoothing noise in battery current measurements from hybrid UAV powertrain testbed.

Table C.1: Comparison of power spectral density of battery current estimate using MHE and EKF.

	Measured	*EKF*	*MHE*
Power spectral density in time range 1	0.105	0.125	0.0601
Power spectral density in time range 2	5.52	0.430	0.302

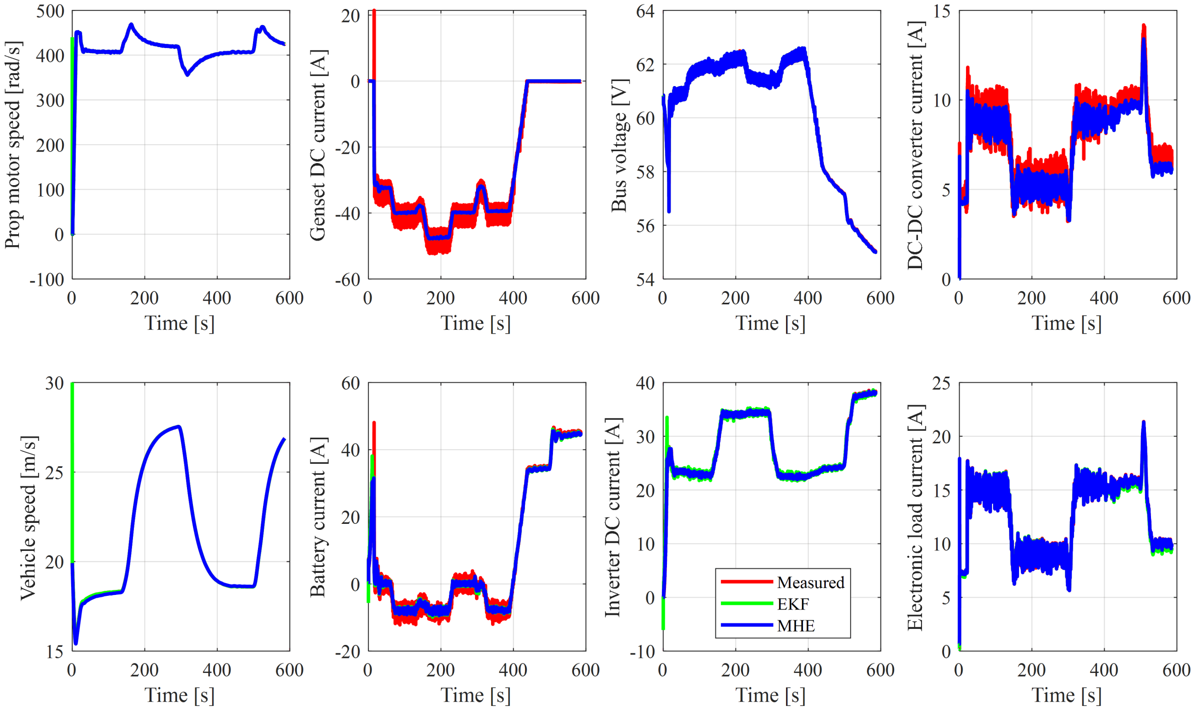


Figure C.2: Example results of moving horizon estimation for hybrid UAV powertrain testbed measurements.

C.2: Hybrid UAV powertrain RRT/ERRT parameters

The trim trajectories used for both offline and online planning case studies were provided in Table 6.13 of Section 6.3.2. Different sets of transitions were used to construct energy primitives in the offline planning scenarios than in the online planning scenarios. For the offline planning scenarios, transitions between trim trajectories consisted of multiple phases following pre-specified paths. The lines plotted in Figure C.3 illustrate the pre-specified paths used to transition between trim trajectories (i.e., operating points). These paths consisted of two classes of transitions: one which changed the power split between the sources and/or sinks while maintaining the same overall power level, and one which changed the overall power level of the system while maintaining the power split between the sources and sinks. The former class of transitions are shown in the figure as lines having the same hue (e.g., the transition from 1-3), and the latter class as lines having a gradient hue between blue and red (e.g., 2-5).

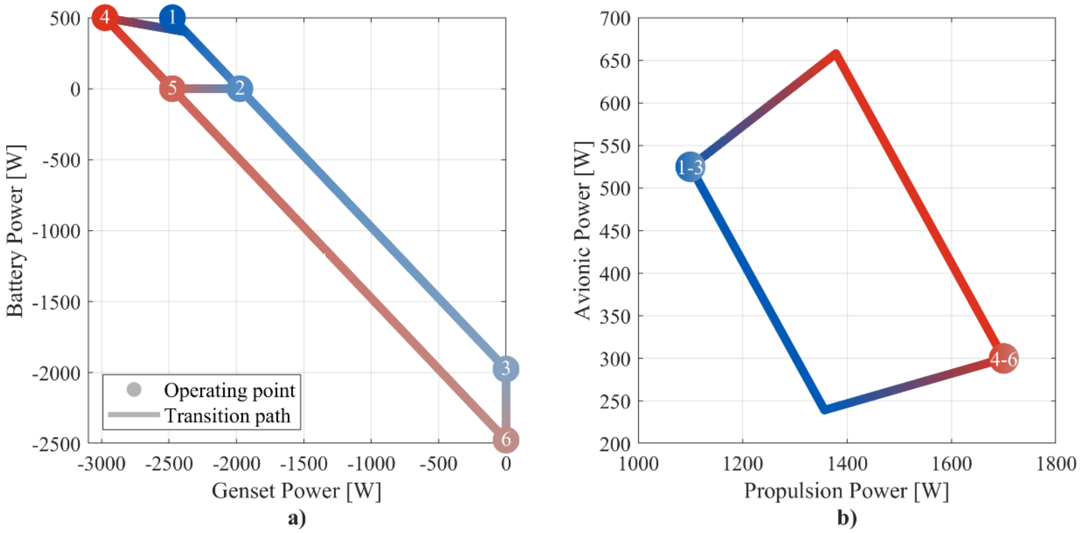


Figure C.3: Energy primitive transitions for hybrid UAV powertrain offline planning scenarios. Subplot a) shows the transitions for genset and battery power states and subplot b) shows the transitions for propulsion and avionic power states.

For the online re-planning scenario, energy primitives were constructed such that each transition consisted of a single phase with a constant ramp rate. This is done using (4.37), (4.38) with $R^{max} = 50\text{W/s}$.

Tracking error bounds for the power states of the hybrid UAV powertrain are given as follows for the trim trajectory corresponding to phase N_R of primitive i. Two distinct sets of error bounds are used depending on whether the genset is on or off.

$$L^i_{N_R-1} = \begin{cases} [50,200,125,125]^T\text{ W} & \text{if } \overline{P}^i_{N^i_R}(1) = 0 \\ [100,100,100,100]^T\text{ W} & \text{otherwise} \end{cases}$$

Likewise, two distinct sets of tracking error bounds during transitions of primitive i are given below, depending on whether the genset is on or off.

$$L^i_k = \begin{cases} [50,200,125,125]^T\text{ W} & \text{if } \overline{P}^i_{N^i_R}(1) = 0 \\ [125,125,125,125]^T\text{ W} & \text{otherwise} \end{cases}$$

A normal distribution was used to select random states. The parameters of the random state selection routine included the covariance $\mathbf{\Sigma}$ of this normal distribution and the weighting matrix Γ_X of the distance metric (4.6). The parameter set used for this case study is given below.

$$\mathbf{\Sigma} = 10^{12} \times diag([3, 0.6, 3, 0.24, 0.0003, 0.0003, 0.025, 0.025])$$

$$\Gamma_X = diag\left([0.5, 20, 10, 20, 0, 0, 0, 0]\right)$$

For the ERRT algorithm used in the online re-planning case study, the waypoint probability p_w was 60% and the fuel consumption weighting parameter was $\beta = 10^5$. Parameter sweeps were performed to inform the selection of these parameters. Figure C.4 and Figure C.5 below show the effect of these parameters on fuel consumption and planning time. Each pixel in these figures represents the average of 100 mission planning scenarios using the corresponding parameter values. Figure C.4 shows a coarse sweep of fuel consumption (FC) weighting values $\beta \in \{10^1, 10^2, 10^3, 10^4, 10^5\}$, demonstrating that small β values have little effect on the algorithm performance. However, at $\beta = 10^5$, Figure C.4 shows a clear increase in planning time and reduction in fuel consumption. Larger waypoint probability values reduce planning time. Figure C.5 shows a fine sweep of fuel consumption weighting values in a neighborhood of $\beta = 10^5$. Figure C.5 shows that increasing β reduces the fuel consumption, but also leads to a steep increase in planning time. Because higher waypoint probabilities generally exhibit lower planning times, this increase in planning time can be mitigated to some extent by increasing waypoint probability.

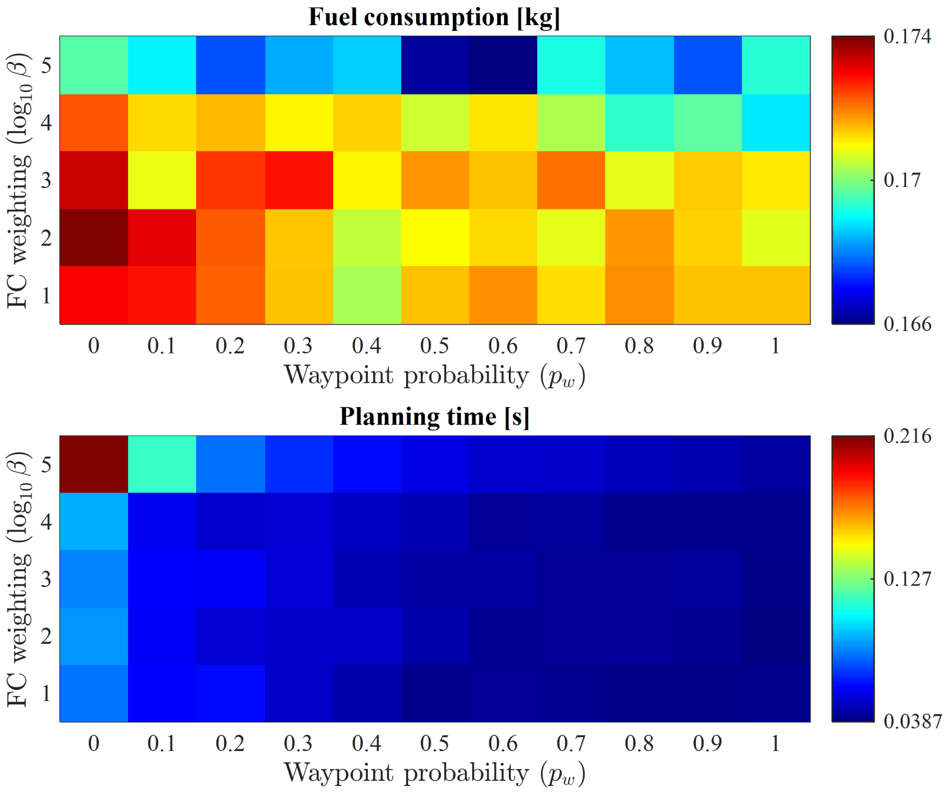


Figure C.4: ERRT parameter sweep: coarse discretization of fuel consumption weighting.

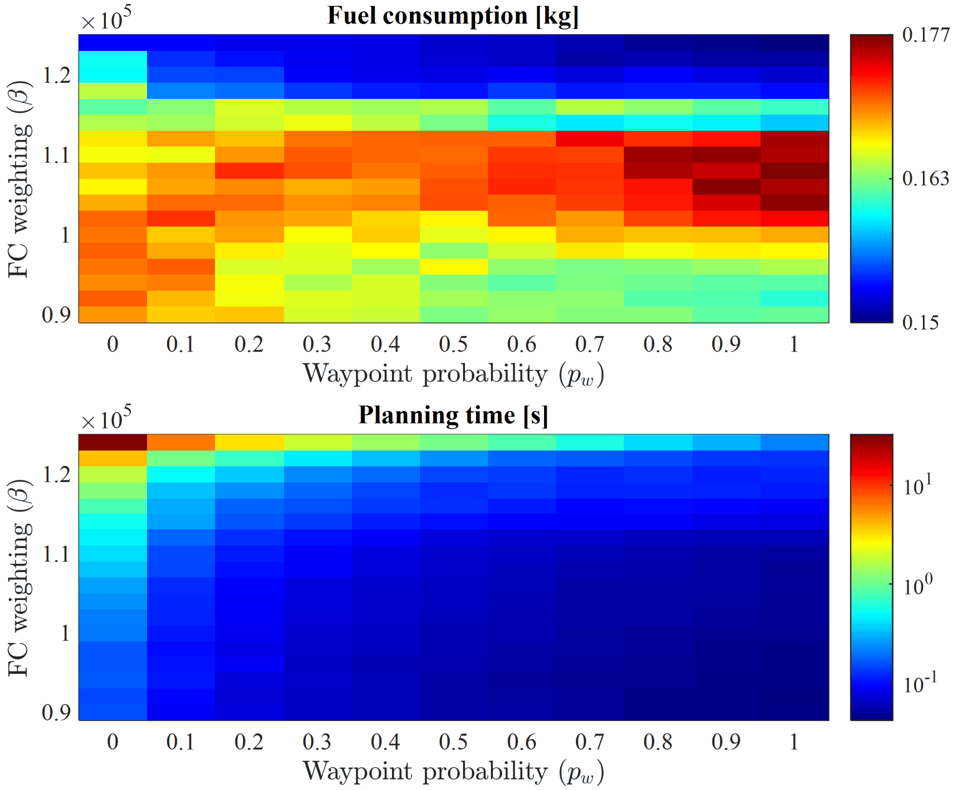


Figure C.5: ERRT parameter sweep: fine discretization of fuel consumption weighting.

C.3: Hybrid UAV powertrain RMPC parameters

Parameters of RMPC include objective function weights, update rate, and prediction horizon length. Values of these parameters for the hybrid UAV powertrain case study are provided in Table C.2.

Table C.2: RMPC parameters for hybrid UAV PPTS case study.

Parameter	*Value*
Tracking error weight	$\Lambda_X = diag(\lambda_X), \lambda_X(i) = \begin{cases} 0.1 & \text{if } i \in \{5,8\} \\ 0.3 & \text{if } i = 6 \\ 0.2 & \text{if } i = 7 \\ 0 & \text{otherwise} \end{cases}$
Reduced order state slack weight	$\Lambda_{s^X} = diag(\lambda_{s^X}), \lambda_{s^X}(i) = \begin{cases} 10 & \text{if } i = 5 \\ 5000 & \text{if } i = 6 \\ 100 & \text{if } i \in \{7,8\} \\ 0 & \text{otherwise} \end{cases}$
State derivative weight	$\Lambda_{dx} = diag(\lambda_{dx}), \lambda_{dx}(i) = \begin{cases} 1.7 \times 10^{-4} & \text{if } i = 5 \\ 8 & \text{if } i = 6 \\ 0 & \text{otherwise} \end{cases}$
Input derivative weight	$\Lambda_{du} = diag([0,0,200])$
Input deviation amount	$\delta_u = [0.1, 0.1, 0.2]$
Horizon length	$N_p = 10$
Update rate	$\Delta T = 0.5\text{s}$
Power state nonlinearity estimator gains	$K_{v1} = 0.233, K_{v2} = 0.632$

C.4: Additional simulation and experimental results for hybrid UAV powertrain case study

Figure C.6 shows that operational constraints for the battery and genset are satisfied in both simulation and experiment for the online re-planning scenario. An animation of online re-planning, including a demonstration of the system entering and exiting the quiet zone, can be found at https://youtu.be/sB7WWQA-NSI.

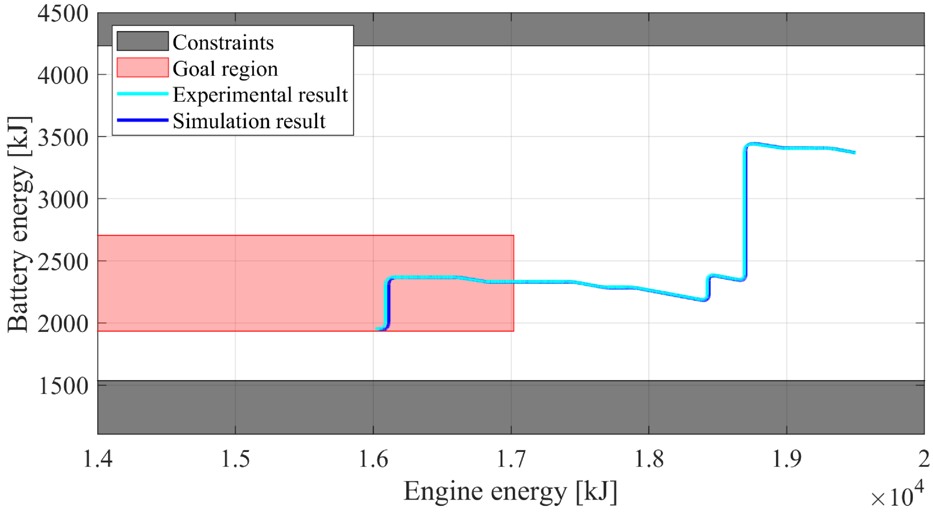


Figure C.6: Operational constraints are satisfied in hybrid UAV powertrain online re-planning case study. Final battery energy lands within the goal region.

Figure C.7 below shows that the computation times of RMPC, including set propagation (i.e., calculation of linearization error reachable sets) and optimization, remain well below the 0.5s sample time.

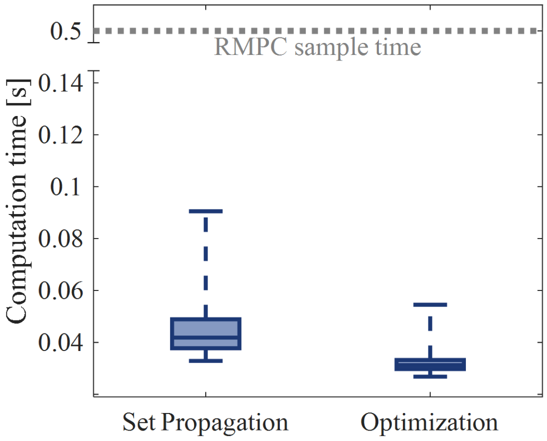


Figure C.7: Computation times of RMPC.

Figure C.8 below compares the reduced order model to several physically meaningful quantities. These results show that the battery energy, genset power, propulsion energy, and avionic power states of the reduced order model (all plotted in red) provide good estimates of the battery SOC, genset current, vehicle distance, and avionic load current, respectively. While there is clearly some slight error in the reduced order model, a key feature of this planning framework is the ability to artificially bloat constraints to account for these types of errors to ensure safety.

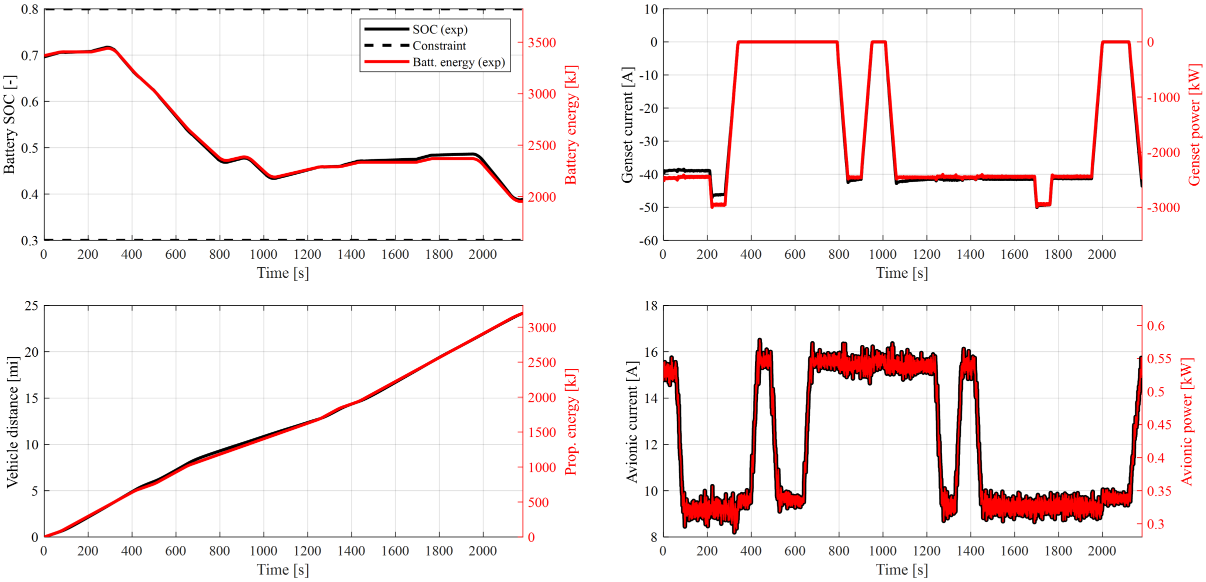


Figure C.8: Comparison of reduced-order model and physically meaningful quantities.

The prediction model used in RMPC is linearized and discretized from the nonlinear graph-based model. This model was derived, parameterized, and experimentally validated in previous work [27], [60], demonstrating good agreement with the experimental system. To show the accuracy of this prediction model, the linearized and nonlinearized models are compared. In particular, the simulation results in Figure C.9 show the nonlinear and linearized models under the same input profile shown in Figure C.10. The discretization step of the linearized model is 0.5s (the same as RMPC). The linearized model was re-linearized only once every 10s about a linearization point consisting of the linearized states and inputs delayed by one time step (0.5s). A 500s mission was simulated to capture a wide range of operation. While the simulation results in Figure C.9 show that there is some drift between the linearized and nonlinear model, it is clear that the linearized model captures the behavior of the true system. The drift is especially apparent in the avionic load current state, I_{load}, due to initial transients, but this drift remains small throughout the simulation. In this work, RMPC uses a prediction model that is re-linearized every time step about a linearization point consisting of the estimated states of the nonlinear model.

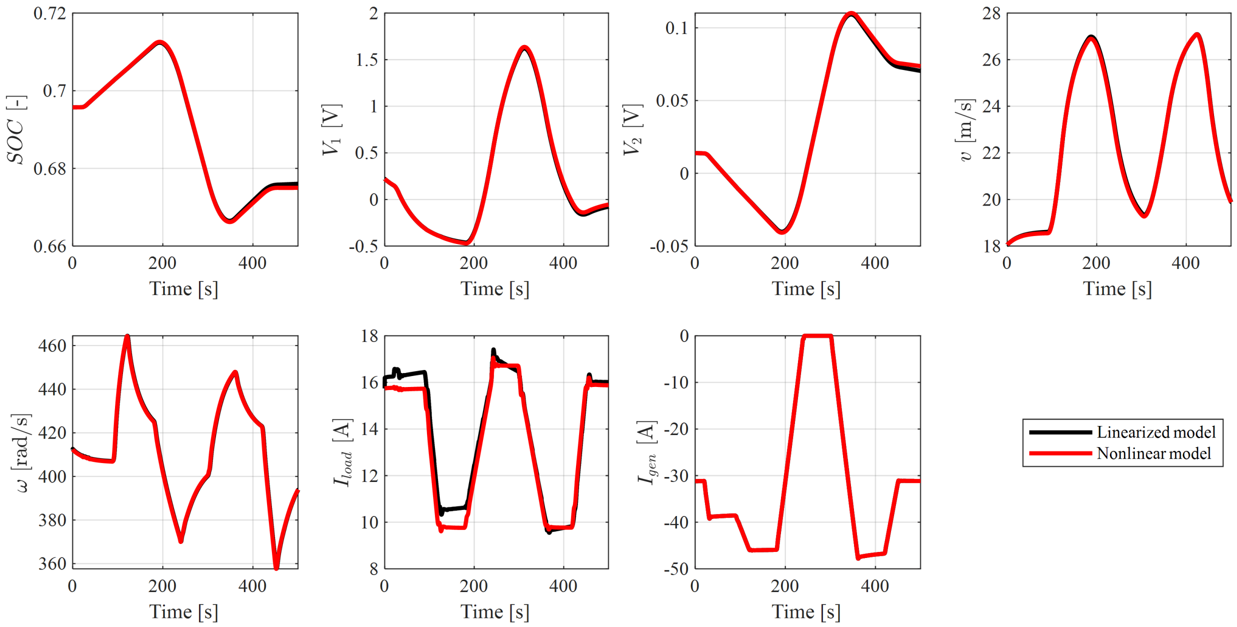


Figure C.9: Linear model validation of hybrid UAV powertrain states.

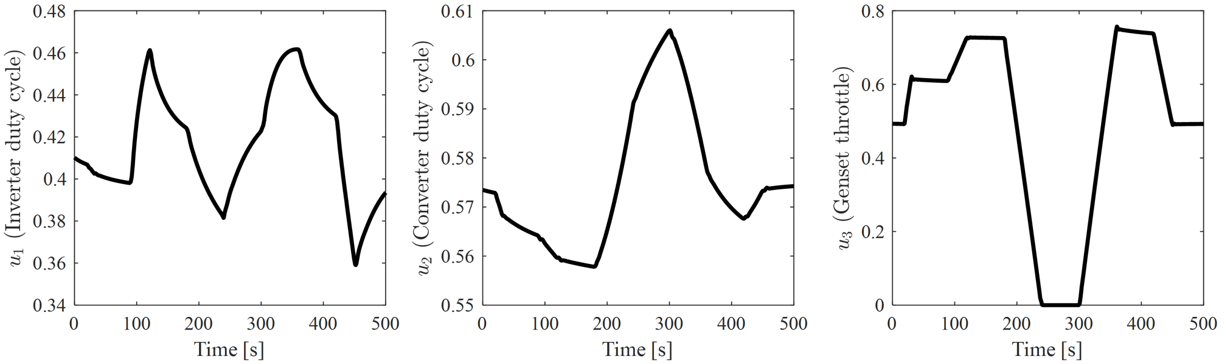


Figure C.10: Inputs for linear model validation of hybrid UAV powertrain.