A Convex Approach to Trajectory Optimization for Advanced Air Mobility

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Yufei Wu

Aug 2024

Acknowledgments

I am profoundly grateful to my advisor, Dr. Zhenbo Wang, for his invaluable insights and guidance throughout my research journey. His thoughtful comments have significantly shaped my publications, and his encouragement was crucial during times when I felt uncertain about my direction.

I also extend my deepest thanks to my lab mates, Yang Shi and Sabrullah Denis. Our conversations have sparked many of my research ideas, and their support during challenging times away from my family has been truly comforting.

I extend my sincere gratitude to my committee members, Dr. Daning Huang, Dr. Hans DeSmidt, and Dr. Xiaopeng Zhao, for their guidance and expert advice. I'm really thankful for their helpful feedback and support, which were crucial in making this dissertation better and getting it done successfully.

I must express my heartfelt appreciation to my parents and family, whose endless support and encouragement have fueled my pursuit of academic and research excellence. I must express my heartfelt appreciation to my parents and family, whose endless support and encouragement have kept me going. Their belief in me has been my main source of strength and motivation throughout this process.

Lastly, I am thankful to all the individuals who have contributed to my research, whether through direct collaboration or by offering support and encouragement. Working with such talented and dedicated individuals has been an extraordinary privilege and has enriched my professional experience tremendously.

Abstract

This dissertation addresses the challenge of real-time trajectory optimization for electric Vertical Take-Off and Landing (eVTOL) vehicles in the context of Advanced Air Mobility (AAM). With the airspace becoming increasingly crowded, ensuring the safety, efficiency, and feasibility of eVTOL operations is crucial. This research primarily focuses on the development and application of convex optimization techniques to solve trajectory optimization problems under different AAM concept of operations.

The study is structured into several critical analyses and methodological developments across three main research tasks. In the first task (Chapter 3), a multi-phase trajectory optimization problem is introduced that transitions from cruise to landing phases. A Sequential Convex Programming (SCP) approach is developed and demonstrated to show how various phases of AAM flight can be seamlessly integrated and optimized using advanced mathematical optimization techniques.

The second task (Chapter 4) dives into high-fidelity trajectory optimization that incorporates aerodynamic effects into nominal flight dynamics, significantly enhancing the model capability of simulating complex flying environment. To solve the problem, the SCP method is employed due to its effectiveness in handling nonconvex problems by decomposing the problem into a series of convex subproblems that are computationally feasible to solve in real-time.

Additionally, in Chapter 5, this dissertation explores decentralized optimal control and convex optimization approaches for the coordination of multiple eVTOLs at merging airways, a critical scenario as AAM systems evolve. This includes addressing the challenges of real-time vehicle coordination and collision avoidance. The results of the developed convex approaches have been compared with those from GPOPS through extensive simulations.

This dissertation contributes significantly to the field of AAM by demonstrating the feasibility of using advanced convex optimization techniques for real-time trajectory optimization in complex AAM settings. The methodologies developed herein have the potential to revolutionize the safety and efficiency of future air transport systems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Transportation is a critical component of modern society, playing a vital role in the daily lives of individuals and the functioning of economies. The concept of on-demand mobility (ODM) has been developed to enhance the efficiency of transportation systems, utilizing fast communication networks to connect passengers with transportation providers in real time [75]. Despite these advancements, ground transportation often suffers from significant drawbacks such as frequent delays due to traffic congestion, especially during peak hours, and limitations imposed by road availability and conditions. In response to these challenges, researchers have proposed extending the ODM concept to the realm of air transportation. This extension aims to offer services that allow passengers to decide the origin, destination, and timing of their trips, with the potential to significantly reduce travel time compared to ground transportation [75].

To realize this vision of on-demand air mobility, the concept of urban air mobility (UAM) has been introduced. UAM aims to provide flexible and efficient air mobility within urban and suburban areas for both passenger and cargo transport [54]. Recent technological advancements in areas such as battery technology, electric propulsion, and autonomous systems have facilitated the development of a new class of aerial vehicles known as electric vertical takeoff and landing (eVTOL) vehicles. These vehicles are designed to meet the unique demands of UAM missions and are expected to be operated primarily through autonomous systems, enhancing both the efficiency and safety of operations compared to traditional helicopters [26].

In the context of autonomous eVTOL operations, trajectory optimization is a critical component that ensures safe and efficient navigation through complex urban landscapes. However, despite its importance, there is currently no established framework for eVTOL trajectory planning that adequately addresses the diverse operational scenarios these vehicles may encounter. This study aims to fill this gap by developing innovative optimal control methodologies designed for eVTOL trajectory optimization under various operational conditions.

1.1 Motivation

The latest advancement in communication, computation, and sensing technologies has been driving a revolution in transportation systems. Urban air mobility, in particular, is currently gaining increasing attention from government agencies, industries, and research institutes as a promising solution to urban congestion and mobility challenges. Since eVTOL vehicles usually operate at low altitudes, it is crucial to have reliable control and optimization tools to ensure that the vehicles can safely and accurately complete flight missions under complex constraints and strict regulations[69].

Unlike autonomous ground vehicles, where optimal control is not always critical due to the powerful of closed-loop control systems in managing trajectory deviations, the scenario is different for aerial vehicles. In autonomous ground vehicles, deviations from a planned trajectory can typically be corrected effectively through real-time responses to environmental inputs, making optimization of initial trajectories less crucial. Furthermore, while energy efficiency is beneficial for electric ground vehicle, it is not usually a top concern, as these vehicles can generally complete their missions even with suboptimal energy usage, provided that sufficient charging infrastructure is available. In contrast, for aerial vehicles, particularly eVTOL aircraft, energy consumption is not only a matter of efficiency but also of critical safety and operational viability. Optimizing energy usage is essential due to the limited energy capacity of batteries and the high power requirements for takeoff, cruising, and landing. Inefficient energy consumption can jeopardize the entire mission, potentially leading to scenarios where the vehicle cannot complete its intended flight path or meet safety margins during critical phases of flight. Therefore, in the realm of eVTOL operations, trajectory planning and optimal control is significant to the success and safety of each mission. This highlights the unique challenges associated with trajectory optimization in urban air mobility, underscoring the need for sophisticated, energy efficient control strategies.

Trajectory optimization, a critical field in aerospace engineering, involves calculating optimal paths by considering the dynamics of the vehicle system while optimizing specific performance indices under predefined constraints [62]. Since the 1950s, researchers have employed various numerical methods to solve optimal control problems, which can broadly categorized into direct and indirect methods [76].In recent years, thanks to powerful computational tools, direct methods have been widely used in aerospace research. On the other hand, with the fast-growing research in the field of artificial intelligence has introduced innovative algorithms, such as reinforcement learning, for pre-flight path planning [29, 93]. However, these AI-driven methods often struggle with constraint integration and real-time control output generation, and traditional numerical approaches may not offer the real-time solutions required for practical applications. Therefore, there is a need to develop a reliable, real-time trajectory optimization framework for eVTOLs that ensures safe and efficient operations.

In this research, we aim to leverage the strengths of convex optimization, a mathematical technique that focuses on minimizing convex functions over convex sets, known for its fast convergence rates and assurance of global minima [12]. By formulating trajectory optimization problems as convex problems, this study seeks to explore and exploit the inherent advantages of convex optimization to address the complex challenges posed by various UAM mission scenarios. This approach is expected to yield a robust, efficient, and scalable solution for real-time eVTOL trajectory optimization.

1.2 Research Objectives

The overall goal of this research is to enable fast yet accurate trajectory optimization for eVTOL vehicles under different operating environments. Towards achieving this goal, three specific research aims will be performed.

The first part of the study focuses on the eVTOL multi-phase trajectory optimization problem of cruising, descending, and landing. In the second part, a high-fidelity dynamic model is introduced to the system, and a high-fidelity eVTOL landing problem is solved. For the third part, the convex optimization approach is extended to solve eVTOL merging problems.

Given the limitations of current technologies, particularly in terms of battery capacities, optimizing energy consumption during flight is a primary concern. Therefore, each component of this research is designed with the objective of minimizing energy costs, thereby enhancing the sustainability and feasibility of eVTOL technologies.

1.2.1 Aim I: Develop a convex approach to multi-phase trajectory optimization of eVTOL vehicles

To fulfill the limited vertiport capacity and limited battery life, the eVTOL operations should follow a four-dimensional (4D) trajectory, which includes the 3D spatial trajectory and an extra dimension of the required time of arrival (RTA) constraint for a given concept of operation (CONOP) [67, 59]. Most eVTOL CONOPS involve more than one phase. Previous work from [59] has shown that the GPOPS solver can successfully solve two-phase eVTOL trajectory optimization problems. However, solving the multi-phase cruise-to-landing optimization problem can take up to 5 minutes with the GPOPS's pseudospectral method, which is undesirable for real-time trajectory optimization. Therefore, the first aim of this research is to design a convex optimization algorithm to solve the multi-phase eVTOL trajectory optimization problem. The resulting trajectory must safely achieve the required arrival time constraint with minimum energy consumption, and the results will be compared with those from the pseudospectral collocation method used in GPOPS.

1.2.2 Aim II: Develop a convex approach to high-fidelity landing trajectory optimization of eVTOL vehicles

Most existing research on the trajectory planning problem utilizes reinforcement learning approaches [39] or optimization methods [59, 89] with simplified dynamic models without considering complex aerodynamic characteristics due to the wind gusts and highly turbulent urban environments [44, 51]. The strong aerodynamic perturbations due to the building-wind interaction, turbulence in the urban wind field, and constantly changing weather conditions can cause the actual eVTOL flight behavior to deviate from the simplified dynamic simulation. Specifically, landing an eVTOL aircraft can be a very challenging task when the environment produces complex dynamical responses in the thrusts and moments.

To address the above challenges and provide precision, safe, and efficient eVTOL landing solutions, the second aim of this research is to develop ODE-based aerodynamic models and integrate these models with the standard flight dynamics to obtain high-fidelity optimal landing trajectories using advanced optimal control methods. Based on simulation results the traditional GPOPS solver can take from couple of minutes to even hours to solve this problem without converging to a local solution. For this study, we want to address the problem and improve the computation time by developing a novel convex optimization approach. Since convex optimization can converge in polynomial time with a global optimal solution, we believe the method will be a great fit for this problem.

1.2.3 Aim III: Develop a convex approach to merging trajectory optimization of eVTOL vehicles

Safety concerns are always the top priority for most transportation systems. In recent years, conflict avoidance has become a critical research area to guarantee high-density yet safe operations of air vehicles. In 2005, NASA proposed their air traffic management algorithm for flight separation assurance [24]. Recently, researchers have extended their effort to UAM areas [11] using deep reinforcement learning approaches [13] and nonlinear optimization-based methods [66]. However, to the best of our knowledge, none of these works has focused on the air vehicle merging problem. Merging coordination and control of vehicles along different airways or air corridors will become one of the bottleneck challenges to ensure the safety of vehicle operations while reducing congestion and travel delays for future UAM applications.

In this research, we extend the convex approach to address the merging control problem for eVTOL vehicles by leveraging computational optimal control and numerical optimization. Specifically, the problem is formulated as an optimal control problem, which is solved by a convex optimization approach [85, 80] and the results will be compared with a pseudospectral method-based solver [52].

1.3 Originality and Significance

The goal of this study is to address some challenging topics for eVTOL trajectory optimization under the concept of future UAM. Successfully solving these problems can reduce the computational speed for eVTOL trajectory generation and improve the flight performance by reducing the energy cost and considering high-fidelity disciplinary models. The proposed method is different from the traditional mechanisms because 1) to the best of our knowledge, it is the first study to use convex optimization to solve multi-phase trajectory optimization problems for eVTOL or UAM applications; 2) it is the first study focusing on using convex optimization for high-fidelity landing trajectory optimization; and 3) it is the first study using convex optimization to address merging operations of eVTOL vehicles. By taking advantage of convex optimization, it is possible to achieve real-time trajectory optimization while ensuring acceptable accuracy for eVTOL flight missions.

This project aims to address the challenges and existing issues facing the UAM field. By approaching the proposed research objectives, this research is expected to contribute a set of reliable trajectory optimization algorithms for safe and efficient operations of future eVTOL vehicles. This framework has the potential to benefit the envisioned UAM system by reducing traffic congestion, minimizing potential collision, improving energy efficiency, and producing reliable flight trajectories. Furthermore, the method and techniques proposed in this project will have broader impacts in other research fields and can be easily extended to many other areas including traditional air vehicles, ground vehicles, aircraft guidance and control, space exploration, and more.

Chapter 2

Background and Literature Review

Ever since the helicopter has been designed and made around the 1940s, people had been thinking about flying vehicles with vertical take-off and landing abilities as urban transportation tools [69]. However, due to the limited technology and operational complexity, several major accidents occurred which forced most helicopters to cease operations in urban areas between the 1960s and mid-1970s [69].

Today, the world's population has more than doubled since the 1960s which generated a very heavy demand for surface transportation. In 2021, Americans spent an average of more than 36 hours each year on road traffic, and in the most traffic-congested city in U.S. - Chicago, people lost more than 104 hours every year in traffic congestion [33]. Thus, people have begun to reimagine and explore urban air travel ideas and hope to find a solution to the traffic congestion problem. Fortunately, with the fast development of battery and aviation technologies, eVTOL aircraft are becoming a potential solution to the road-traffic congestion problems by leveraging the third-dimensional urban space to better meet the mobility demands [23].

Compared with the traditional vertical takeoff and landing vehicles like helicopters, eVTOL vehicles are potentially more efficient, flexible, and generate much less noise to the environment [60]. Government agencies including NASA, FAA, and major companies such as Boeing, Airbus, Joby Aviation, EHang, etc. are developing their eVTOL vehicles with different design concepts [69].

To fully realize the potential of eVTOL vehicles as a solution for urban mobility, efficient and precise trajectory optimization is crucial. Trajectory optimization is the process of determining the path for a vehicle to follow from one point to another while meeting various constraints such as time, energy consumption, and safety requirements. For eVTOLs, which rely on electric power, optimizing the trajectory is crucial for minimizing energy usage and extending the operational range, thereby enhancing the overall efficiency and sustainability of urban air transport systems. In addition to its role in operational efficiency, trajectory optimization is integral to the aircraft design phase through multidisciplinary design optimization (MDO) [78, 15], which allows for the simultaneous consideration of aerodynamics, structural integrity, and energy management. This approach enables the design of eVTOLs with enhanced performance characteristics, such as improved range and reduced energy consumption.

Moreover, since eVTOL vehicles are expected to operate autonomously [91], trajectory optimization plays a critical role in ensuring safe and reliable operations without human intervention. Autonomous flight systems must be capable of dynamically adjusting flight paths in real-time to avoid obstacles, adapt to changing weather conditions, and respond to unexpected events. By optimizing trajectories, eVTOLs can ensure that flights are conducted in the most energy-efficient manner, reducing the likelihood of battery depletion and enhancing operational safety. This is particularly important for eVTOLs, as they cannot simply recharge at any location or be towed to charging station like ground vehicles if they run out of power.

Therefore, limited onboard energy storage of eVTOLs makes trajectory optimization not just a matter of efficiency but also a safety-critical aspect. Unlike ground vehicles, eVTOL vehicles must manage their energy reserves carefully to complete their missions safely. Optimized trajectories ensure that energy consumption is kept within safe limits, allowing eVTOL vehicles to complete their missions without risking mid-flight power loss. This careful management of energy reserves through trajectory optimization is essential to ensuring the reliability and safety of eVTOL operations, particularly in densely populated urban environments where safety margins are minimal.

This chapter is structured as follows: In Section 2.1, we will introduce the general concept and brief history of Urban Air Mobility (UAM), providing context for the development and significance of eVTOL technologies. Section 2.2 will then dive into the specifics of eVTOL vehicles, including their types, applications, and the advantages they offer over traditional flying vehicles. Following that, Section 2.3 will focus on trajectory optimization, discussing its fundamental principles, the challenges specific to eVTOL operations, and the state-of-the-art methods currently employed. Finally, Section 2.5 will explore the beauty of convex optimization and how the convex approach can improve the trajectory optimization process for eVTOL flights.

2.1 Urban Air Mobility

UAM refers to urban transportation systems that transport people or cargo by air, which offers an alternative to traditional ground-based transportation methods. It aims to leverage the vertical dimension of the urban space to alleviate traffic congestion and provide faster, more efficient travel options within cities[30]. Recent market research on UAM has shown that air metro markets alone can generate over 2.8 billion in profit when operating among the top 15 largest U.S. cities by the time of 2030 [27]. UAM is a broad concept that covers a variety of operations including passenger transportation, medical emergency missions, logistics, precision agriculture, and much more [73].

The concept of UAM dates back to the mid-20th century with the development of helicopters. However, despite the promise of reducing traffic congestion, early efforts faced numerous challenges, including safety concerns and high operational costs, which limited public adoption. Recent advancements in battery technology, new materials, and autonomous systems have reignited interest in UAM. Innovations in electric propulsion and digital systems have made it feasible to develop and operate eVTOL vehicles, which are crucial for the success of UAM. UAM has the potential to significantly reduce travel times, improve connectivity in congested urban areas, and create new economic opportunities.

In 2020, NASA proposed a framework to categorize the UAM maturity levels (UMLs) for the anticipated evolutionary stages of the UAM system, where Level 1 indicates the initial stage, and Level 6 indicates that UAM is fully integrated into people's daily lives [30]. Based on the current definition of the UMLs, we are still under UML-Level 1, and a lot of research and certifications must be performed before we can deploy UAM system in daily life. UAM aircraft can operate both inside and outside of the UAM Operating Environment (UOE), but UAM aircraft that operate outside the UOE must follow the local air traffic regulations. Thus, the UAM aircraft must be operated more flexibly compared to traditional airplanes and general aviation.

Several cities and regions worldwide are actively exploring UAM. Examples include Los Angeles, which is testing UAM as part of its urban mobility strategy, and Dubai, which plans to launch UAM services in the near future [5]. Leading companies like Joby Aviation, EHang, and Volocopter are at the forefront of UAM development, conducting extensive testing and trials. These initiatives and efforts are part of a larger global movement to integrate UAM into urban landscapes, which aligns with strategic frameworks developed by organizations such as NASA. Under NASA'S UAM framework, they defined five major pillars/barriers: airspace system design and implementation, individual aircraft management, and operations, airspace operations management, aircraft development and production, and community integration [30]. To successfully achieve the UAM concept, researchers have been working across different research fields and disciplines. Public perception is a critical factor in the successful adoption of UAM technologies. Among the primary concerns are safety, noise pollution, and potential airspace congestion. Incidents of mechanical failure, mid-air collisions, or crashes could have severe consequences and are among the top concerns for deploying UAM services. Noise pollution is another major issue, as the constant operation of eVTOLs could significantly increase noise levels in cities, potentially disrupting daily life and affecting public health. Companies like Whisper Aero are actively researching and developing technologies to reduce aircraft noise, aiming to make eVTOLs quieter and more acceptable to urban communities [36]. Additionally, there are fears that the introduction of numerous eVTOLs could complicate air traffic management and lead to new forms of urban congestion. Fully autonomous operations, however, offer a potential solution to these challenges by enhancing precision in navigation and coordination, thereby streamlining traffic flow and reducing the likelihood of congestion. Figure 2.1 below shows the five pillars of UAM research, with this project focusing on pillar three: Individual aircraft management, which aims to make aircraft safer, smarter, and more efficient in their operations [30].

2.2 Electric Vertical Take-Off and Landing

In this proposed research, we focus on air transport services with vertical take-off and landing (VTOL) aircraft, and especially eVTOL vehicles. EVTOL vehicles are a type of aircraft that utilizes electric power (a combination of the battery pack and electrical motors), to hover, take off, and land vertically. The concept of eVTOL aircraft emerged around 2011 and was officially introduced by the American Helicopter Society (now the Vertical Flight Society) and the American Institute of Aeronautics and Astronautics (AIAA) in 2014 [53]. These vehicles are seen as a transformative solution for congested cities, offering rapid, point-to-point transportation with minimal infrastructure compared to traditional ground-based transportations.

Compared with traditional airplanes, eVTOL aircraft use vertiports to take off and land with required space even smaller than heliports and have a strong ability to hover and maneuver in the air. When compared with helicopters, eVTOL vehicles have much less noise pollution, less operational expenses, and most importantly are more secure than helicopters.

EVTOLs primarily rely on electric propulsion systems, utilizing electric motors to drive propellers or ducted fans. Advances in battery technology, particularly in lithium-ion and emerging solid-state batteries, are crucial for providing the necessary energy density and capacity to support efficient and sustainable flight. There are three common types of eVTOL configurations, each with distinct advantages.

The first is the multicopter configuration, which is characterized by multiple rotors that provide stability and simplicity in vertical flight, making them ideal for urban air taxis and short-distance travel [46]. EHang is one of the companies adopting this approach, and their eVTOL, the EHang 216 (2.2a), recently obtained both a Production Certificate and an Airworthiness Certificate in April 2024.

The second configuration is the tiltrotor design, which combines the vertical takeoff and landing capabilities of a helicopter with the forward flight efficiency of an airplane. This

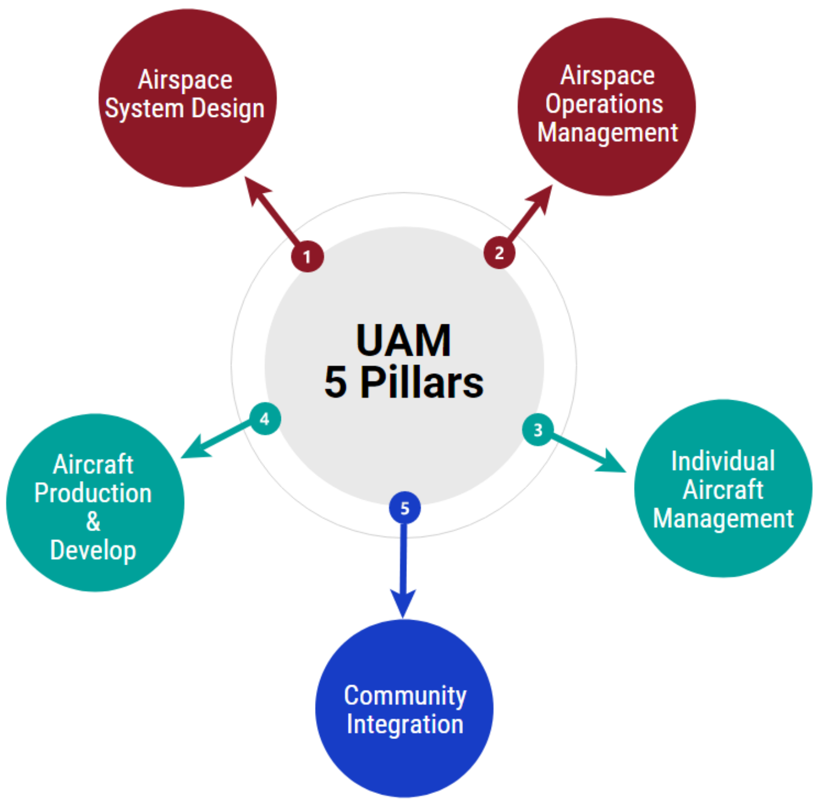


Figure 2.1: Five pillars of UAM.

design features rotors that can tilt to provide vertical lift during takeoff and landing and horizontal thrust during flight, allowing for greater range and speed compared to multicopters [68]. An example of this approach is Joby Aviation's S4 (2.2b), which has been extensively tested and is set to begin commercial operations in the near future .

The third configuration is the fixed-wing hybrid, also known as the Lift + Cruise design, which integrates the benefits of both rotary and fixed-wing aircraft. This design uses rotors for vertical lift and a fixed wing for efficient horizontal flight, significantly extending the range and operational efficiency of the aircraft. Volocopter's VoloConnect (2.2c) is a notable example of this type, designed for regional air mobility, offering the ability to travel longer distances with reduced energy consumption [43].

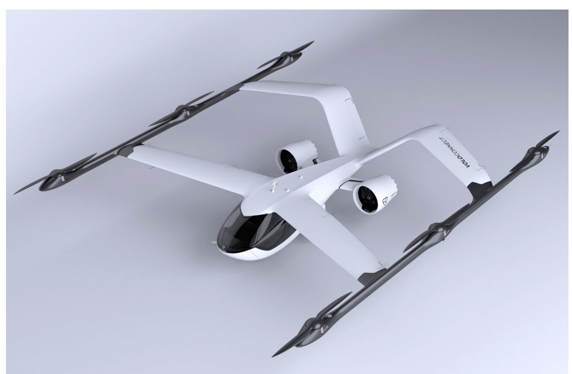
Modern eVTOLs are designed to equip with advanced autonomous systems that enable precise navigation and operation. These systems include autopilots, GPS, LIDAR, and computer vision technologies, which ensure safe and efficient flight in complex urban environments. In 2014, Dr. Andrea and his group in Zurich successfully developed a control algorithm that can prevent quadcopter from crashing with three dead motors [49]. More recently, in 2021, Sihao [70] proposed an algorithm that combines fault-tolerant control and onboard vision-based state estimation, which can make a quadrotor successfully achieve a mission subjected to the complete failure of one rotor. Thus, The integration of AI in navigation and control systems further enhances the capability of eVTOLs to operate autonomously, reducing the need for pilot intervention and increasing operational safety.

Overall, apart from the challenges including design, manufacturing, operational management, and flight control, eVTOL vehicles are highly desirable under the envisioned UAM concept of operations. However, because of the current technology limit and complicated infrastructure development process, most of the existing Unmanned Aircraft Systems (UAS) Traffic Management (UTM) and operations of eVTOLs suffer from limited battery endurance and vertiport capacity during early UAM operations [73]. Researchers have been working on multiple disciplines such as infrastructure design [34], traffic control [35], and vehicle control [59] to improve the safety and efficiency of operations.

In this research, we aim to focus on trajectory optimization, a critical aspect of improving the safety and efficiency of autonomous flight for eVTOL vehicles. Trajectory optimization

(a) Multicopter Design - EH216

(b) Tiltrotor Design - Joby Aviation S4



(c) Fixed-Wing Hybrid Design - VoloConnect

Figure 2.2: Different eVTOL configurations: (a) multicopter design, (b) tiltrotor design, and (c) fixed-wing hybrid design.

not only minimizes the risk of collisions and operational failures but also contributes to more efficient energy management, which is vital for the performance and sustainability of eVTOLs. A detailed background review of trejactory optimization will be provided in the next section 2.3.

2.3 Trajectory Optimization

In this section, we want to first briefly introduce the entire framework for a completely autonomous system, and then move to the topic of trajectory optimization. As shown in Figure 2.3 below, a general autonomous flight control system consists of several subsystems including navigation and perception, mission planning, tactical planning, trajectory optimization, feedback control, and actuator commands [45].

The first step in an autonomous flight system is navigation and perception, where data from various sensors is collected and processed to understand the vehicle's environment and position. This includes systems like the Global Positioning System (GPS), which provides accurate location data, and vision-based navigation systems that use cameras and other sensors to interpret the surroundings [65, 32]. Advances in sensor fusion techniques have allowed for more robust and accurate perception, enabling eVTOLs to navigate complex urban environments more effectively.

Next, the mission planning phase involves defining the mission goals, flight constraints, and possible waypoints for the flight mission. This process requires input from the user to outline the objectives and constraints of the flight. For instance, a mission might involve transporting medical devices across a city, requiring the planner to identify potential obstacles and regulatory constraints. Research in this area has explored various methodologies, including model-based approaches that rely on predefined models of the environment and data-driven approaches that use machine learning to adapt to real-time conditions [14, 35, 59].

In the tactical planning phase, the system must adapt the mission goals to the real-world environment, considering dynamic factors such as other air traffic, weather conditions, and changing regulatory restrictions. This involves updating the mission plan with real-time data to ensure safe and efficient flight paths. Techniques such as real-time path re-planning and obstacle avoidance are critical in this stage to handle unexpected changes in the flight environment.

Then, trajectory optimization is required to find a trajectory with feasible state and control profiles while satisfying some flight objectives. Some early research on eVTOL trajectory planning includes tilt-wing eVTOL takeoff trajectory optimization [16], eVTOL merging control [89], and wind-optimal trajectory planning [58].

After trajectory optimization, feedback control comes into play to ensure that the eVTOL follows the planned trajectory accurately, despite disturbances such as wind gusts, sensor inaccuracies, or other environmental factors. Feedback control systems use real-time data to adjust the control inputs and maintain the desired flight path. Model predictive control (MPC) is a widely used approach in this context, providing a way to predict future states of the system and adjust controls accordingly [25, 2]. For eVTOLs, MPC can offer robust control by continuously updating the control signals based on the latest sensor data and environmental conditions [31].

Lastly, the control command will be sent to the actuator (e.g., distributed propulsors) to produce the required control forces and moments.

This work focus on trajectory optimization, one of the key modules to enable safe and efficient eVTOL operations. Trajectory optimization problems have a long history. The first trajectory optimization problem, the brachistochrone problem, was proposed in 1697 by Swiss mathematician Johann Bernoulli, with the objective to find the fastest descent path between two points [71]. The famous calculus of variation method can obtain the solution to this problem. The terms trajectory optimization and optimal control are sometimes interchangeable when a problem is studied to design a trajectory that minimizes or maximizes some performance index while satisfying a set of constraints [61]. Overall, the purpose of trajectory optimization is to find an open-loop solution for the vehicle to follow by solving the defined optimal control problem.

In general, an optimal control problem involves finding a control history $u(t)$ and the associated states $x(t)$ to maximize or minimize a given objective function of the form [38]:

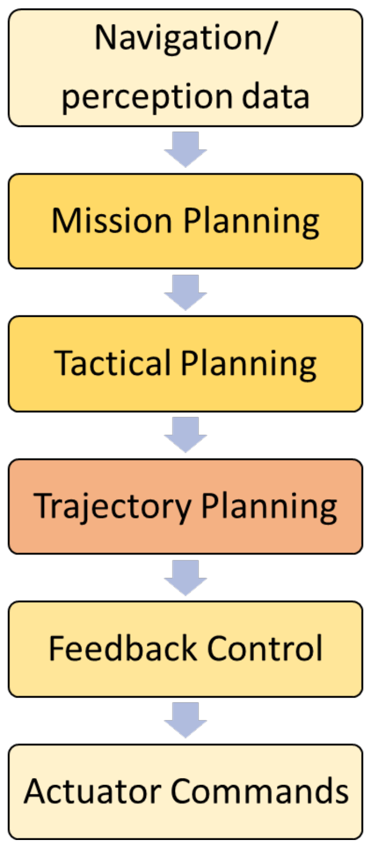


Figure 2.3: Road map for an autonomous control system.

$$J = \int_{t_0}^{t_f} f(t, x(t), u(t)\, dt + \Theta(x(tf)) \tag{2.1}$$

, where Θ is the terminal cost. The problem is subject to a set of constraints, including system dynamic constraints, path constraints, and boundary conditions. The dynamic constraints usually are a set of differential equations that take the following general form:

$$\dot{x}(t) = g(t, x(t), u(t)) \tag{2.2}$$

For some classical problems and problems with low-dimensional linear systems, the solution can be derived analytically from the necessary and sufficient optimality conditions. However, optimal control problems with highly nonlinear differential equations are difficult to solve analytically in general, and people have been developing a variety of numerical methods to solve the problem [61].

Numerically, trajectory optimization problems are usually solved by two main methods: the indirect and the direct methods. The indirect methods leverage the calculus of variations or Pontryagin's maximum principle to determine the first-order optimality conditions for the problem and then transfer the original problem to a two-point (or multi-point) boundary value problem [76]. The indirect methods take a lot of effort to derive the adjoint differential equations and require a good initial guess for the problem, which can be burdensome work in real-world applications. The direct methods, on the other hand, do not require explicit derivation of optimality conditions for the original problem. Instead, the direct methods discretize the states and/or controls of the problem using transcription methods such as direct collocation or direct shooting to convert the original continuous-time optimal control problem into a finite-dimensional parameter optimization problem, which is a nonlinear programming (NLP) problem in general [76]. The resulting NLP problem can then be solved using NLP algorithms such as the sequential quadratic programming (SQP) method or the penalty function method. In recent years, thanks to powerful interior-point NLP solvers such as IPOPT [77], direct methods have been widely used in aerospace and transportation areas. However, these methods suffer from some drawbacks. First, solving NLP problems, in particular those with highly nonlinear dynamics and complex constraints, is computationally expensive and the solution time is usually unpredictable. Second, the algorithm can not guarantee to converge to the optimal solution, even a local optimum. Third, an accurate initialization of the problem is usually required for most problems. Consequently, these approaches may not be suitable for onboard, real-time eVTOL trajectory optimization for future UAM applications, where flight safety and performance are of vital importance.

2.4 Pseudospectral Method

Pseudospectral methods represent a specialized subset of direct methods in optimal control, characterized by their ability to discretize continuous-time control problems into parameter optimization problems. In this research, we propose using the GPOPS solver as a baseline to compare our solutions. GPOPS is built on the pseudospectral method [55].

Pseudospectral methods have gained traction within the field of aerospace trajectory optimization and are now integral to several commercial optimal control software solutions due to their superior accuracy, lower initial value sensitivity, and quicker convergence rates [64]. The essence of pseudospectral discretization lies in using N^{th}-order weighted interpolating polynomials at orthogonal collocation points to approximate continuous trajectories.

The pseudospectral method typically involves three key elements: the domain transformation, N^{th}-order collocation points at y_N, and the weighting function W.

For finite-horizon optimal control problems, the time domain $[t_0, t_f] \ni t$ is mapped to the computational domain $[-1, 1] \ni \tau$ using the affine transformation:

$$t = \Gamma(\tau) = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}, \quad \tau = \Gamma^{-1}(t) = \frac{2}{t_f - t_0}(t - \frac{t_f + t_0}{2}) \tag{2.3}$$

The state trajectory $X(\tau)$ is approximated on these collection points using Lagrange polynomials as:

$$X(\tau) \approx X_N(\tau) = \sum_{i=0}^{N} X_i L_i(\tau), \quad -1 \leq \tau \leq 1 \tag{2.4}$$

where $L_i(\tau)$ represents the N^{th}-order Lagrange interpolating polynomial.

Once the time domain is discretized, and mappings of states and controls are defined, the dynamics function is transcribed to approximation equations by differentiating the interpolating polynomial:

$$\dot{X}(\tau_k) \approx \dot{X}_N(\tau_k) = \sum_{i=0}^{N} X_i \dot{L}_i(\tau_k) \tag{2.5}$$

where $\dot{L}_i(\tau_k)$ is derived from the Gauss pseudospectral differentiation matrix.

Combining these elements, the state-space constraints for the collocated control problem can be expressed as:

$$\sum_{i=0}^{N} D_{ki} X_i = \frac{t_f - t_0}{2} f(X_k, u_k, \tau_k) \tag{2.6}$$

where $f(X_k, u_k, \tau_k)$ denotes the dynamics function.

Through these steps, the continuous-time optimal control problem is transformed into a discrete-time numerical optimization problem and can be solved with NLP solvers.

2.5 Convex Optimization

To overcome the potential issue from traditional optimal control method, over the last few years, convex optimization has been widely used to solve optimal control and trajectory optimization problems in aerospace engineering due to the advantages of fast convergence and guaranteed global minima for a single convex problem [74, 41, 40]. Convex optimization is a subset of optimization that deals with problems where the objective function is convex and the feasible region is a convex set. This ensures that any local minimum is also a global minimum, making these problems easier to solve compared to non-convex problems. Over the years, convex optimization has been used in fields ranging from operations research to machine learning, and now plays a vital role in autonomous systems and aerospace engineering [79].

If a problem can be formulated as a convex optimization problem, such as linear programming (LP), quadratic programming (QP), second-order cone programming (SOCP), or semidefinite programming (SDP), then the relaxed problem can be solved in polynomial time due to its low complexity and convex structure [12]. As such, convex optimization approaches have great potential for onboard, real-time applications. One popular numerical method to solve convex optimization problems is the interior-point method. Interior-point methods solve optimization problems by iteratively approximating the feasible region from the inside and moving towards the boundary where the optimal solution lies [63]. Tools like CVX [28], Gurobi [3], and MOSEK [6], which all adopt this method, are widely used for solving convex optimization problems. These tools provide robust solvers for various types of convex problems and are integral in both research and industry applications.

Nevertheless, two significant challenges for the convex approaches are how to recognize a convex problem and how to transform a nonconvex problem into a convex problem. Sequential convex programming (SCP) has emerged as a method that solves nonconvex optimization problems by finding the solutions of a sequence of convex sub-problems [22]. Sequential convex programming (SCP) addresses these challenges by solving non-convex optimization problems through a series of convex sub-problems [22]. In the SCP approach, convex elements of the problem remain untouched while non-convex components are modified using convex approximations or relaxations. This often involves the introduction of slack variables, the relaxation of constraints, and the linearization of highly nonlinear dynamics through first-order Taylor series expansions [4]. Then, the solution can be obtained numerically by using the interior-point methods with appropriate solvers.

The SCP method has found significant applications in aerospace, where it has been crucial for scenarios demanding high precision and reliability. Specific applications include powered descent guidance for Mars landings, where the method's ability to handle complex, dynamic environments under strict fuel and time constraints is invaluable [4, 72]. It has also been applied to launch vehicle trajectory planning, demonstrating its effectiveness in managing the stringent ascent profiles necessary to achieve orbit [7, 8, 9, 81]. Furthermore, SCP has been utilized in spacecraft rendezvous operations, which require precise maneuvering to dock or approach other spacecraft [84, 48, 83]. In addition to these applications, the use of convex optimization extends to UAV wireless communication networks, optimizing trajectory and power control to ensure effective data transmission across variable terrains [52].

Convex optimization offers robust solutions that are critical for the advancement of aerospace technology and autonomous systems. As computational resources and theoretical methods continue to evolve, the potential for further integration of convex optimization into more complex and varied applications remains vast. The ongoing development of algorithms and techniques promises to enhance the capabilities and efficiency of these systems, driving forward the frontiers of technology in aerospace and beyond.

Chapter 3

Aim I: Develop a convex approach to multi-phase trajectory optimization of eVTOL vehicles

In our previous work [85], we formulated a single-phase, cruise-to-landing eVTOL trajectory optimization problem and solved it using a convex optimization approach that has shown to be able to converge to an optimal trajectory in real-time. Single-phase eVTOL trajectory optimization can provide us with a quick, accurate solution but may not be applicable to real-world scenarios due to environmental constraints and additional mission requirements.

Most envisioned UAM missions involve multiple flight phases, including take-off, climb, cruise, hover, descent, and landing. Therefore, in this research, we develop a novel convex optimization approach to obtaining numerical solutions to the minimum-energy-cost, multi-phase eVTOL trajectory optimization problem. First, the problem is formulated as a general nonlinear trajectory optimization with a fixed time of arrival (RTA) constraint. Then, the nonlinearity in the flight dynamics is reduced via a convenient change of variables. Finally, the nonconvex control constraint is relaxed into a convex form, and the solution to the original trajectory optimization problem is sought by solving a series of SOCP problems using the SCP algorithm. Two different eVTOL cruise-to-landing CONOPs are solved with the proposed method. Results of the simulation cases are provided to demonstrate the performance of the proposed approach through comparisons with a state-of-the-art pseudospectral optimal control solver [17].

3.1 Problem Formulation

In this section, we present the details of the problem formulation. The dynamic system utilized in this study is developed based on a quad-rotor eVTOL aircraft and is described in subsection 3.1.1. Following that the flight constraints are discussed in subsection 3.1.2. Finally, the objective function and the overall trajectory optimization problem are presented in subsection 3.1.3.

3.1.1 System Dynamics

For this research, the dynamic model of the eVTOL vehicle is adapted from [89]. The vehicle is assumed to take a geodesic path, and only the longitudinal motion of the vehicle is modeled to formulate a two-dimensional (2-D) trajectory optimization problem. The flight dynamic equations are as follows:

$$\dot{x} = V_x \tag{3.1}$$

$$\dot{z} = V_z \tag{3.2}$$

$$\dot{V}_x = \frac{T \sin\theta}{m} - \frac{D_x}{m} \tag{3.3}$$

$$\dot{V}_z = \frac{T \cos\theta}{m} - \frac{D_z}{m} - g \tag{3.4}$$

where x (along-track distance) and z (altitude) compose the position vector of the vehicle; V_x and V_z represent the horizontal and vertical components of the vehicle speed, respectively; T is the net thrust; m is the vehicle mass; g is the gravitational acceleration; and θ is the rotor tip-path-plane pitch angle. The state vector and the control vector for the system are defined as

$$\mathbf{y} = [x, z, V_x, V_z]^T \quad \text{and} \quad \mathbf{u} = [T, \theta]^T, \tag{3.5}$$

respectively. the maximum ground speed of the eVTOL vehicle is assumed not exceed 100 km/h. Thus, the drag force on the fuselage of the aircraft can be calculated from the incompressible flow theory, and the net drag is assumed to be equivalent to the drag force on the fuselage. As such, the horizontal (D_x) and vertical (D_z) components of the aerodynamic drag force can be obtained from [92],[59]

$$D_x = \frac{\rho V_x^2 C_D S_x}{2} \quad \text{and} \quad D_z = \frac{\rho V_z^2 C_D S_z}{2}, \tag{3.6}$$

respectively, where ρ is the atmosphere density, C_D is the aerodynamic drag coefficient, and S_x and S_z are the reference front and top flat plate areas of the fuselage, respectively. For simplicity, all these four variables are assumed to be constant in this study. Therefore, D_x and D_z are only dependent on the velocity components.

3.1.2 Flight Constraints

To ensure the safety of the flight mission and passenger comfort, the eVTOL vehicle has to satisfy several constraints. In this study, the eVTOL vehicle is assumed to take a fixed RTA to arrive at the target vertiport; however, the RTA at the intermediate, cruise-to-descent transition waypoint is free to be optimized according to different CONOPs. Thus, a two-phase free-waypoint-time problem can be formed. First, the initial and terminal conditions for the aircraft is imposed by introducing the following boundary conditions:

$$\mathbf{y}(t_0) = \mathbf{y}_0 \tag{3.7}$$

$$\mathbf{y}(t_f) = \mathbf{y}_f \tag{3.8}$$

Then, we define the maximum along-track distance and maximum altitude as follows:

$$0 \leq x \leq x_{\max} \tag{3.9}$$

$$0 \leq z \leq z_{\max} \tag{3.10}$$

Meanwhile, the maximum speed and maximum thrust are bounded based on vehicle specifications:

$$\sqrt{V_x^2 + V_z^2} \leq V_{\max} \tag{3.11}$$

$$0 \leq T \leq T_{\max} \tag{3.12}$$

Moreover, the vehicle's pitch angle, θ, is restricted based on the passenger's comfort level as shown below:

$$|\theta| \leq \theta_{\max} \tag{3.13}$$

Because the problem considered in this study consists of two phases (i.e., cruise and descent/landing), linkage constraints are imposed to ensure the continuity between these two phases. The general linkage constraints are shown below:

$$\mathbf{y}^p(t_f^p) - \mathbf{y}^{p+1}(t_0^{p+1}) = 0 \tag{3.14}$$

$$t_f^p - t_0^{p+1} = 0 \tag{3.15}$$

where p is set to 1 because only the transition from cruise to descent/landing is considered in this research.

3.1.3 Performance Index and Optimal Control Problem

For the multi-phase trajectory optimization problem, the performance index is defined to minimize the overall control effort with respect to the entire mission, and the objective function is shown below:

$$J = \int_{t_0}^{t_f} \frac{1}{2} T^2 dt \tag{3.16}$$

Then, the performance index of the two-phase problem can be written as follows:

$$J = \sum_{p=1}^{2} \int_{t_0^p}^{t_f^p} \frac{1}{2} T^2 dt \tag{3.17}$$

With the objective function and constraints, an optimal control problem can be formulated as:

Problem 1:

$$\text{Minimize:} \quad (3.17)$$

$$\text{Subject to:} \quad (3.1), (3.2), (3.3), (3.4), (3.7), (3.8), (3.9), (3.10),$$
$$(3.11), (3.12), (3.13), (3.14), (3.15)$$

Overall, the problem aim to minimize the objective function (3.17) while satisfying the system dynamics (3.1)-(3.4), the boundary conditions (3.7) and (3.8), the inequality state and control constraints (3.9)-(3.13), and the linkage constraints (3.14) and (3.15).

Remark 1: In Problem 1, the objective function is convex. However, the dynamics are nonlinear, and the controls and states are highly coupled, which will cause NLP algorithms hard to converge. In the following section, we will apply a series of convexification techniques to transform Problem 1 into a convex optimization problem and develop a sequential convex approach to solve the solution.

3.2 Problem Reformulation via Convex Relaxations

As discussed above, one potential issue with the direct methods is that the computational time may become significant when the problem is highly nonlinear and requires a large number of iterations to converge. To address this issue, we will leverage recent advances in convex optimization that allows us to obtain the optimal solutions in real time. Multi-phase convex optimization method has been successfully used to solve problems such as reentry trajectory optimization [94] and rocket launch trajectory optimization [9, 10]. Motivated by these previous works, we develop and customize a convex approach to the multi-phase eVTOL trajectory optimization problem for UAM scenarios in this section.

It is worth mentioning that considered multi-phase UAM problem is fundamentally different from the multi-phase problems solved in [9, 10, 94] in terms of the path constraints, relaxed control constraints, and objective functions involved. For example, bounds on the state variables are enforced in this research such that the vehicle flies within more confine airspace. More relaxed control constraints are considered in this research due to the constrained tilt angle for safe and comfort UAM missions. Further, minimum control effort, instead of minimum flight time, is considered in the objective function. As a result, the underlying assumptions that ensure the validity of convexification are different, and the convexification procedure needs to be re-derived.

The convexification process for the considered multi-phase trajectory optimization problem is adapted partially from our previous work [85]; however, the linkage constraints make the approach much more complicated than that of the single-phase problem. The process consists of a change of variables and convexification of the control constraint and dynamics detailed in subsection 3.2.1, an alternative strategy for addressing possible nonequivalent relaxations in subsection 3.2.2, convexification of the free time-of-arrival at the intermediate waypoint in subsection 3.2.3, and problem discretization in subsection 3.2.4, followed by a multi-phase SCP algorithm in subsection 3.2.5.

3.2.1 The First Convexification Approach

From Problem 1, we recognize that the system dynamics are nonlinear, which will affect the convergence of the method. To address this issue, we first introduce the following new variables:

$$u_1 = T\sin\theta, \quad u_2 = T\cos\theta, \quad \text{and} \quad u_3 = T^2 \tag{3.18}$$

Along with the above new variables, a new constraint should be imposed based on the trigonometric identities:

$$u_1^2 + u_2^2 = u_3 \tag{3.19}$$

Then, the original system dynamics in (3.1)-(3.4) become:

$$\dot{x} = V_x \tag{3.20}$$

$$\dot{z} = V_z \tag{3.21}$$

$$\dot{V}_x = \frac{u_1}{m} - \frac{D_x}{m} \tag{3.22}$$

$$\dot{V}_z = \frac{u_2}{m} - \frac{D_z}{m} - g \tag{3.23}$$

and the objective function takes the following new form:

$$J = \sum_{p=1}^{2} \int_{t_0^p}^{t_f^p} \frac{1}{2} u_3 dt \tag{3.24}$$

The state boundary conditions and inequality constraints remain the same; however, a set of inequality constraints has to be enforced on the new controls to ensure that (3.12) and (3.13) are met:

$$0 \leq u_3 \leq T_{\max}^2 \tag{3.25}$$

$$u_2 \tan(-\theta_{\max}) \leq u_1 \leq u_2 \tan(\theta_{\max}) \tag{3.26}$$

Notice that the feasible set defined by the new equality control constraint in (3.19) represents the surface of a convex cone, which is nonconvex. However, we apply a relaxation technique to convert the original constraint to an inequality constraint with the following form:

$$u_1^2 + u_2^2 \leq u_3 \tag{3.27}$$

Remark 2: Note that constraint (3.26) is valid only within a specific range of θ. In this research, considering real-world eVTOL operations and passenger comfort in UAM applications, the pitch angle θ is bounded between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, which makes constraint (3.26) equivalent to the original constraint (3.13).

Now, u_3 is bounded by (3.26) and (3.27), which together constitute a solid convex cone defined by two convex constraints. The new inequality constraint enables the control variables to reach the inside of the convex cone, rather than being constrained on its surface. Therefore, the new constraints define a larger feasible set than the original one. The geometric representation of this relaxation is shown in Fig. 3.1.

The equivalence of the constraint relaxation in (3.27) largely depends on the constraint settings, particularly the lower and upper bounds of the path constraints in Problem 1 and the new control constraints in (3.25) and (3.26). These settings essentially shape the feasible solution space and thus influence the solution of the trajectory optimization problem. The equivalence of the relaxation can be proved by constructing the Karush-Kuhn-Tucker (KKT) conditions with complementary slackness under some assumptions as detailed below.

Assumption 1: The control constraints in (3.25) and (3.26) are inactive almost everywhere on $[t_0, t_f]$, i.e., the thrust of the vehicle does not become zero or reach its maximum value almost everywhere on $[t_0, t_f]$, and the pitch angle of the vehicle during the flight does not reach its minimum or maximum angular limit almost everywhere on $[t_0, t_f]$.

Assumption 2: The along-track distance x does not touch its lower or upper limit almost everywhere on $[t_0, t_f]$, i.e., the constraint in (3.9) is inactive almost everywhere on $[t_0, t_f]$. Also, the altitude z does not touch its lower limit almost everywhere on $[t_0, t_f]$, i.e., the constraint $0 \leq z$ in (3.10) is inactive almost everywhere on $[t_0, t_f]$.

Proposition 1: If $\{\mathbf{y}^*(t), \mathbf{u}^*(t)\}$ is an optimal solution to the problem after relaxation, then the optimal control $\mathbf{u}^*(t)$ lies on the boundary of the cone defined by (3.27), i.e., $u_1^*(t)^2 + u_2^*(t)^2 = u_3^*(t)$ is satisfied almost everywhere on $[t_0, t_f]$ under Assumptions 1 and 2.

Proof: We can arrive at the conclusion drawn in Proposition 1 by leveraging the direct adjoining method in optimal control [50]. First, the Hamiltonian for the relaxed problem is formulated as follows:

$$H(\mathbf{y}, \mathbf{u}, \lambda_0, \lambda) = \lambda_0 \frac{1}{2} u_3 + \lambda_1 V_x + \lambda_2 V_z + \lambda_3 \left(\frac{u_1}{m} - \frac{D_x}{m} \right) + \lambda_4 \left(\frac{u_2}{m} - \frac{D_z}{m} - g \right) \tag{3.28}$$

where $\lambda_0 \geq 0$ is a constant, and $\lambda = [\lambda_1; \lambda_2; \lambda_3; \lambda_4]$ is the costate vector. Note that, the constraint on the upper limit of z in (3.10) and the maximum speed constraint in (3.11) are not required to be inactive. To derive the KKT conditions for minimizing H over the

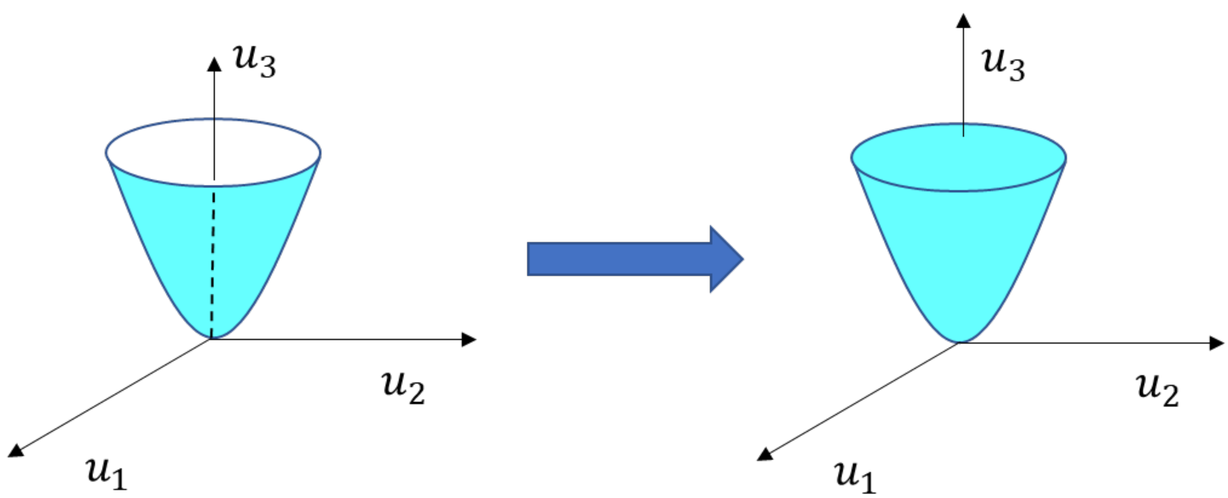


Figure 3.1: Relaxation of control constraint.

constraints, we introduce the Lagrangian L as follows based on Assumptions 1 and 2:

$$L(\mathbf{y},\mathbf{u},\lambda_0,\lambda,\nu) = H(\mathbf{y},\mathbf{u},\lambda_0,\lambda)+\nu_1(u_3-u_1^2-u_2^2)+\nu_2(z_{\max}-z)+\nu_3(V_{\max}^2-V_x^2-V_z^2) \tag{3.29}$$

where $\nu = [\nu_1;\nu_2;\nu_3]$ is the Lagrangian multiplier vector. Note that, the airspeed constraint in (3.11) has been rewritten into an equivalent form $V_x^2+V_z^2 \leq V_{\max}^2$ for simpler derivation. The costate equations are derived below:

$$\dot{\lambda}_1 = -\frac{\partial L}{\partial x} = 0 \tag{3.30}$$

$$\dot{\lambda}_2 = -\frac{\partial L}{\partial z} = \nu_2 \tag{3.31}$$

$$\dot{\lambda}_3 = -\frac{\partial L}{\partial V_x} = -\lambda_1 + \lambda_3\frac{\rho C_D S_x V_x}{m} + 2\nu_3 V_x \tag{3.32}$$

$$\dot{\lambda}_4 = -\frac{\partial L}{\partial V_z} = -\lambda_2 + \lambda_4\frac{\rho C_D S_z V_z}{m} + 2\nu_3 V_z \tag{3.33}$$

The stationary conditions are as follows:

$$\frac{\partial L}{\partial u_1} = \frac{\lambda_3}{m} - 2\nu_1 u_1 = 0 \tag{3.34}$$

$$\frac{\partial L}{\partial u_2} = \frac{\lambda_4}{m} - 2\nu_1 u_2 = 0 \tag{3.35}$$

$$\frac{\partial L}{\partial u_3} = \frac{\lambda_0}{2} + \nu_1 = 0 \tag{3.36}$$

The complementary slack conditions are below:

$$\nu_1 \geq 0, \quad \nu_1(u_3^* - u_1^{*2} - u_2^{*2}) = 0 \tag{3.37}$$

$$\nu_2 \geq 0, \quad \nu_2(z_{\max} - z^*) = 0 \tag{3.38}$$

$$\nu_3 \geq 0, \quad \nu_3(V_{\max}^2 - V_x^{*2} - V_z^{*2}) = 0 \tag{3.39}$$

The Informal Theorem 4.1 in [50] states that there exist a constant λ_0, a continuous costate vector $\lambda(t)$, and a continuous multiplier vector $\nu(t)$ such that the nontriviality condition $[\lambda_0;\lambda(t);\nu_2(t);\nu_3(t)] \neq \mathbf{0}$ holds for $t \in [t_0,t_f]$ and the optimality conditions in Eqs. (3.30)-(3.39) are satisfied almost everywhere on $[t_0,t_f]$.

The proof can be achieved by contradiction. To this end, we assume that Proposition 1 does not hold and the control constraint in (3.27) is inactive at the optimal solution to the relaxed problem, i.e., $u_1^*(t)^2 + u_2^*(t)^2 < u_3^*(t)$, for some time interval $t \in [t_1, t_2]$ within $[t_0, t_f]$. Then, from (3.37) we have $\nu_1 = 0$. Using $\nu_1 = 0$ in (3.34) and (3.35), we can get $\lambda_3 = \lambda_4 = 0$. With this result, we solve (3.32) and (3.33) for λ_1 and λ_2 respectively as follows:

$$\lambda_1 = 2\nu_3 V_x \tag{3.40}$$

$$\lambda_2 = 2\nu_3 V_z \tag{3.41}$$

From (3.30), we know that λ_1 is constant, while V_x is varying during the flight. Therefore, to make (3.40) hold, we have $\nu_3 = 0$ and then $\lambda_1 = 0$. Further, we get $\lambda_2 = 0$ from (3.41). Up to this point, we have shown that $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$, $\nu_1 = 0$, and $\nu_3 = 0$. Using $\lambda_2 = 0$ in (3.31), we have $\nu_2 = 0$. At last, solving (3.36) for λ_0 we obtain $\lambda_0 = -2\upsilon_1 = 0$. Therefore, we have proved that $[\lambda_0; \lambda(t); \nu_2(t); \nu_3(t)] = \mathbf{0}$, which contradicts the nontriviality condition $[\lambda_0; \lambda(t); \nu_2(t); \nu_3(t)] \neq \mathbf{0}$. As such, we have proved Proposition 1, i.e., the relaxed control constraint is active at the optimal solution, and the convex relaxation of the control constraint in (3.27) is equivalent. $\square$

Finally, the only nonconvex term left in the problem is the nonlinear dynamic system defined by (3.20)-(3.23). Based on the state and control vectors defined in (3.5), the equations of motion can be rewritten as a state representation as follows:

$$\dot{\mathbf{y}} = f(\mathbf{y}) + B\mathbf{u} \tag{3.42}$$

where

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} V_x \\ V_z \\ -\frac{\rho V_x^2 C_D S_x}{2m} \\ -\frac{\rho V_z^2 C_D S_z}{2m} - g \end{bmatrix} \quad \text{and} \quad B\mathbf{u} = \begin{bmatrix} 0 \\ 0 \\ \frac{u_1}{m} \\ \frac{u_2}{m} \end{bmatrix} \tag{3.43}$$

To convexify (3.42), we apply a successive linearization method to replace this nonlinear equation with a first-order Taylor series expansion around a reference solution that will be updated sequentially (will be detailed in subsection 3.2.5). With the linear approximation method, the state equation becomes:

$$\dot{\mathbf{y}} \approx f(\mathbf{y}^*) + A(\mathbf{y}^*)(\mathbf{y} - \mathbf{y}^*) + B\mathbf{u} \tag{3.44}$$

where A is the state matrix defined by partial derivatives and evaluated at the reference solution as shown below:

$$A(\mathbf{y}^*) = \left[\frac{\partial f(\mathbf{y})}{\partial \mathbf{y}}\right]_{\mathbf{y}=\mathbf{y}^*} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & & 0 & 1 \\ 0 & 0 & -\frac{\rho V_x C_D S_x}{m} & 0 \\ 0 & 0 & 0 & -\frac{\rho V_z C_D S_z}{m} \end{bmatrix}_{\mathbf{y}=\mathbf{y}^*} \tag{3.45}$$

and B is a constant control matrix:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \end{bmatrix} \tag{3.46}$$

Lastly, a trust-region constraint is added to the problem to improve the convergence of the sequential optimization process:

$$|\mathbf{y} - \mathbf{y}^*| \leq \boldsymbol{\delta} \tag{3.47}$$

where $\boldsymbol{\delta}$ is a fixed trust-region radius.

Remark 3: In general, a smaller trust region restricts the search space and may lead to slower convergence, although it can potentially prevent the optimization process from taking too large steps, which would potentially lead to inaccurate or infeasible solutions. In contrast, a larger trust region allows for more aggressive search and possibly faster convergence at the risk of overshooting the optimal solution and causing divergence of the algorithm. The trust-region size can be potentially adjusted based on the iteration process of the algorithm for better convergence [82]. In this project, we choose to fix the trust-region radius and focus on the development and demonstration of a basic SCP method. However, the impact of the trust-region size on the developed method for the specific problem considered will be demonstrated by the simulation results presented in subsection 3.3.3.

Remark 4: Assumptions 1 and 2 can be satisfied by the optimal solutions when the bounds on the states and controls are properly defined and no extreme maneuvers need to be performed by the vehicle; otherwise, the equivalence may not be guaranteed, and the method may converge to a solution that may not be a solution to the original problem. This motivates our second convexification approach that eliminates the need of u_3 and the corresponding control relaxation, which will be detailed in the following subsection.

3.2.2 The Second Convexification Approach

Although constraint relaxation is a common technique to convexify optimization problems, its effectiveness is highly dependent on the problem setting. For the UAM problem considered in this research, it is possible that the relaxed control constraint (3.27) may not be guaranteed to be active at the optimal solution at all times due to the possible violation of the underlying assumptions as mentioned in *Remark 4*. This observation has led us to develop a second approach. Instead of introducing variable u_3, we propose to only introduce u_1 and u_2 as the control variables:

$$u_1 = T\sin\theta, \quad u_2 = T\cos\theta \tag{3.48}$$

Therefore, the constraints (3.25) and (3.27) can be combined as:

$$u_1^2 + u_2^2 \le T_{\max}^2 \tag{3.49}$$

with the same inequality constraint for u_1 and u_2 as in (3.26). Then, the objective function can be rewritten as follows:

$$J = \sum_{p=1}^{2} \int_{t_0^p}^{t_f^p} \frac{1}{2}(u_1^2 + u_2^2)\, dt \tag{3.50}$$

Remark 5: Note that the new control constraint (3.49) is equivalent to the original thrust constraint (3.12), which already represents a convex set. Therefore, no further relaxation is needed and the objective function is already convex, leading to a simpler problem formulation due to the absence of both T and u_3.

Remark 6: It is worth mentioning that the above two convexification methods follow different strategies while sharing the same goal of relaxing the original problem into a convex problem. Both approaches are presented in this research because each of them has its pros and cons. The first convexification approach has been frequently used in the literature and leads to potentially more general convex subproblems; however, its validity holds under specific assumptions as detailed above. After discretization, the first approach will result in an SOCP problem, while the second approach will lead to a convex QP problem, which is a subclass of SOCP. In the situations where the underlying assumptions and equivalent relaxation are guaranteed, the first approach is expected to produce more accurate solutions than the second approach. In some cases such as the scenarios studied in Section 3.3, the second approach seems superior to the first one in terms of convergence and stability.

3.2.3 Convexification of Free Transition Time

For the two-phase eVTOL trajectory optimization problem, the arrival time to the transition waypoint is usually free to optimize. Consequently, to account for the free travel time in the procedure, we define a new variable τ over a fixed domain $[0, 1]$ to replace the original time variable t [10]. The time dilation σ between t and τ is defined as:

$$\sigma = \frac{dt}{d\tau} = t_f - t_0 \tag{3.51}$$

Then, the objective functions (3.24) and (3.50) are therefore respectively changed to:

$$J = \sum_{p=1}^{2} \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} u_3 \sigma \, d\tau \tag{3.52a}$$

$$J = \sum_{p=1}^{2} \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} (u_1^2 + u_2^2) \sigma \, d\tau \tag{3.52b}$$

Since we have introduced the new time variable σ to the problem, the new objective function is nonconvex. Similar to the linear approximation in Eq. (3.44), we convexify the objective functions by taking the first-order linear approximation around the reference trajectory obtained from the last step of the SCP process as follows:

$$J = \sum_{p=1}^{2} \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2}\Big[\bar{u}_3\bar{\sigma} + \bar{\sigma}(u_3 - \bar{u}_3) + \bar{u}_3(\sigma - \bar{\sigma})\Big] d\tau \tag{3.53a}$$

$$J = \sum_{p=1}^{2} \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2}\Big[(\bar{u}_1^2 + \bar{u}_2^2)\bar{\sigma} + \bar{\sigma}(u_1^2 + u_2^2 - \bar{u}_1^2 - \bar{u}_2^2) + (\bar{u}_1^2 + \bar{u}_2^2)(\sigma - \bar{\sigma})\Big] d\tau \tag{3.53b}$$

For the new equations of motion due to the free time of arrival at the transition waypoint, we apply a similar successive linearization method to replace the nonlinear equations with a first-order Taylor series expansion around a reference solution that will be updated sequentially within SCP. With the linear approximation method, the dynamic equations become:

$$\mathbf{y}' = \frac{d\mathbf{y}}{d\tau} = \sigma\mathbf{f}(\mathbf{y}(\tau), \mathbf{u}(\tau)) \approx A\mathbf{y} + B\mathbf{u} + F\sigma + c \tag{3.54}$$

where A, B, F, and c are obtained from below:

$$A(\tau) = \bar{\sigma}\frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \tag{3.55}$$

$$B(\tau) = \bar{\sigma}\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \tag{3.56}$$

$$F(\tau) = \mathbf{f}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) = \mathbf{f}(\bar{\mathbf{y}}) + B\bar{\mathbf{u}} \tag{3.57}$$

$$c(\tau) = -(A\bar{\mathbf{y}} + B\bar{\mathbf{u}}) \tag{3.58}$$

Note that all the variables with a bar on top demote the reference solution and value that will be updated and obtained from the previous step of the SCP procedure that will be detailed in subsection 3.2.5.

With the above development and through the application of the two convexification approaches, the overall optimal control problems are now defined respectively as follows:
Problem 2 (first convexification approach):

$$\text{Minimize:} \quad (3.53a)$$

$$\text{Subject to:} \quad (3.54), (3.7) - (3.11), (3.14), (3.15), (3.25) - (3.27), (3.47)$$

Problem 3 (second convexification approach):

$$\text{Minimize:} \quad (3.53b)$$
$$\text{Subject to:} \quad (3.54), (3.7) - (3.11), (3.14), (3.15), (3.49), (3.47)$$

Remark 7: Through a series of techniques including change of variables, linear approximation, and convexification of the free time-of-arrival at the intermediate waypoint as detailed above, the original problem (Problem 1) has been transformed into a convex continuous-time optimal control problem (Problem 2 or Problem 3) with or without the convex relaxation of control constraint. To facilitate the implementation of convex optimization algorithms and obtain numerical solutions, the continuous-time problem must be discretized into a finite-dimensional numerical optimization problem, which is described in the following subsection.

3.2.4 Discretization

To obtain a numerical solution, we will discretize the continuous-time optimal control problem (Problem 2 and Problem 3) into a parameter optimization problem with a finite set of variables and constraints that can be solved by numerical optimization solvers. In this research, we apply a trapezoidal rule for discretization [80]. First, we split the independent variable $\tau \in [0, 1]$ of each phase into N segments with $N + 1$ nodes. The step size is $\Delta\tau = (1-0)/N$, and the discretized nodes are denoted by$\{\tau_0, \tau_1, ..., \tau_N\}$ with $\tau_i = \tau_{i-1} + \Delta\tau$, $i = 1, 2, ..., N$. The corresponding state and control profiles are then discretized at the $N+1$ points.

By applying the trapezoidal rule, the current state $\mathbf{Y}_i$ can be obtained from the previous step $\mathbf{Y}_{i-1}$ as follows:

$$\begin{aligned}\mathbf{Y}_i = \mathbf{Y}_{i-1} + \frac{\Delta\tau}{2}[(&\sigma^{k-1}A_{i-1}^{k-1}\mathbf{Y}_{i-1} + \sigma^{k-1}B\mathbf{u}_{i-1} + \sigma(\mathbf{f}_{i-1}^{k-1} + B\mathbf{u}_{i-1}^{k-1}) - (\sigma^{k-1}A_{i-1}^{k-1} + \sigma^{k-1}B\mathbf{u}_{i-1}^{k-1})) \\ &+(\sigma^{k-1}A_i^{k-1}\mathbf{Y}_i + \sigma^{k-1}B\mathbf{u}_i + \sigma(\mathbf{f}_i^{k-1} + B\mathbf{u}_i^{k-1}) - (\sigma^{k-1}A_i^{k-1} + \sigma^{k-1}B\mathbf{u}_i^{k-1}))]\end{aligned} \tag{3.59}$$

After rearrangement, we have:

$$
\begin{aligned}
&(I - \frac{\Delta\tau}{2}\sigma^{k-1}A_i^{k-1})\mathbf{Y}_i - (I + \frac{\Delta\tau}{2}\sigma^{k-1}A_{i-1}^{k-1})\mathbf{Y}_{i-1} - \frac{\Delta\tau}{2}\sigma^{k-1}B\mathbf{u}_i - \frac{\Delta\tau}{2}\sigma^{k-1}B\mathbf{u}_{i-1} = \\
&\frac{\Delta\tau}{2}[(\sigma(f_{i-1}^{k-1} + B\mathbf{u}_{i-1}^{k-1}) - (\sigma^{k-1}A_{i-1}^{k-1} + \sigma^{k-1}B\mathbf{u}_{i-1}^{k-1})) \\
&+ (\sigma(f_i^{k-1} + B\mathbf{u}_i^{k-1}) - (\sigma^{k-1}A_i^{k-1} + \sigma^{k-1}B\mathbf{u}_i^{k-1}))]
\end{aligned}
\tag{3.60}
$$

The linearized objective functions in Eqs. (3.53a) and (3.53b) can be discretized and represented respectively as follows:

$$
J = \frac{\Delta\tau}{2}\sum_{p=1}^{2}\sum_{i=1}^{N-1}\left[\bar{u}_{3i}^{k-1}\bar{\sigma}^{k-1} + \bar{\sigma}^{k-1}(u_{3i} - \bar{u}_{3i}^{k-1}) + \bar{u}_{3i}^{k-1}(\sigma - \bar{\sigma}^{k-1})\right] \tag{3.61a}
$$

$$
\begin{aligned}
J = \frac{\Delta\tau}{2}\sum_{p=1}^{2}\sum_{i=1}^{N-1}\Big[&(\bar{u}_{1i}^{2(k-1)} + \bar{u}_{2i}^{2(k-1)})\bar{\sigma}^{k-1} \\
&+ \bar{\sigma}^{k-1}(u_{1i}^2 + u_{2i}^2 - \bar{u}_{1i}^{2(k-1)} - \bar{u}_{2i}^{2(k-1)}) + (\bar{u}_{1i}^{2(k-1)} + \bar{u}_{2i}^{2(k-1)})(\sigma - \bar{\sigma}^{k-1})\Big]
\end{aligned}
\tag{3.61b}
$$

Lastly, because an RTA constraint is considered in this research, the final arrival time for the eVTOL mission is fixed. Therefore, we add the following time constraint to ensure that the sum of the flight times for both phases meets the final time constraint:

$$
\sum_{p=1}^{2}\sigma^p = t_f \tag{3.62}
$$

After the above convexification and discretization process, the final optimization problems are established below:

Problem 4 (first convexification approach):

$$
\begin{aligned}
&\text{Minimize:} \quad (3.61a) \\
&\text{Subject to:} \quad (3.60), (3.7) - (3.11), (3.14), (3.15), (3.25) - (3.27), (3.47), (3.62)
\end{aligned}
$$

Problem 5 (second convexification approach):

$$\text{Minimize:} \quad (3.61b)$$
$$\text{Subject to:} \quad (3.60), (3.7) - (3.11), (3.14), (3.15), (3.49), (3.47), (3.62)$$

Remark 8: At this point, Problem 2 and Problem 3 have been discretized into Problem 4 and Problem 5, respectively, which are SOCP problems that minimize linear objective functions subject to linear equality and inequality constraints as well as second-order cone constraints. If a solution to Problem 4 and Problem 5 exists, then it is guaranteed to be a globally optimal solution. However, due to the approximations made above, we cannot solve a single Problem 4 or Problem 5 to obtain the optimal solution to the original trajectory optimization problem (Problem 1). Instead, a successive convex approach is presented and described in the next section to obtain an approximate optimal solution to Problem 1.

It is worth noting that SOCP is a special type of convex optimization problems. A general convex optimization problem has the following structure [12]:

$$\begin{aligned} &\text{Minimize} \quad f_0(\mathbf{y}) \\ &\text{Subject to} \quad f_i(\mathbf{y}) \leq b_i \quad i = 1, ..., m \end{aligned} \tag{3.63}$$

where the functions f_i, $i = 0, 1, \ldots, m$: $\mathbf{R}^n \rightarrow \mathbf{R}$ are convex, satisfying

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \tag{3.64}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$ and all α, $\beta \in \mathbf{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$.

3.2.5 Multi-Phase Sequential Convex Programming

In this work, for the first time, an SCP method is developed to obtain the solution to the multi-phase nonconvex UAM trajectory optimization problem. Figure 3.2 shows a flow chart that illustrates the SCP algorithm developed in this research. The SCP method for solving the two-phase problem is described in the following steps:

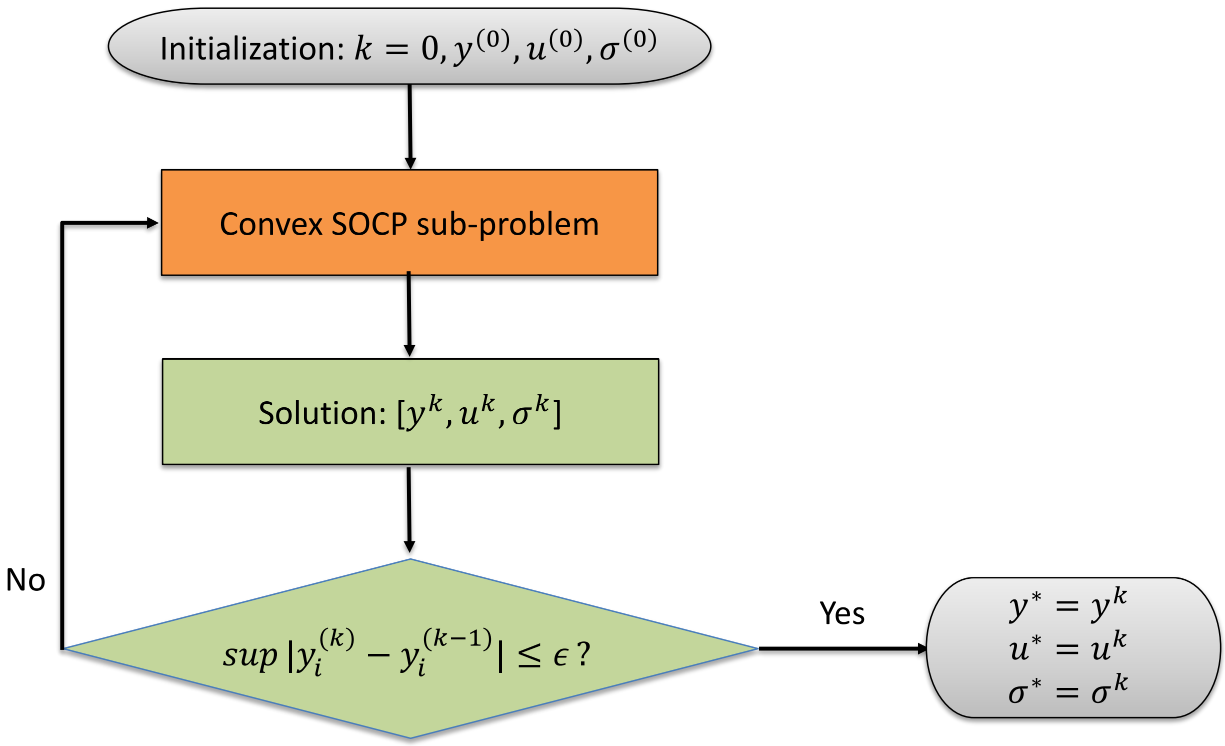


Figure 3.2: SCP flow chart.

1. Initialize the iteration index $k = 0$ and phase number $i = 0$. Specify the initial state $\mathbf{y}_i(t_0)$. Propagate the equations of motion with the initial condition to obtain an initial trajectory $\mathbf{y}^{(0)}$. In this research, the initial trajectory is obtained by simple linear interpolation between the initial and final states of the first phase.

2. Change the phase number to $i = 1$ and repeat step 1 to obtain the initial trajectory for the second phase. Set $k = k + 1$.

3. For $k > 0$, parameterize the convex subproblem (Problem 4 or Problem 5) using the solution from the previous iteration to find a solution $\{\mathbf{y}_i^{(k)}, \mathbf{u}_i^{(k)}, \sigma_i^{(k)}\}(i = 0, 1)$ at the current step.

4. Check the convergence condition

$$\forall i \in \{0, 1\}, \quad |y_i^{(k)} - y_i^{(k-1)}| \leq \varepsilon_i \tag{3.65}$$

where $\boldsymbol{\varepsilon}$ is a preset tolerance. If the condition is satisfied, the algorithm moves to step 5; otherwise, set $k = k + 1$ and go back to step 3.

5. The algorithm is converged and a solution for the problem is found to be $\{\mathbf{y}_i^{(k)}, \mathbf{u}_i^{(k)}, \sigma_i^{(k)}\}(i = 0, 1)$.

3.3 Numerical Simulation

In this section, we validate the effectiveness and performance of the proposed convexification approach through numerical simulations based on the EHang 184 model. The vehicle parameters are listed in Table 3.1, the initial and terminal conditions are shown in Table 3.2, and the numbers of the discretized segments are defined in Table 3.3. The eVTOL vehicle starts at a cruise speed of 13.83 m/s and 20,000 m away from the destination vertiport. All the simulations are performed on a desktop with a 64-bit operating system and an AMD Ryzen 7 1800X Eight-Core Processor. The convex optimization problems derived in this research are implemented in the YALMIP [42] optimization toolbox with ECOS [21] as the convex optimization solver.

Table 3.1: Vehicle parameters for simulations

Parameter	Value
Vehicle's mass, m	240 kg
Reference front plate area, S_x	2.11 m^2
Reference top plate area, S_z	1.47 m^2
Drag coefficient, C_D	1
Atmospheric density, ρ	1.225 kg/m^3
Gravitational acceleration, g	9.81 m/s^2
Maximum along-track distance, $x_{\max}$	20,000 m
Maximum altitude, $z_{\max}$	500 m
Maximum airspeed, $V_{\max}$	27.78 m/s
Maximum net thrust, $T_{\max}$	4,800 N
Maximum rotor tip-path-plane pitch angle, $\theta_{\max}$	6 deg
Time of flight, t_f	25 min

Table 3.2: Initial and terminal conditions

Parameter	Value
Initial along-track distance, x_0	0 m
Initial altitude, z_0	500 m
Initial along-track airspeed,V_{x0}	13.83 m/s
Initial vertical airspeed, V_{z0}	0 m/s
Terminal along-track distance, x_f	20,000 m
Terminal altitude, z_f	0 m
Terminal along-track airspeed, V_{xf}	0 m/s
Terminal vertical airspeed, V_{zf}	0 m/s

Table 3.3: Number of discretized nodes for each phase of two CONOPs

CONOPs	N of Phase One	N of Phase Two
1	50	50
2	50	70

For both simulation cases considered in this study, the trust-region size $\boldsymbol{\delta}$ in Eq. (3.47) is selected as $\boldsymbol{\delta} = [200\,\mathrm{m}, 5\,\mathrm{m}, 3\,\mathrm{m/s}, 3\,\mathrm{m/s}, 0.1 \cdot \sigma_{1\mathrm{init}}, 0.1 \cdot \sigma_{2\mathrm{init}}]^T$ where $\sigma_{1\mathrm{init}}$ and $\sigma_{2\mathrm{init}}$ are the guessed flying times for the first and second phases, respectively. The stopping criteria $\boldsymbol{\varepsilon}$ in Eq. (3.45) is selected as $\boldsymbol{\varepsilon} = [0.5\,\mathrm{m}, 0.5\,\mathrm{m}, 0.3\,\mathrm{m/s}, 0.3\,\mathrm{m/s}, 0.8\,\mathrm{s}, 0.8\,\mathrm{s}]^T$.

3.3.1 Case One: Vertical Descent and Landing

Figure 3.3 shows the schematic for the first scenario considered. Under this scenario, the eVTOL vehicle will cruise to the fixed top of descent (TOD) point right above the vertiport and then perform a vertical descent to the destination. The flight time to the mid waypoint is free to optimize while the final arrival time is fixed to be 1,500 seconds.

Both convexification methods are used to solve the problem and the results are compared in Figs. 3.4 to 3.6. The subplots on the left illustrate the results of the first convexification approach, while the subplots on the right display the outcomes of the second convexification approach. Figure 3.4 presents the convergence of the objective value by the two convexification approaches. It can be observed that both approaches reached a steady state. However, the second approach demonstrated convergence in just five iterations, whereas the first approach required 10 iterations. It is also important to note that, since the objective functions for both problems differ, they converge to distinct final optimal objective values, yet both achieve steady states.

Figures 3.5 illustrate the convergence of differences between consecutive intermediate steps in the vertical directions. Both convexification approaches have meet the final convergence criteria. The Δ operation measures the maximum difference in the state variable between the current step and the previous step, and Δx and Δz are defined as $\Delta x := \max|x^{(k)}(\tau_i) - x^{(k-1)}(\tau_i)|$ and $\Delta z := \max|z^{(k)}(\tau_i) - z^{(k-1)}(\tau_i)|$, respectively, where $i = 1, 2, ..., N$. In each step, ECOS takes about 1.5 seconds to solve each convex subproblem. If a faster computational time of the algorithm is favored, we can either reduce the number of discretized nodes or choose a looser convergence tolerance for the algorithm.

Figure 3.6 shows a clear convergence of the along-track airspeed vs. time profiles with curves converge from “cool” (dark blue) to “warm” (red) color. As mentioned in previous sections, the initial trajectory for the SCP method is a linear interpolation between the

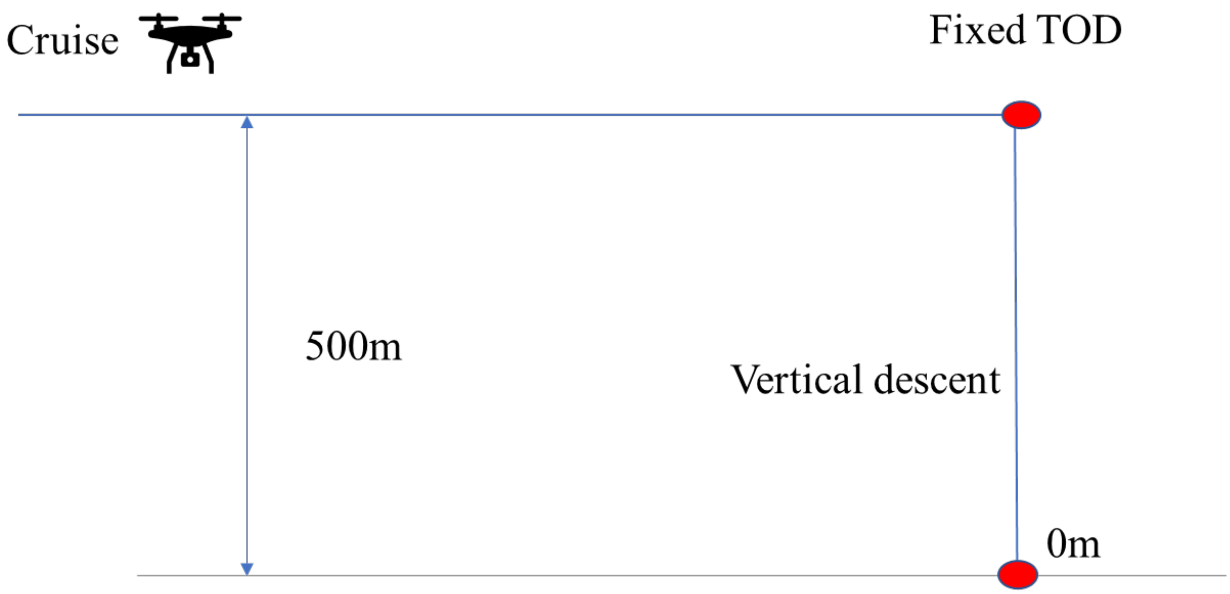
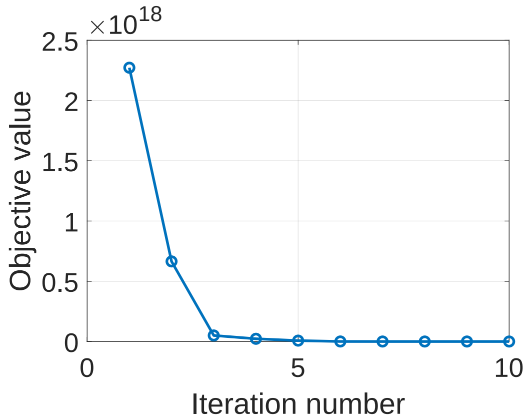


Figure 3.3: Schematic for the first scenario.

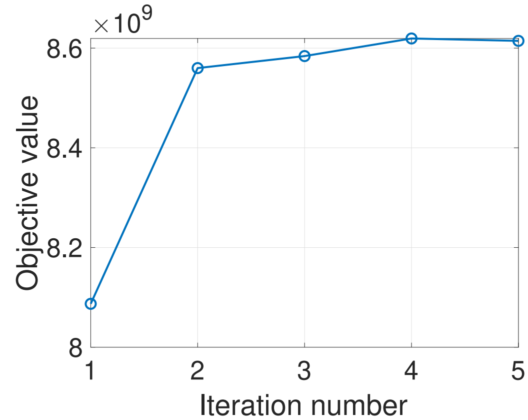
starting and landing points. Note that the algorithm gradually converges from simple straight-line guesses at earlier steps to smooth velocity profiles in the final solutions. This is aligned with the fact that the intermediate steps of SCP usually do not produce the optimal solutions but the converged solution serves as an approximate optimal solution to the original problem. If a more accurate initial trajectory is chosen, faster convergence is expected with fewer iterations.

For comparison, GPOPS is used to solve the original problem (Problem 1) under the first scenario. Figures 3.7 to 3.11 show the detailed comparisons between the results from the GPOPS solver and our proposed SCP methods. From Fig. 3.7, we can see that both solvers and both convexification approaches converge to very similar altitude profiles, and the optimal trajectories almost overlap with each other. This alignment can also be observed from the airspeed profiles in Figs. 3.8 and 3.9. From Figs. 3.10 and 3.11, we can see that the net thrust curves obtained by the SCP methods are smoother than that of the GPOPS solver in the last descent and landing phase. However, the pitch angle curves from the SCP algorithms have more jitters during the second phase. Overall, the control curves from both solvers follow very similar trend, and all the terminal constraints as well as the state and control constraints are satisfied.

Furthermore, the GPOPS solver requires approximately 20 seconds to converge. In contrast, the SCP method exhibits a quicker convergence: with the first approach, it converges in 10 iterations, taking a total of 10 seconds; with the second approach, it converges in just five iterations, requiring less than six seconds. The results of the first scenario demonstrate the effectiveness of our developed SCP algorithm, which reaches a solution of similar accuracy but with faster computational speed. In the first simulation case, both convexification approaches converged to very similar results. However, the second approach meets the convergence criteria much faster than the first, which may be caused by the possible breakdown of Assumptions 1 and 2 that are required to guarantee the exactness of the relaxed constraint (3.27) throughout the flight mission.

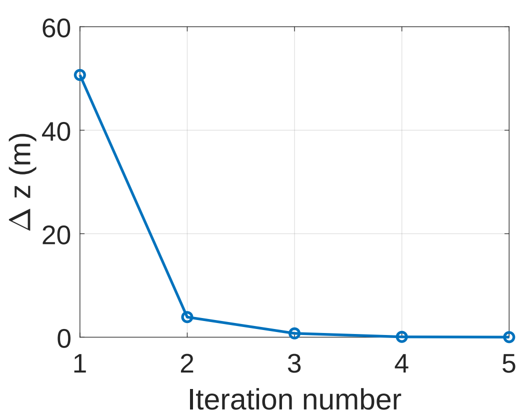


(a) First convexification approach

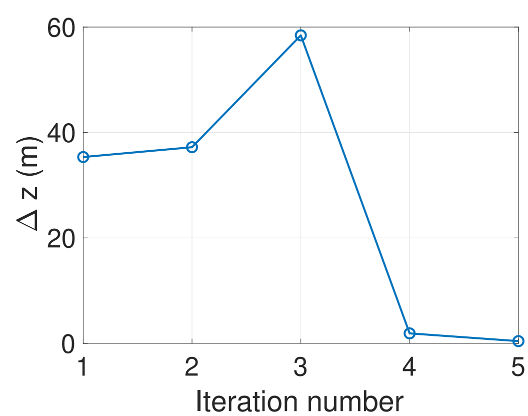


(b) Second convexification approach

Figure 3.4: Convergence of objective value for the first scenario.

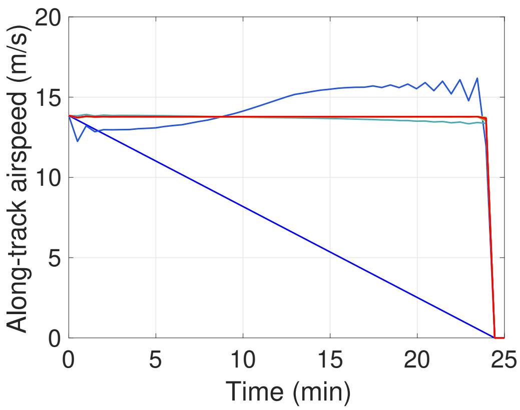


(a) First convexification approach

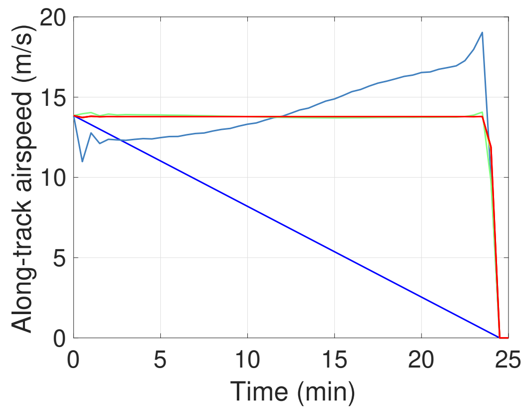


(b) Second convexification approach

Figure 3.5: Convergence of altitude for the first scenario.

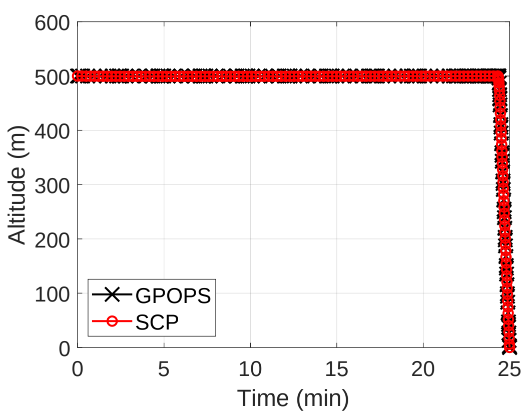


(a) First convexification approach

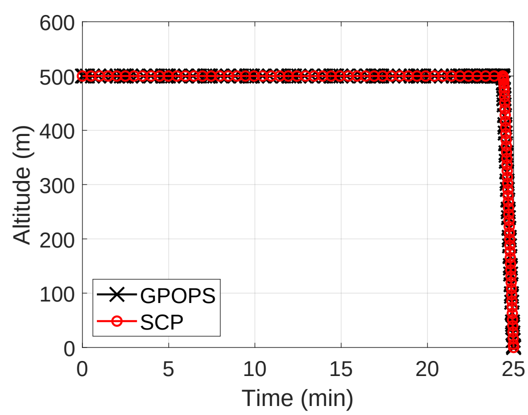


(b) Second convexification approach

Figure 3.6: Convergence of along-track airspeed for the first scenario.

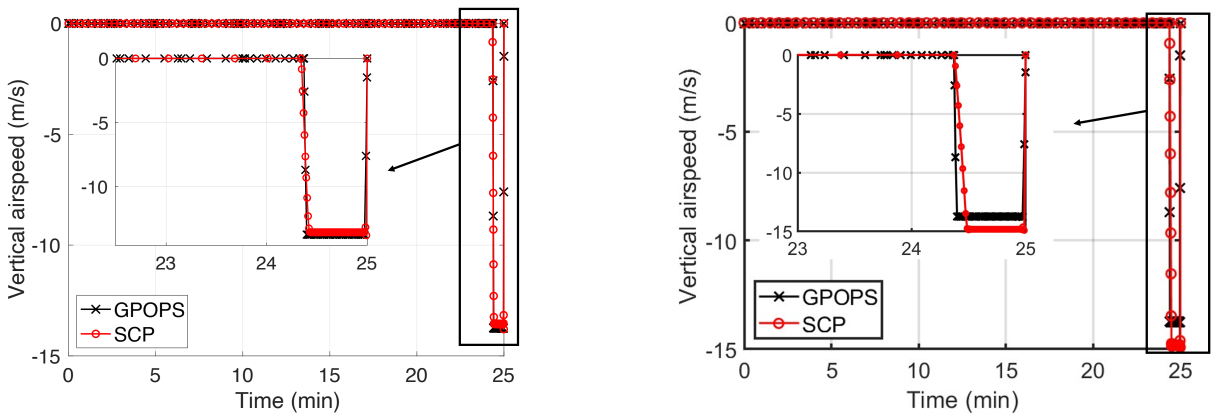


(a) First convexification approach



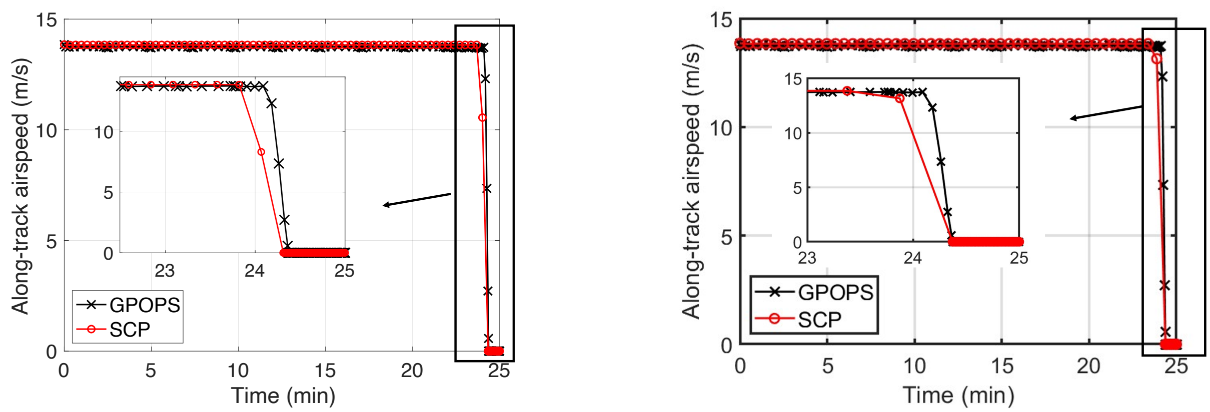
(b) Second convexification approach

Figure 3.7: Altitude vs. time profiles for the first scenario.



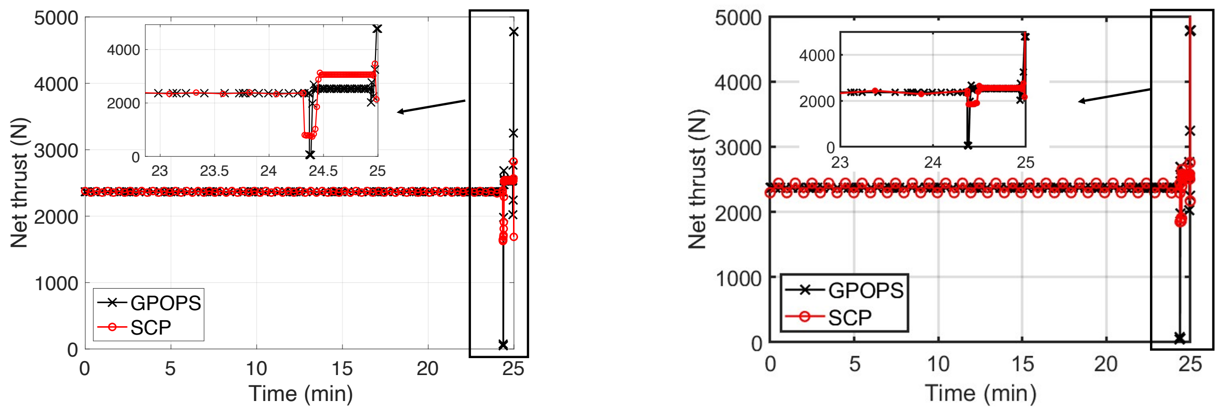
(a) First convexification approach **(b)** Second convexification approach

Figure 3.8: Vertical speed vs. time profiles for the first scenario.



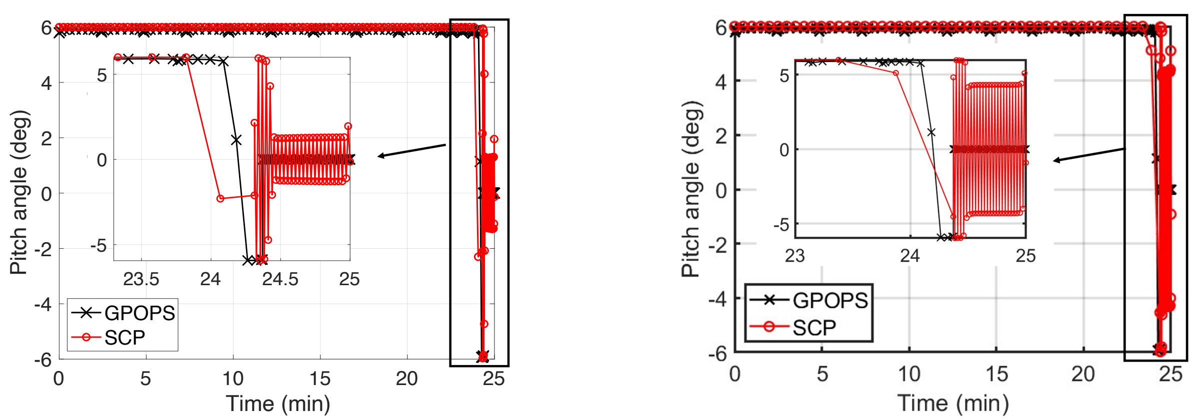
(a) First convexification approach **(b)** Second convexification approach

Figure 3.9: Along-track speed vs. time profiles for the first scenario.



(a) First convexification approach **(b)** Second convexification approach

Figure 3.10: Net thrust vs. time profiles for the first scenario.



(a) First convexification approach **(b)** Second convexification approach

Figure 3.11: Pitch angle vs. time profiles for the first scenario.

3.3.2 Case Two: Inclined Descent and Landing

In the second scenario as shown in Fig. 3.12, the eVTOL vehicle starts at a cruise speed and altitude, flies to a fixed TOD position, and then descends to the destination vertiport within the specified time of flight. In this simulation, the TOD point is set at halfway between the start and end points with a specific horizontal distance from the target vertiport. The TOD position can be modified to match real-world UAM demand. The flight time to mid waypoint is again free to optimize. This scenario usually happens in the suburban or rural area, where the approach and landing trajectory could be more flexible in a wider operating environment. Once again, both the GPOPS solver and the SCP algorithm are employed to solve the same problem. The results derived from the two convexification approaches as well as GPOPS are then compared with each other.

Figure 3.13 depicts the variation in the objective value at each iteration when applying the SCP method. The curves clearly demonstrate that both convexification approaches achieved convergence in five iterations. Figures 3.14 and 3.15 display the convergence of the changes in the distances along the x and z directions, respectively. Figure 3.16 shows the trajectories of all iterations from the SCP method. Again, the dark blue color indicates the initial trajectory and the red color represents the converged optimal trajectory. We can see that the proposed SCP method converges very quickly, and the trajectories become very close and almost overlapped after two to three steps. The convergence plots from Figs. 3.13 to 3.16 for both convexification approaches exhibit very similar trends. These plots suggest a high degree of similarity in the performance and convergence characteristics of the two methods.

Detailed comparisons between the results from GPOPS and that from the SCP method are shown in Figs. 3.17 to 3.21. Figure 3.17 shows that the optimal trajectories follow a similar trend but converge to a slightly different mid waypoint arrival time and descend and land along different trajectories in the second phase. The difference between the obtained waypoint arrival times is less than 1% of the entire flight. In addition, it can be noted from Figs. 3.18 to 3.21 that the SCP method converges to much smoother state and control profiles than GPOPS. High-frequency jitters are observed from the curves by GPOPS, which

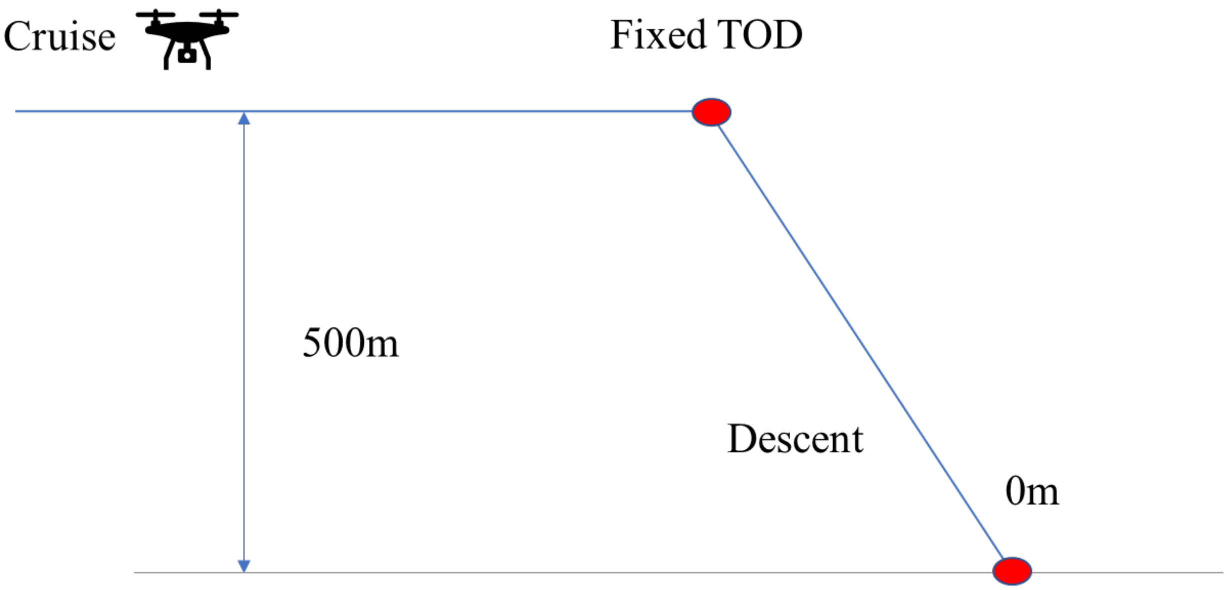
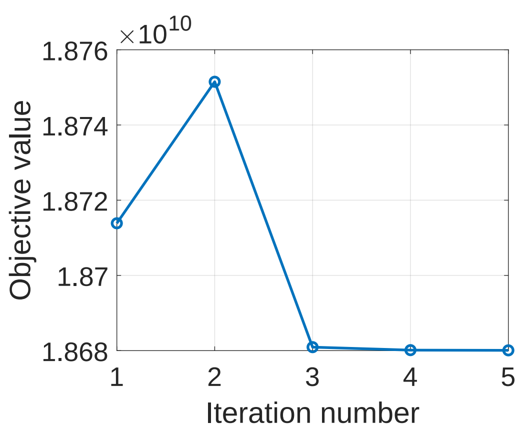
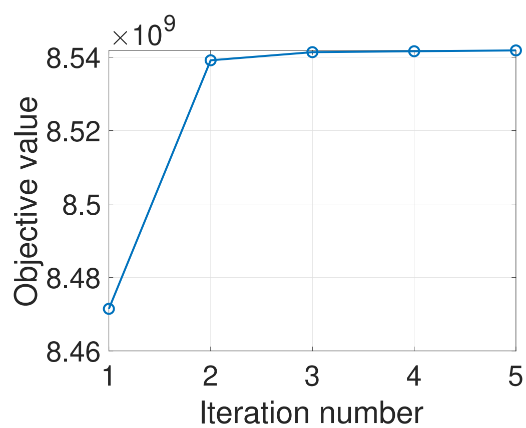


Figure 3.12: Schematic for the second scenario.

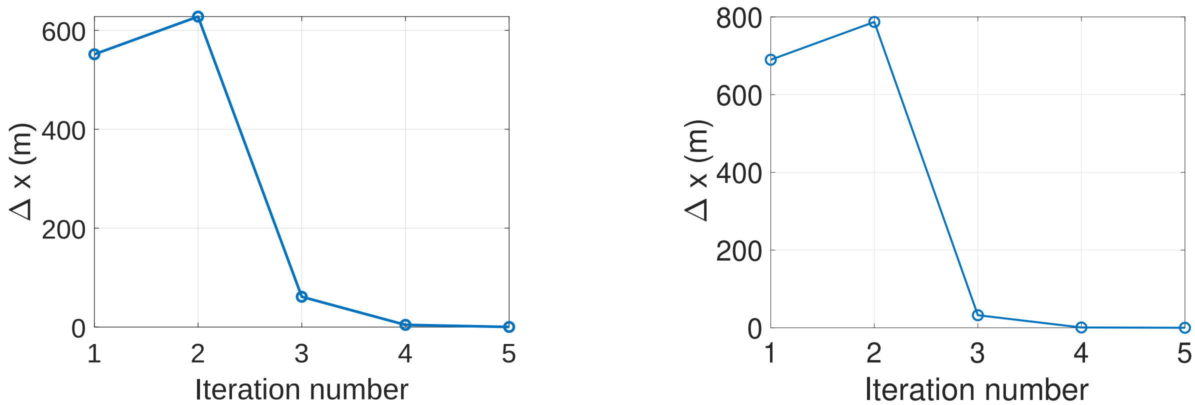


(a) First convexification approach



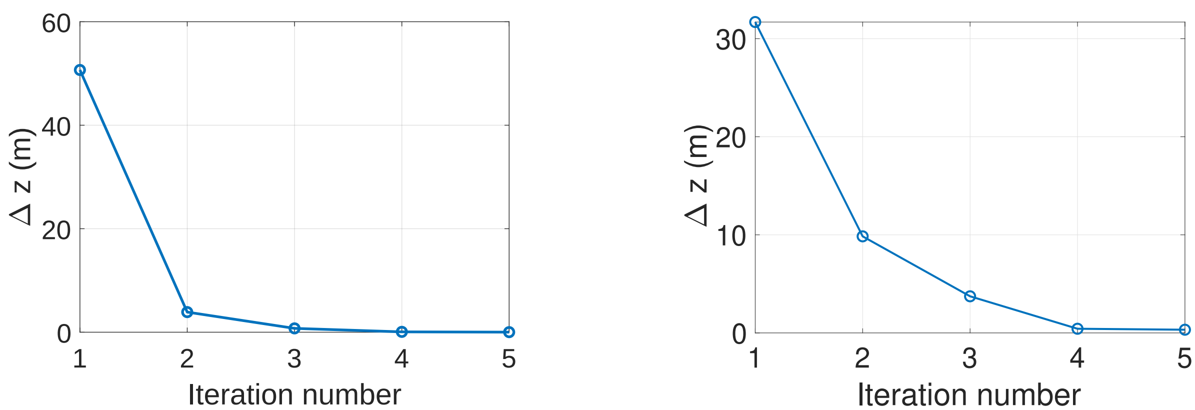
(b) Second convexification approach

Figure 3.13: Convergence of objective value for the second scenario.



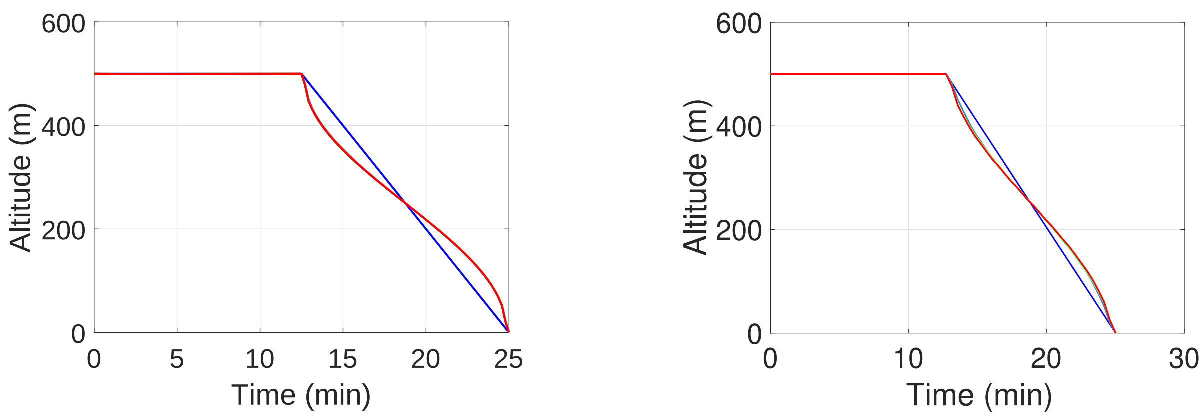
(a) First convexification approach **(b)** Second convexification approach

Figure 3.14: Convergence of along-track distance for the second scenario.



(a) First convexification approach **(b)** Second convexification approach

Figure 3.15: Convergence of altitude for the second scenario.



(a) First convexification approach

(b) Second convexification approach

Figure 3.16: Convergence of trajectories for the second scenario.

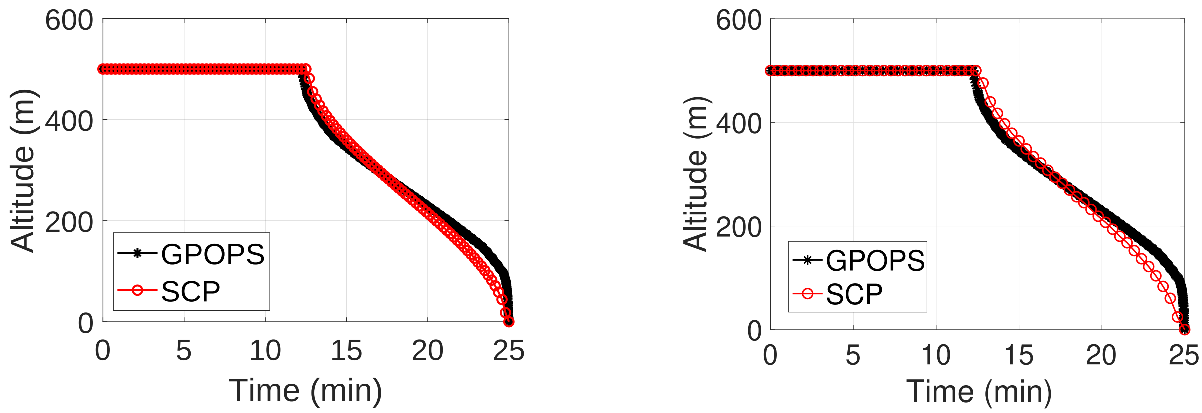
may be caused by the highly coupled state and control variables in the nonconvex problem formulation solved in GPOPS. The convexification process utilized in this research has proved to be helpful to reduce such jitters. Furthermore, since the second convexification method does not involve the constraint relaxation that the first approach requires, it effectively addresses almost the same problem as the GPOPS solver. Consequently, the results from the second convexification approach are nearly identical to those of GPOPS, with their outcomes almost overlapping. Based on this observation, the second convex method will be used for the analysis of trust region, computational time, solution optimality, and robustness as detailed in the following subsections.

In addition, it costs around 20 seconds for GPOPS to solve the second scenario. In contrast, the SCP method only takes five iterations for both approaches to converge and around 1.5 seconds to solve each SOCP subproblem. Compared to the first simulation case, the longer computational time in each iteration may be caused by more discretized nodes used in the second case and a slightly more complicated trajectory involved. Overall, the SCP method shows a faster convergence speed than the general-purpose GPOPS solver, and thus the SCP algorithm has the potential to solve the eVTOL trajectory optimization problem in real time for possible on-board applications.

3.3.3 Analysis of Trust Region

To assess the impact of the trust region size on the convergence of the SCP method, we conducted a series of experiments. The results are presented in Table 3.4, which summarizes the relationship between the trust region size and the corresponding feasibility and the number of iterations required for convergence.

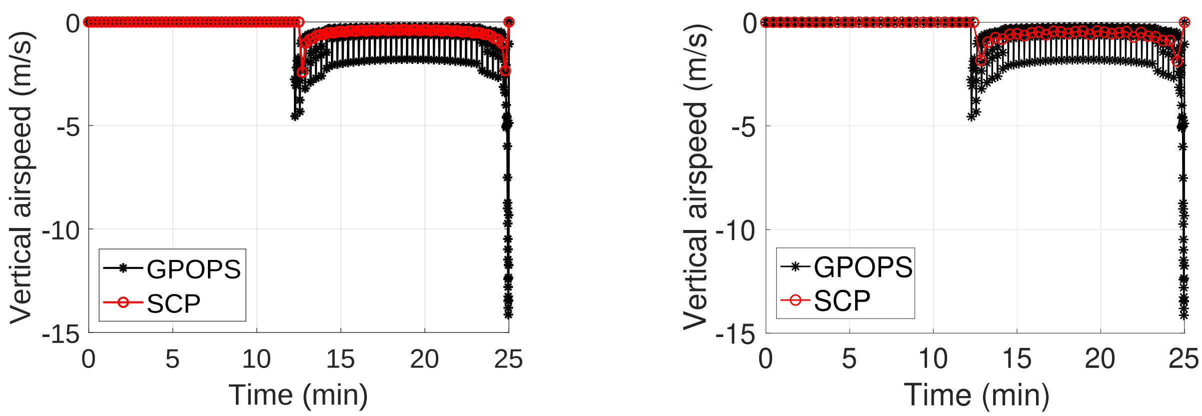
The experimental results presented in Table 3.4 illustrate the significant impact that the size of the trust region has on the convergence behavior of our SCP algorithm. Specifically, a trust region that is too restrictive ($< 5\%$ of the baseline value) may lead to infeasible subproblems. This aligns with the our expectation that after linearization and discretization of the original problem, an overly restrictive trust region can make the convex subproblem infeasible due to an overly confined search space. Conversely, when the trust region is too large ($> 80\%$ of the baseline value), the SCP method struggles to converge, especially



(a) First convexification approach

(b) Second convexification approach

Figure 3.17: Altitude vs. time profiles for the second scenario.



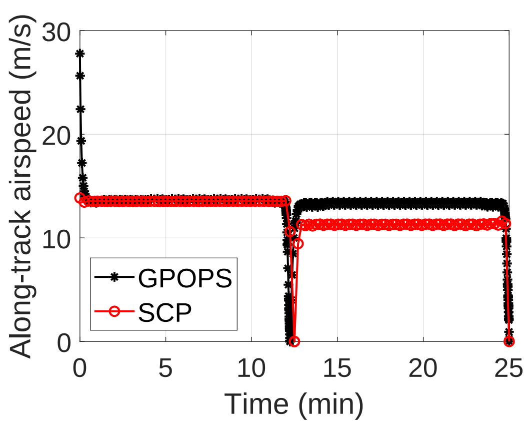
(a) First convexification approach

(b) Second convexification approach

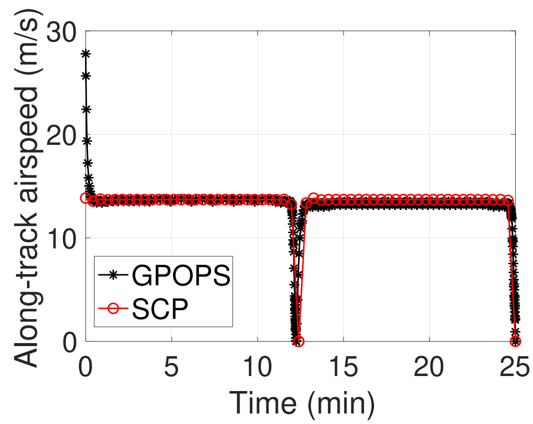
Figure 3.18: Vertical speed vs. time profiles for the second scenario.

Table 3.4: Impact of trust region size on SCP convergence

Trust region size (% of baseline value)	Number of iterations to converge
$< 5\%$	Infeasible subproblems
7% - 10%	10 (may trap in local minima)
10% - 80%	5 - 6 (balanced search space)
$> 80\%$	> 8 (risk of non-convergence)

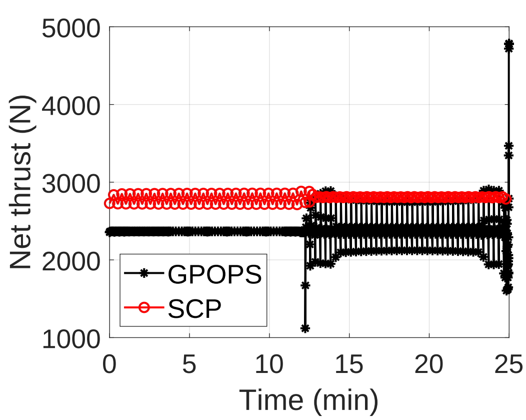


(a) First convexification approach

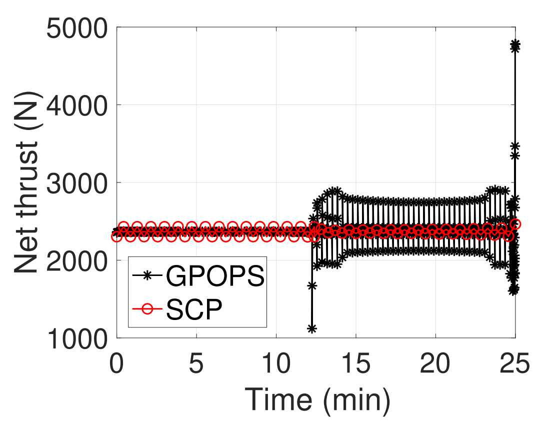


(b) Second convexification approach

Figure 3.19: Along-track speed vs. time profiles for the second scenario.

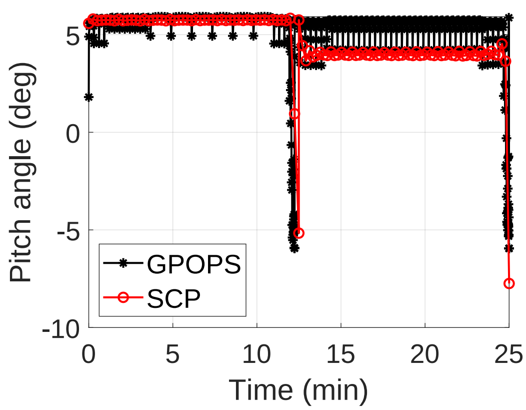


(a) First convexification approach

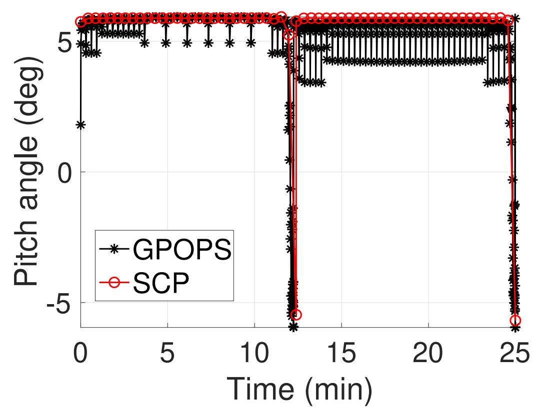


(b) Second convexification approach

Figure 3.20: Net thrust vs. time profiles for the second scenario.



(a) First convexification approach



(b) Second convexification approach

Figure 3.21: Pitch angle vs. time profiles for the second scenario.

under strict convergence criteria. This can be attributed to the search space being too large, suggesting that too large trust region size can lead to inefficient search and potentially divergent behavior. An optimal balance for this problem is achieved with a trust region sized between 10% and 80% of the state value, which results in a balanced search space and requires a moderate number of iterations (five to six) to converge. Interestingly, a small trust region size between 7% and 10% of the state value tends to require more iterations to converge, suggesting that the algorithm may trapped in local minima. These findings demonstrate the importance of selecting an appropriate trust region size to ensure both the feasibility of the SCP method and its efficient convergence.

3.3.4 Analysis of Computational Speed and Optimality

To further investigate the computational efficiency of the GPOPS solver and our SCP method, we implement different convergence criteria and tolerances and compare their results in Table 3.5 to Table 3.8 for both the first and second scenarios. The convergence of the GPOPS solver depends on the maximum mesh error tolerance, which is typically set between 1×10^{-2} (for less accurate solutions) and 1×10^{-5} (for more accurate solutions). In contrast, the SCP method solves a convex problem in each iteration and measures the difference in the solutions between two consecutive iterations. The convergence tolerance for the SCP method is set within the range of 1 to 1×10^{-3} for our applications. Due to these key differences and the nature of the underlying algorithms and techniques utilized for these solvers, it is difficult and nearly impossible to compare these two solvers with the same accuracy ranges and convergence thresholds. Instead, we are interested in studying the variation in the computational efficiency of these solvers when the desired accuracy level and convergence tolerance vary.

To provide a statistical analysis of computational efficiency, we have run 100 simulations for each solver with each tolerance under each CONOP by introducing random Gaussian noise to the initial trajectory in each trial. For each simulation, we calculated the mean, median, and standard deviation of the computational time as shown in Table 3.5 to Table 3.8. From these tables, we can observe that as the convergence tolerance becomes stricter, the computational time of GPOPS exhibits an exponential growth, while the increase in the computational time for the SCP method follows a polynomial trend. For GPOPS, the exponential increase in its computational time with decreasing mesh tolerance can be attributed to the solver's inherent approach, which involves transforming the original optimal control problem into a mesh grid and solving the resulting discretized nonlinear programming problem. As the required solution precision increases (i.e., as the mesh tolerance decreases), the mesh grid becomes finer, leading to a larger-scale discretized problem that requires significantly more computational resource. Notably, if GPOPS fails to find a solution, it can become stuck repeatedly attempting the same sub-step without advancing. This indicates that under tight tolerance settings, GPOPS' performance might degrade exponentially, mainly due to its adaptive mesh refinement strategy that intensifies computation as it seeks finer solutions. In contrast, the SCP method's computational time exhibits a polynomial growth as the convergence tolerance decreases because the SCP method forms a convex approximation of the problem at each iteration and then solves a convex problem in each step. As the convergence tolerance becomes more stringent, the number of iterations required will grow but the cost of each iteration stays roughly the same, leading to a polynomial increase in computational time. It is important to note that these observations are not direct comparisons with the same accuracy or convergence thresholds but rather indicators of the scalability and computational behavior inherent to each method. The results reveal that the SCP is suitable for a wider range of real-time applications because of the polynomial versus exponential growth in computational time relative to GPOPS.

While our SCP method demonstrates an advantage in terms of computational speed in most cases, it is crucial to examine the trade-offs involved, particularly in terms of the accuracy and optimality of the solutions obtained. It is important to note that the objective function is linearized during the SCP process, which might lead to variations in objective values. To ensure consistency in the comparison, we have integrated the control effort over the time domain for both SCP and GPOPS and compared the results. Table 3.9 presents a summary of the objective values obtained using both methods across two different CONOPs.

As indicated in this table, for the first CONOP, the objective value achieved using GPOPS is 8.4083×10^9 , while SCP with the second convexification approach yielded a slightly higher value of 8.4778×10^9. This suggests that while SCP provides a gain in

Table 3.5: Computational time of GPOPS for the first CONOP

Mesh tolerance	Average time	Median time	Standard deviation
1×10^{-2}	3.94 s	3.76 s	0.62 s
1×10^{-3}	16.52 s	14.50 s	4.23 s
1×10^{-4}	38.73 s	35.07 s	9.72 s
1×10^{-5}	98.34 s	94.72 s	14.78 s

Table 3.6: Computational time of SCP for the first CONOP

Convergence tolerance	Average time	Median time	Standard deviation
1	8.01 s	8.63 s	0.51 s
1×10^{-1}	10.29 s	9.93 s	0.64 s
1×10^{-2}	19.69 s	19.83 s	1.17 s
1×10^{-3}	48.82 s	47.74 s	2.26 s

Table 3.7: Computational time of GPOPS for the second CONOP

Mesh tolerance	Average time	Median time	Standard deviation
1×10^{-2}	6.08 s	5.91 s	0.27 s
1×10^{-3}	17.54 s	18.30 s	3.29 s
1×10^{-4}	46.50 s	43.23 s	9.55 s
1×10^{-5}	180.50 s	172.67 s	20.05 s

Table 3.8: Computational time of SCP for the second CONOP

Convergence tolerance	Average time	Median time	Standard deviation
1	7.09 s	7.54 s	1.12 s
1×10^{-1}	14.52 s	15.15 s	3.3 s
1×10^{-2}	19.91 s	19.99 s	2.27 s
1×10^{-3}	36.87 s	35.43 s	2.32 s

computational efficiency, it may do so at the expense of a marginal increase in the objective value, implying a potential loss in solution optimality. However, in the second CONOP, SCP not only demonstrated superior computational speed but also achieved a lower objective value 8.4486×10^9 compared to 8.4819×10^9 for GPOPS. This outcome indicates that SCP can be more efficient and at times more effective, depending on the specific problem characteristics.

It is important to note that both methods belong to the direct collocation method. Therefore, the solutions obtained are not guaranteed to be global minima. The application of SCP requires additional domain knowledge and a deeper mathematical understanding for the convexification process, which can be seen as a trade-off against its computational benefits.

3.3.5 Analysis of Robustness

To provide an in-depth evaluation of the robustness and reliability of the proposed SCP algorithm, Monte Carlo analysis is conducted in this subsection. Each simulation run is performed with a set of perturbed key parameters including vehicle mass, initial altitude, and initial velocity. All these parameters are perturbed with normal distribution and random variation from their baseline values as shown in Table 3.10. Mild variations of $\pm 5\%$ are considered first, and then more substantial $\pm 10\%$ variations in these parameters are introduced to mimic more severe uncertainties and perturbations in real-world operations to test the capability and robustness of the algorithm in handling such discrepancies. Also, 300 simulations are run for each variation, with the expectation of capturing the possible situations considered and providing meaningful statistical results. Given the range of the considered parameters, we believe that 300 runs are sufficient for statistical analysis of the problems considered in this research.

For the cases with $\pm 5\%$ variations, Fig. 3.22 showcases the converged trajectory profiles of the SCP method across 300 simulations, and Fig. 3.23 presents the corresponding velocity profiles from these simulations. The results indicate consistent convergence of SCP within four to six iterations in the vast majority of the trials. For the cases with substantial $\pm 10\%$ variations, both SCP and GPOPS were used to solve 300 runs. However, the GPOPS solver consistently failed to converge under $\pm 10\%$ variations. Even with reduced variations

Table 3.9: Comparison of objective values for both scenarios

CONOP	GPOPS objective value	SCP objective value
First	8.4083×10^9	8.4778×10^9
Second	8.4819×10^9	8.4486×10^9

Table 3.10: Variation of key parameters for robustness analysis of SCP

Parameter	Baseline value	Variation 1	Variation 2
Initial altitude	500 m	$\pm 5\%$	$\pm 10\%$
Initial velocity	13.85 m/s	$\pm 5\%$	$\pm 10\%$
Vehicle mass	240 kg	$\pm 5\%$	$\pm 10\%$

of $\pm 5\%$, a looser convergence tolerance of 1×10^{-3} was necessary for GPOPS to achieve convergence within finite iterations. The objective values of these simulations from both SCP and GPOPS are compared in Fig. 3.24. Each boxplot shows the results of 300 simulation runs. The resulting distributions of objective values reveal that when the GPOPS solver successfully converges, it yields stable objective values. On the other hand, the SCP method consistently achieved lower objective values across the 300 runs, although the distribution of its objective values appears broader compared to that of the GPOPS solver. Furthermore, the simulations with $\pm 10\%$ variations by SCP show more outlier values, which is aligned with our expectations due to wider ranges of parameters considered.

Finally, we conducted a comparative analysis of the computational times between the GPOPS and SCP methods under various parameter variations, and the results are presented in Fig. 3.25. With a looser tolerance and random variations of $\pm 5\%$, GPOPS achieved an average convergence time of 31.5 seconds, while the SCP method consistently converged in less than 10 seconds at $\pm 5\%$ variations and within 12 seconds at the $\pm 10\%$ variation level. Notably, the SCP method displays wider variations in solution times with less than five outliers require more than 20 seconds to converge. These results show that when convergence is achieved, the GPOPS solver exhibits more stable performance across all runs than SCP, while SCP converges faster than GPOPS in general. Overall, the rapid convergence of the SCP method demonstrates not only the efficiency of the algorithm but also its robustness in adapting to a range of uncertain and perturbed conditions. In addition, it is important to mention that we am not trying to prove that SCP is superior than GPOPS on all aspects. In fact. GPOPS is a relatively mature and widely recognized optimal control solver, whereas our SCP method developed in this section is still in its early stage of development. It is our expectation that the performance of the SCP method will be enhanced when more advanced techniques, such as more efficient discretization rules, adaptive meshing, and sparsity, are introduced.

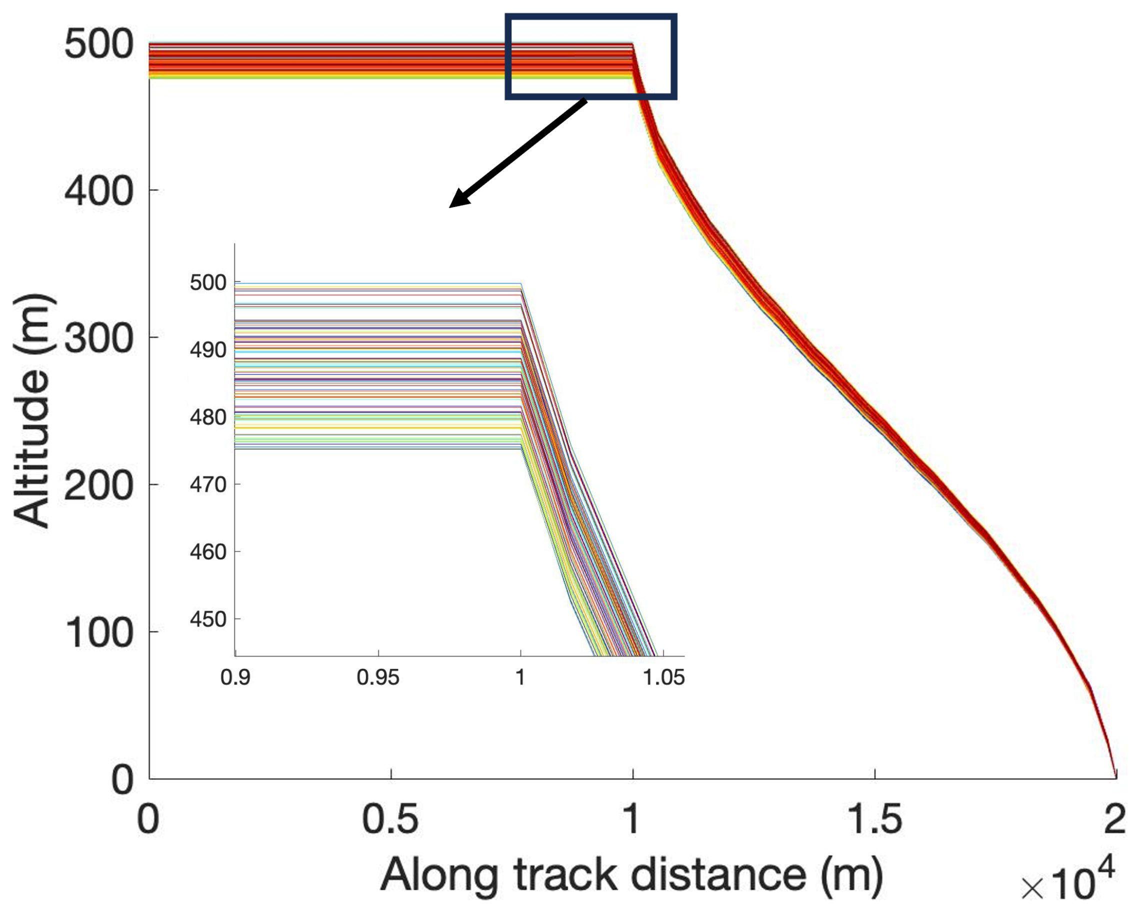


Figure 3.22: SCP trajectory profiles from different initial positions with ±5% variations.

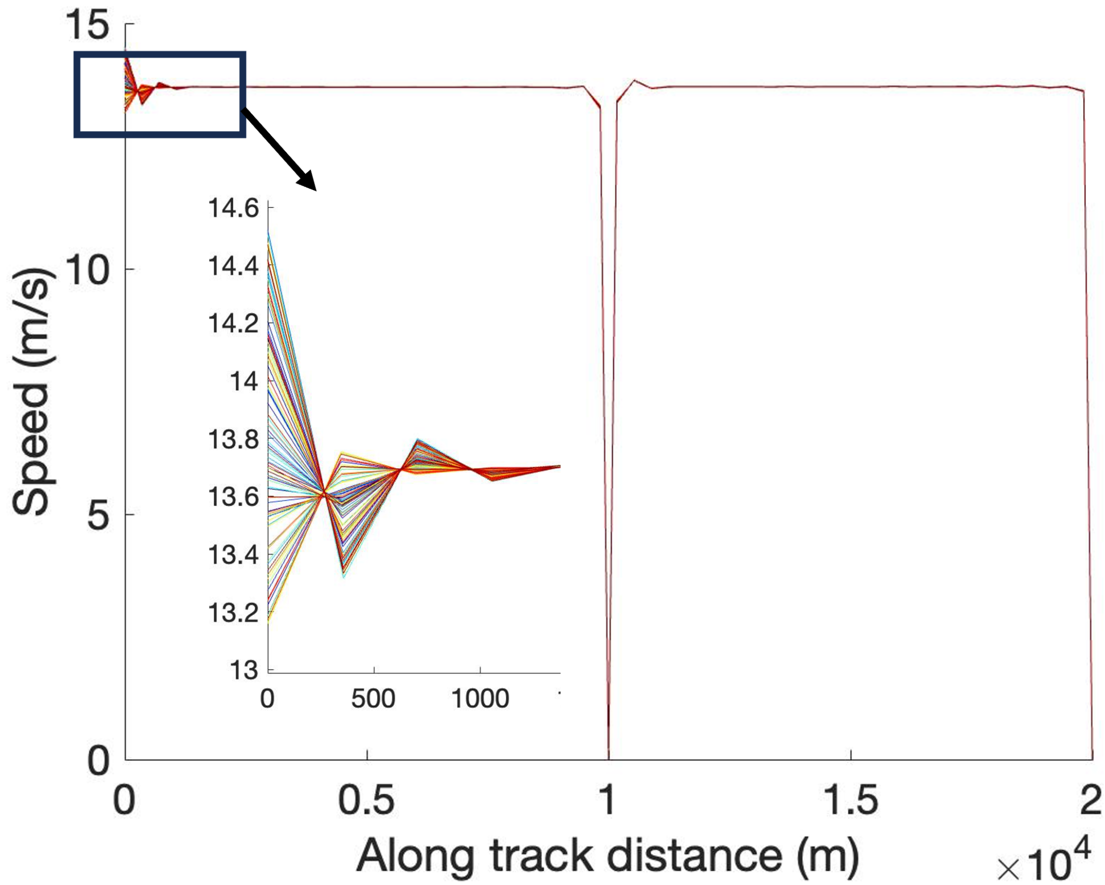


Figure 3.23: SCP velocity profiles from different initial velocities with ±5% variations.

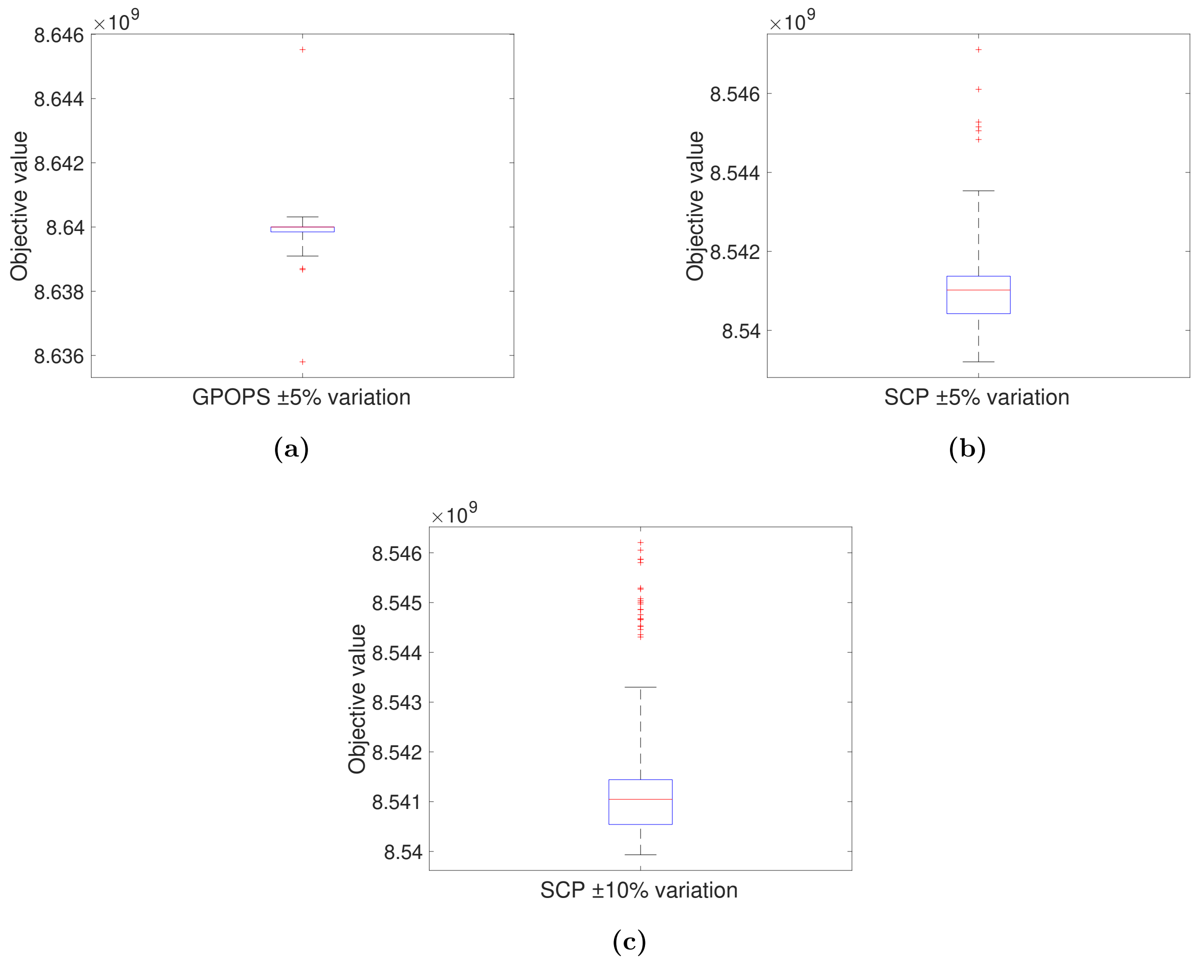
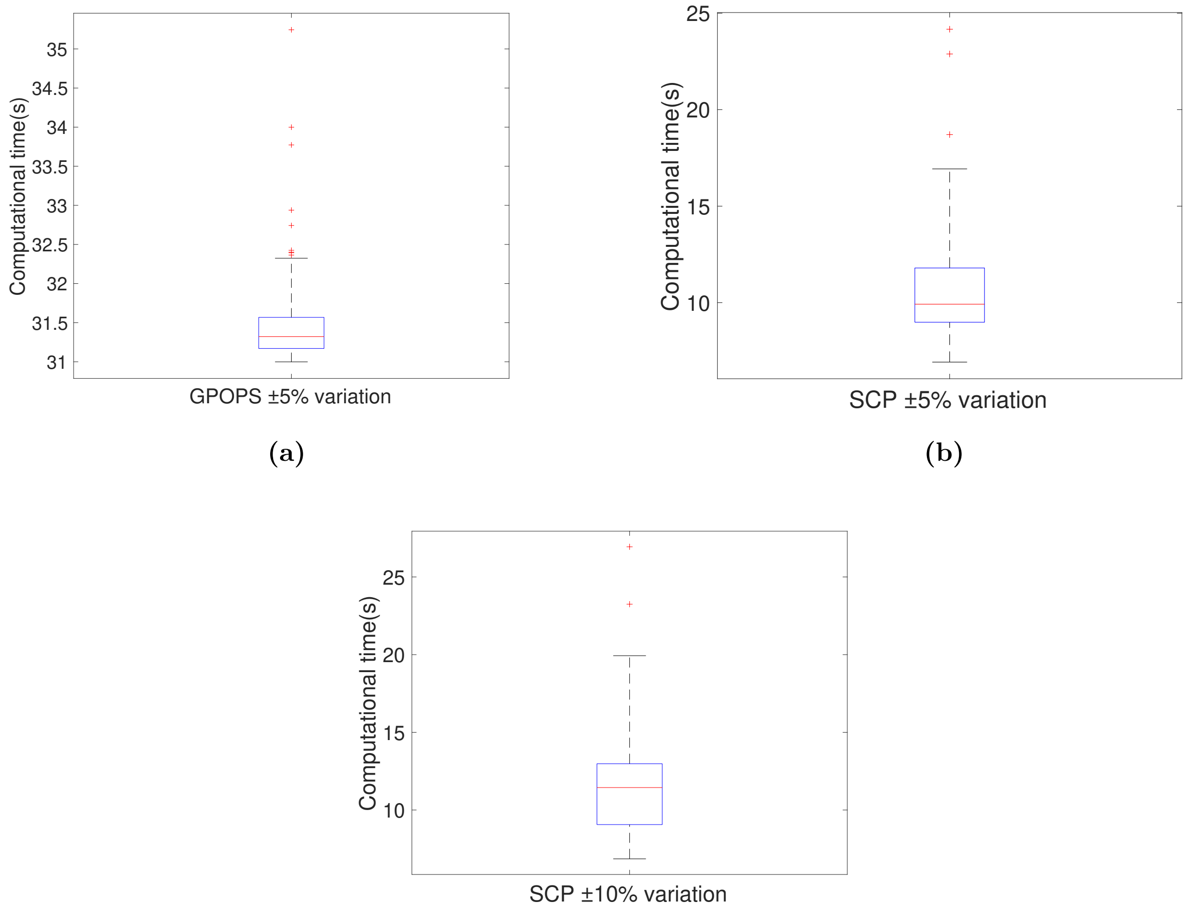


Figure 3.24: Boxplot comparison of objective values for SCP and GPOPS.



(a) **(b)** **(c)**

Figure 3.25: Boxplot comparison of computational times for SCP and GPOPS.

3.4 Summary

Real-world UAM missions encompass multiple flight phases, including take-off, climb, cruise, descent, and landing. These missions require onboard computers to efficiently plan feasible or even optimal trajectories in real-time within complex urban environments. In this research, we presented a successive convex approach to solve the multi-phase trajectory optimization problem for eVTOL vehicles from cruise to descend and land on the destination vertiport under different scenarios. To solve this nonconvex optimal control problem, we first decoupled the states and controls and reduced the nonlinearity in the flight dynamics through change of variables. Then, we transformed the original nonconvex problem into an SOCP problem via a convex relaxation of the control constraint and successive linear approximations of the nonlinear objective function and nonlinear dynamics. In view of possible violation of the underlying assumptions for exact relaxation, an alternative convexification method is developed by eliminating the need of control constraint relaxation. Finally, an SCP algorithm is developed to find a solution to the original trajectory optimization by solving a sequence of SOCP subproblems. Numerical simulations of two different scenarios are performed to test the performance of our proposed SCP algorithm through comparisons with the state-of-the-art GPOPS solver.

The results revealed that the SCP method exhibited a polynomial growth in computational time versus an exponential increase in computational time shown by GPOPS when the convergence tolerance becomes smaller. For scenarios requiring lower accuracy, GPOPS may perform better and offer sufficient performance with less computational cost than SCP. As the demand for higher accuracy increases, however, SCP may become more advantageous because of its polynomial versus exponential growth in computational time relative to GPOPS, demonstrating the suitability of SCP for a wider range of real-time applications than GPOPS.

Chapter 4

Aim II: Develop a convex approach to high-fidelity landing trajectory optimization of eVTOL vehicles

Among the flight phases of eVTOLs, the landing stage is inherently intricate due to the dynamic, disturbing weather conditions (e.g., wind gusts) and the complex infrastructure (e.g., high-rise buildings) within urban landscapes. Most existing research in this field utilizes reinforcement learning approaches [57, 18, 19, 20] or various optimization methods [59, 85, 47, 90, 87] to solve the eVTOL landing problems. These approaches typically employ simplified models of vehicle dynamics, which, while effective in control designs, often fail to accurately capture aerodynamic interactions in real-world urban environments for precision decision-makings. These interactions include wind gusts and the effects of turbulence caused by the urban infrastructure, which can significantly influence the behavior of eVTOL vehicles [44, 51].

Since the existing research largely relies on simplified dynamic models ignoring the deviations in aircraft behavior that can occur under real-world urban conditions, it can lead to landing paths that are not only suboptimal but also potentially unsafe. In contrast, high-fidelity models that account for aerodynamic effects could potentially enable more accurate and realistic landing trajectory solutions. However, solving the resulting complicated problem can be computationally expensive [86, 88]. Efficiently addressing the high-fidelity landing problems is of paramount importance for safe and precision UAM operations, marking it an attractive area of research in recent years in this field.

In this chapter, we propose an innovative approach to this challenge. Specifically, we employ a novel convex optimization framework and a sequential convex programming (SCP) algorithm to solve a sophisticated eVTOL landing problem that couples high-fidelity aerodynamics with nominal flight dynamics. Our approach addresses a 6 degrees-of-freedom (6-DoF) eVTOL landing trajectory optimization problem incorporating an ordinary differential equation (ODE)-based aerodynamic model. To develop the convex approach, we begin with the formulation of a general, highly nonconvex optimal control problem that seamlessly integrates the ODE-based aerodynamic model with the 6-DoF flight dynamics and various flight constraints. This integration is crucial for capturing the complex interactions between the vehicle and its operating environment. We then isolate the nonlinear components of the dynamic equations of motion from the original formulation. This separation is a critical step in managing the complexity of the problem. Subsequently, we employ the first-order Taylor series expansion to approximate these nonlinear components, thereby transforming the problem into a convex form. The potential artificial infeasibility issues due to the linearization are handled by virtual controls. The final step involves discretization of the problem using the standard trapezoidal discretization technique. By solving a set of relaxed convex sup-problems using mature convex optimization solvers, we obtained the solution to the landing trajectory optimization problem and compare the results with those from other existing solvers.

4.1 Problem Formulation

In this study, the ODE-based aerodynamic rotor models are leveraged and integrated with the traditional 6-DoF eVTOL vehicle flight dynamics model and together formulate an optimal control landing problem. The first subsection introduces the calculation of the thrust coefficient of each rotor. In the second subsection, we present the rotor inflow model used in this study. In the third subsection, we formulate the ODE-based model for the rotor dynamics. In the fourth and fifth subsections, we present the 6-DoF eVTOL flight dynamic model and the essential flight constraints. Finally, we formulate the overall optimal control problem for eVTOL landing in the last subsection.

4.1.1 Thrust Coefficient of Single Rotor

In this section, the blade element momentum theory (BEMT) is exploited to calculate the lift and thrust of a single rotor blade. The BEMT model calculates the total lift and drag forces for the rotor blade by dividing the blade into n small elements and then summing up the lift and drag forces obtained from every single element [37].

As shown in Figure 1 and Figure 2, the location of the blade element is defined by the radial and azimuthal coordinates, (r, φ). The freestream velocity is composed of both the tangential component, u, and the perpendicular component, w. Then, the advanced ratio is obtained by $\mu_x = u/(\Omega R)$ and the inflow ratio $\mu_z = w/(\Omega R)$. According to the conservation of momentum, the induced flow will be created when generating thrust, resulting in an inflow distribution $\lambda_i(r, \varphi)$.

For blade element at location (r, φ), the local total freestream speed is defined as:

$$\frac{u_\infty}{\Omega R} = \sqrt{(r/R + \mu_x \sin\varphi)^2 + \lambda_i(r,\varphi)^2} \equiv \bar{u}_\infty(r,\varphi) \tag{4.1}$$

and the local lift and drag are:

$$dL = \frac{1}{2}\rho u_\infty^2 c_l(\vartheta, r) c dr, dD = \frac{1}{2}\rho u_\infty^2 c_d(\vartheta, r) c dr \tag{4.2}$$

where the effective angle of attack is:

$$\vartheta = \alpha - \tan^{-1}\frac{\lambda_i}{\mu_x} \equiv \alpha - k \tag{4.3}$$

The loads in the rotor coordinate system at each blade element can therefore be calculated as:

$$dF_z = dL\cos k - dD\sin k \tag{4.4}$$

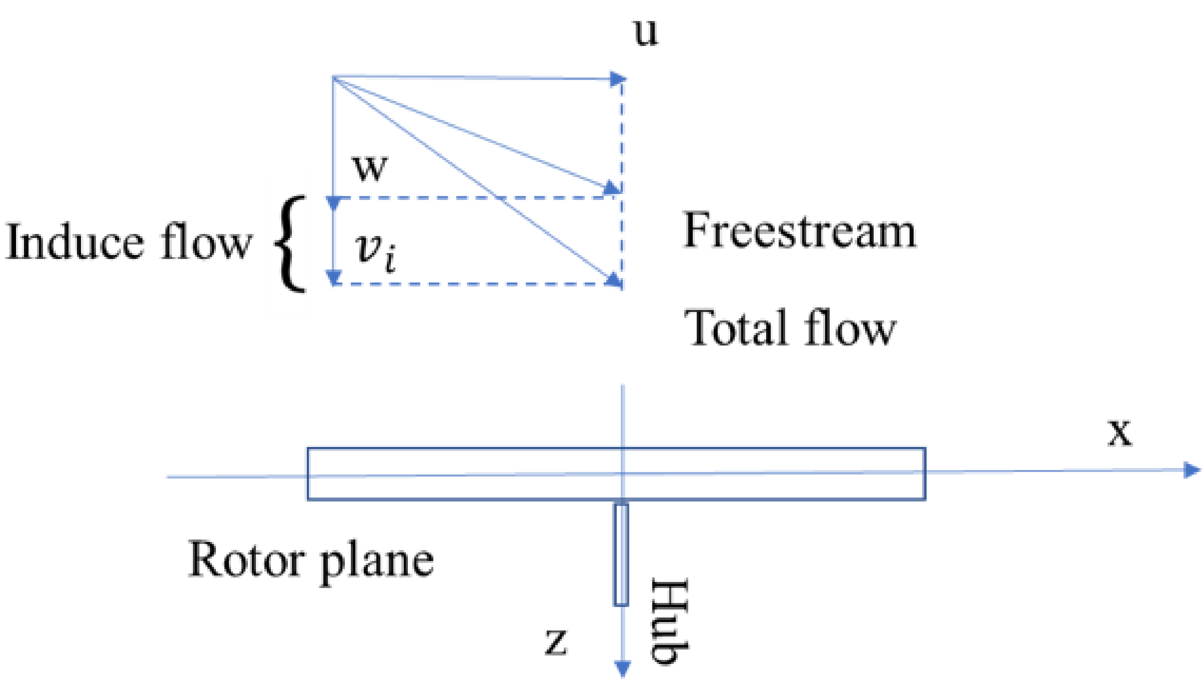


Figure 4.1: BEMT for single rotor side view

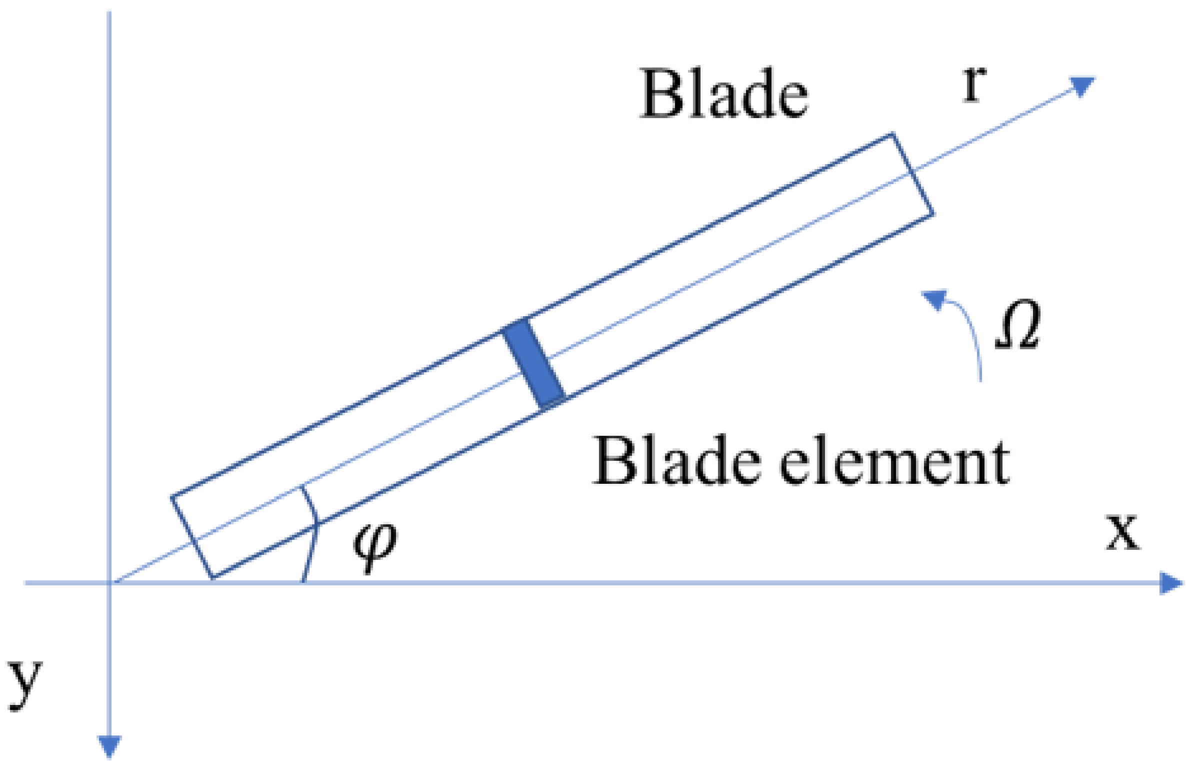


Figure 4.2: BEMT for single rotor top view

$$dF_x = -dL\sin k + dD\cos k \tag{4.5}$$

In this study, We assume that the drag term $dD\sin k$ is small when compared to the lift term. Also, we assume constant installation angle, constant chord length, and constant airfoil with $c_l(\vartheta, \bar{r}) = 2\pi\vartheta$. The thrust coefficient can be calculated as:

$$C_t(\mu_x, \mu_z, \lambda_i) = \sigma \int_0^{2\pi} \int_0^{1} [\bar{u}_\infty^2 (k-\alpha) cosk] \, d\bar{r} d\varphi \tag{4.6}$$

4.1.2 Rotor Inflow Model

In this study, the rotor inflow model is adapted from Cao [15] and was developed by Pitt and Peters [56] using a system of linear ODEs to describe the dynamic inflow of a rotor disk. The model relates the dimensionless aerodynamic loading to dimensionless induced flow distribution, which is described by the equation below:

$$\lambda(\bar{r}, \psi, t) = \lambda_0(t) + \lambda_s(t)\bar{r}\sin\psi_r + \lambda_c(t)\bar{r}\cos\psi_r \tag{4.7}$$

where $\bar{r}$ is r/R, and R is the rotor radius. λ_0, λ_s, λ_c are the uniform, side-to-side, and fore-to-aft components of the induced airflow, λ is the local dimensionless induced airflow velocity defined by:

$$\lambda = u_x/\Omega R \tag{4.8}$$

where u_x is the induced airflow velocity, and Ω is the angular velocity of the propeller. Then, we can write the linear first-order state-space representation of the inflow dynamics as follows:

$$\begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} = -M^{-1}L^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} + M^{-1} \begin{bmatrix} C_T \\ C_{roll} \\ C_{pitch} \end{bmatrix} \tag{4.9}$$

In the above equation, α is defined as the angle of incidence, and $\bar{V}_\infty$ represents the dimensionless free-stream velocity that can be obtained from:

$$\bar{V}_\infty = \frac{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}{2\pi n R}, \alpha = \pi/2 - \cos^{-1}(\frac{[\dot{x}, \dot{y}, \dot{z}] \cdot \mathbf{n}_{rotordisk}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}) \tag{4.10}$$

where n in the first equation represents the propeller rotational speed, and $\mathbf{n}$ in the second equation stands for the normal vector of the rotor disk.

Equation 4.9 calculates the inflow dynamics for a single rotor. In the next subsection, we will discuss the state-space model composed of four motors inflow aerodynamic system.

4.1.3 ODE Model of Rotor Dynamics

Following the inflow dynamic model for the rotor disk, we can construct the state-space model for the 4 rotors to calculate the thrust, torques, and also dynamic inflow states for each motor. The output of this system will feed into the vehicle dynamic model that will be discussed in the next subsection. In this study, we consider the states of the system including the inflow states of each rotor along with the total vertical thrust F_z, the sum of the moment respect to x-, y-, and z-axes, which are denoted as τ_x, τ_y, τ_z, together form the state vector $\mathbf{x}_0 \in \mathbb{R}^{16\times 1}$:

$$\mathbf{x}_0 = [\lambda_{0,r1}, \lambda_{s,r1}, \lambda_{c,r1}, \ldots, \lambda_{0,r4}, \lambda_{s,r4}, \lambda_{c,r4}, F_z, \tau_x, \tau_y, \tau_z]^T \tag{4.11}$$

where $\lambda_{0,ri}, \lambda_{s,ri}, \lambda_{c,ri}$ are the inflow states of i-the rotor. The control vector $\mathbf{u}_0 \in \mathbb{R}^{4\times 1}$ is considered to be the square of each motor's rotational speed:

$$\mathbf{u}_0 = [n_1^2, n_2^2, n_3^2, n_4^2]^T \tag{4.12}$$

Consequently, we can formulate the state-space form of the aerodynamic system as follows:

$$\dot{x}_0 = A_0 \cdot \mathbf{x}_0 + B_0 \cdot \mathbf{u}_0 \tag{4.13}$$

where A_0 and B_0 matrices are respectively defined as:

$$A_0 = \begin{bmatrix} -M^{-1}L^{-1} & & & & \\ & -M^{-1}L^{-1} & & & \\ & & -M^{-1}L^{-1} & & \\ & & & -M^{-1}L^{-1} & \\ & & & & -I^{4x4} \end{bmatrix} \tag{4.14}$$

and

$$B_0 = \begin{bmatrix} 0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1} \\ b_1 & b_2 & b_3 & b_4 \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} & -(b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) & -(b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) \\ k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \end{bmatrix} \tag{4.15}$$

where $b_1 = b_2 = b_3 = b_4 = C_T \rho D^4$, $k_{\text{roll}} = C_{\text{roll}} \rho D^5$, $k_{\text{pitch}} = C_{\text{pitch}} \rho D^5$, and $k_{\text{yaw}} = C_Q \rho D^5$.

The above aerodynamic model will be combined with the flight dynamics to formulate the landing trajectory optimization problem.

4.1.4 Flight Dynamics

In this subsection, we present the flight dynamics of the eVTOL vehicle. In this study, we consider a 6-DoF flight dynamic model, and the state is defined as:

$$\mathbf{X} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^T \in \mathbb{R}^{12} \tag{4.16}$$

where x, y, z represent the displacement of the mass center of the vehicle in the Earth reference frame (North, East and Downwards); the rotational displacement components are defined as ψ, θ, ϕ; $\dot{x}, \dot{y}, \dot{z}$ represent the velocity in the inertial frame; and p, q, r define the relative rotational rates to the body-fixed frame. The control input is defined as:

$$\mathbf{U} = [F_z \quad \tau_x \quad \tau_y \quad \tau_z]^T \tag{4.17}$$

After considering the assumptions made above in this research, the model in the inertial frame is simplified as:

$$\begin{aligned}
\ddot{x} &= -\frac{F_z}{m}[\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)] \\
\ddot{y} &= \frac{F_z}{m}[\cos(\psi)\sin(\phi) - \cos(\phi)\sin(\psi)\sin(\theta)] \\
\ddot{z} &= g - \frac{F_z}{m}[\cos(\phi)\cos(\theta)] \\
\ddot{\phi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_x}{I_x} \\
\ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_y}{I_y} \\
\ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{\tau_z}{I_z}
\end{aligned} \tag{4.18}$$

The system can rewrite in state space form as:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = \mathbf{f}(\mathbf{X}) + B(\mathbf{X})\mathbf{U} \tag{4.19}$$

where

$$\mathbf{f}(\mathbf{X}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ 0 \\ 0 \\ g \\ C_{Ix}qr \\ C_{Iy}pr \\ C_{Iz}pq \end{bmatrix} \tag{4.20}$$

which $C_{Ix} = \frac{I_y - I_z}{I_x}$, $C_{Iy} = \frac{I_z - I_x}{I_y}$, and $C_{Iz} = \frac{I_x - I_y}{I_z}$.

4.1.5 Flight Constraints

In this research, we consider a landing problem with a fixed time of flight. To ensure the safety of the flight mission, the vehicle must satisfy some constraints.

First, we impose the initial and terminal conditions for the aircraft by introducing the following boundary conditions:

$$\mathbf{X}(t_0) = \mathbf{X}_0 \tag{4.21}$$

$$\mathbf{X}(t_f) = \mathbf{X}_f \tag{4.22}$$

Then, we define the maximum distance to the landing pad and maximum altitude as:

$$0 \leq x \leq x_{max} \tag{4.23}$$

$$0 \leq y \leq y_{max} \tag{4.24}$$

$$0 \leq z \leq z_{max} \tag{4.25}$$

Meanwhile, the maximum rotor speed is restricted based on vehicle specifications:

$$\Omega_{min} \leq \Omega_i(t) \leq \Omega_{max} \tag{4.26}$$

where the subscript i denotes the motor number, and in the model considered in this research, the system have four motors.

4.1.6 Performance Index and Optimal Control Problem

In this research, the optimal control problem is formulated to minimize the energy cost:

$$J = \int_{t_0}^{t_f} \sum_{i=1}^{4} \Omega_i^2(t)dt \tag{4.27}$$

where Ω_i^2 denotes the squared rotor speed for each rotor. Combining the objective function, aerodynamics, flight dynamics, and constraints, the optimal control problem is formulated as:

Problem 1:

$$\text{Minimize:} \quad (4.27)$$

$$\text{Subject to:} \ (4.13), (4.19), (4.21), (4.22), (4.23), (4.24), (4.25), (4.26)$$

Overall, we want to minimize the objective function (4.27) while satisfying the rotor aerodynamics (4.13), the flight dynamics (4.19), the boundary conditions (4.21) and (4.22), and the inequality state and control constraints (4.23)-(4.26).

Problem 1 is not convex and difficult to solve. In the next section we will discuss the convexification process of the problem, and then develop a sequential convex optimization approach to solve the problem iteratively.

4.2 Convex Approach

In eVTOL systems, the integration of standard flight dynamics with intricate aerodynamic equations presents a unique challenge. These systems do not naturally forms a convex pproblem due to their inherent nonlinearity and the multidimensional nature of the dynamics involved. Transforming these high-fidelity problems into a convex format often requires a lot of effort, such as the introduction of slack variables to manage nonconvex constraints and the utilization of first-order Taylor series approximations to linearize highly nonlinear dynamics. In this research, we introduced a novel convexification approach that separate the linear part of the dynamic system from the overall system and applied first-order Taylor series approximations for the remaining terms. The detailed process is explained in the following subsections.

4.2.1 Convexification

For the eVTOL landing problem, we first rewrite the overall state space system as:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = A\mathbf{X} + B(\mathbf{X})\mathbf{U} + \mathbf{g}^* \tag{4.28}$$

where A is the Jacobian matrix, and $\mathbf{g}^*$ represents the constant gravity vector, which are as follows:

$$
A = \begin{pmatrix}
-M^{-1}L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & -M^{-1}L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{l}{I_x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Ix}r & C_{Ix}q & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{l}{I_y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iy}r & 0 & C_{Iy}p & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{l}{z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iz}q & C_{Iz}p & 0 & &
\end{pmatrix} \tag{4.29}
$$

$$
\mathbf{g}^* = \begin{bmatrix} 0^{1x24} & -g & 0 & 0 & 0 \end{bmatrix}^T \tag{4.30}
$$

where

$$
\begin{aligned}
g_1 &= -\frac{1}{m}[\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)] \\
g_2 &= \frac{1}{m}[\cos(\psi)\sin(\phi) - \cos(\phi)\sin(\psi)\sin(\theta)] \\
g_3 &= -\frac{1}{m}[\cos(\phi)\cos(\theta)]
\end{aligned} \tag{4.31}
$$

The constants in the matrix A are defines as: $C_{Ix} = \frac{I_y - I_z}{I_x}$, $C_{Iy} = \frac{I_z - I_x}{I_y}$, $C_{Iz} = \frac{I_x - I_y}{I_z}$, and the control matrix is defined as:

$$
B = \begin{bmatrix}
0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1} \\
b_1 & b_2 & b_3 & b_4 \\
b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} & -(b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} \\
b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) & -(b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) \\
k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \\
0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1}
\end{bmatrix} \tag{4.32}
$$

To convexify the system, we further write the above equation as:

$$
\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = A_L \mathbf{X} + \mathbf{g}^* + B\mathbf{U} + \mathbf{f}_{NL}(\mathbf{X}, t) \tag{4.33}
$$

where A_L is the linear part of the A matrix that can be written as:

$$
A_L = \begin{bmatrix}
-M^{-1}L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -M^{-1}L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & l/I_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & l/I_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \tag{4.34}
$$

and $\mathbf{f}_{NL}$ is defined as:

$$\mathbf{f}_{NL}(X,t) = \begin{bmatrix} 0^{22\times 1} \\ F_z g_1 \\ F_z g_2 \\ F_z g_3 \\ C_{Ix} qr \\ C_{Iy} pr \\ C_{Iz} pq \end{bmatrix} \tag{4.35}$$

To linearize the nonlinear part of the function, we applied first-order linearization method and obtain:

$$\mathbf{f}_{NL}(\mathbf{X},t) \approx \mathbf{f}_{NL}\left(\mathbf{X}^*,t\right) + \frac{\partial f_{NL}}{\partial X}\left(\mathbf{X}^*,t\right)\left(\mathbf{X}-\mathbf{X}^*\right) \tag{4.36}$$

After substituting to original equation, the full equation can be write as:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X},\mathbf{U},t) \approx A_L\mathbf{X} + g^* + B\mathbf{U} + \mathbf{f}_{NL}\left(\mathbf{X}^*,t\right) + \frac{\partial f_{NL}}{\partial X}\left(\mathbf{X}^*,t\right)\left(\mathbf{X}-\mathbf{X}^*\right) \tag{4.37}$$

where the partial derivative matrices $\frac{\partial f_{NL}}{\partial X}$ are shown below:

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g1^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_1}{\partial \phi} & \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g2^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_2}{\partial \phi} & \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g3^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_3}{\partial \phi} & \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Ix} r & C_{Ix} q \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iy} r & 0 & C_{Iy} p \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iz} q & C_{Iz} p & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
\frac{\partial g_1}{\partial \phi} & \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial \psi} \\
\frac{\partial g_2}{\partial \phi} & \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial \psi} \\
\frac{\partial g_3}{\partial \phi} & \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial \psi}
\end{bmatrix}
= \frac{F_z}{m}
\begin{bmatrix}
-\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & -\cos\psi\cos\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\
\cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\sin\psi\cos\theta\cos\phi & -\sin\psi\sin\phi - \cos\psi\sin\theta\cos\phi \\
\cos\theta\sin\phi & \sin\theta\cos\phi & 0
\end{bmatrix}
\tag{4.38}
$$

To develop the SCP method, the last step is to discretize the problem. In this work, we use the trapezoidal approach for discretization. The trapezoidal method is defined as:

$$
x_i = x_{i-1} + \frac{\Delta t}{2}\left(\dot{x}_{i-1} + \dot{x}_i\right) \tag{4.39}
$$

For our problem, after rearranging all the terms, we obtain the following discretized state equation:

$$\left[\left(I - \frac{\Delta t}{2}(A_L^{k-1} + A_{NL_i}^{k-1})\right)\mathbf{X}_i\right] - \left[\left(I + \frac{\Delta t}{2}(A_L^{k-1} + A_{NL_{i-1}}^{k-1})\right)\mathbf{X}_{i-1}\right] - \frac{\Delta t}{2}B\mathbf{U}_{i-1} - \frac{\Delta t}{2}B\mathbf{U}_i$$
$$= \frac{\Delta t}{2}\left[\left(f_{NLi-1}^{k-1} - A_{NL_{i-1}}^{k-1}X_{i-1}^{k-1} + g^*\right) + \left(f_{NL_i}^{k-1} - A_{NL_i}^{k-1}x_i^{k-1} + g^*\right)\right] \tag{4.40}$$

Finally, we formulate the convex form of the landing problem as:

Problem 2:

$$\text{Minimize:} \quad (4.27)$$
$$\text{Subject to:} \ (4.13), (4.21) - (4.27), (4.36)$$

Now, Problem 2 is in a convex form and is ready for the application of the SCP algorithm to compute the optimal solution successively.

4.2.2 Virtual Control

During the process of convexification, one significant challenge is the phenomenon of artificial infeasibility. This issue typically arises during the initial stages of the optimization process, often due to an inadequate initial guess. Artificial infeasibility occurs when there is discrepancy between the linearized model and the true nonlinear original system dynamics, resulting in an optimization problem that, in its current linearized form, has no feasible solution.

To mitigate the problem, we introduce the virtual control vector $\boldsymbol{v}$ to the system dynamics. Virtual control acts as a corrective mechanism in the optimization process, helping to fix artificial infeasibility. The key idea behind virtual control is to introduce additional degrees of freedom into the system dynamics, allowing the optimization algorithm to have greater flexibility in finding feasible solutions, especially during the initial iterations where the solution space may not be well-defined. Vector $\boldsymbol{v}$ usually has the same length as the state vector. With the virtual control the state equation will become:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) \approx A_L\mathbf{X} + \tilde{B}\mathbf{U} + \mathbf{g}^* + \mathbf{f}_{NL}(\mathbf{X}^*, t) + \frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{X}}(\mathbf{X}^*, t)(\mathbf{X} - \mathbf{X}^*) + E\boldsymbol{v} \tag{4.41}$$

where E is an identity matrix. The primary function of E in the problem is to help the states of the system in reaching a feasible region within a finite time frame. By multiplying with the virtual control vector, E ensures that each state variable can be individually adjusted.

It is important to note that the virtual control is not a physical control input but a variable to improve the convergence of the optimization algorithm. As the algorithm progresses and the solution converges, the influence of the virtual control diminishes, eventually becoming negligible as a feasible and optimal solution is approached. Since the virtual control term is a auxiliary control variable that only acts to prevent the infeasibility raised during the algorithm's convergence steps, it will be heavily penalized in the objective function. The new objective function is thus formed as:

$$J = \int_{t_0}^{t_f} \sum_{i=1}^{4} \Omega_i^2(t)dt + w_v \cdot \text{ess sup}\|E(t)\boldsymbol{v}(t)\|_1 \tag{4.42}$$

where w_v is a large penalty coefficient. The essential supremum, denoted as ess sup, is the smallest value such that $\|E(t)\boldsymbol{v}(t)\|_1$ is less than or equal to it for almost all $t \in [t_0, t_f]$, excluding any set of measure zero ($\{t \in [t_0, t_f] \mid \|E(t)\boldsymbol{v}(t)\|_1 > \text{ess sup}\}$ has measure zero).

4.2.3 Sequential Convex Programming

In this research, for the first time, an SCP method that solves a series of local convex optimization problems defined in Problem 2, is developed to obtain approximate solutions to the original nonconvex high-fidelity eVTOL vehicle landing problem. The SCP method used in this research consists of the following steps:

1. Initialize the iteration index $k = 0$ and define the initial states $\mathbf{X}(t_0)$. Then, insert the initial conditions to the equations of motion to obtain an initial trajectory $\mathbf{X}^{(0)}$. Set $k = k + 1$.

2. For $k > 0$, parameterize a convex subproblem, Problem 2, using the solution from the previous iteration and solve this subproblem to find a solution pair $[\mathbf{X}^{(k)}, \mathbf{U}^{(k)}]$at the current step.

3. Check the convergence condition

$$\sup |\mathbf{X}_i^{(k)} - \mathbf{X}_i^{(k-1)}| \le \varepsilon \tag{4.43}$$

   where ε is a preset tolerance. If the condition is satisfied, the algorithm moves to step 4; otherwise set $k = k + 1$ and go back to step 2.

4. The algorithm is converged and a solution for the problem is found to be $\{\mathbf{X}^{(k)}, \mathbf{U}^{(k)}\}$.

4.3 Numerical Simulations

In this research, a small quadrotor serves as the example model for simulations that emulate the UAM landing scenario. The vehicle parameters, which include mass, rotor configuration, and aerodynamic coefficients, are essential for accurately modeling the dynamics of the quadrotor. These parameters are detailed in Table 4.1.

The simulations started from specified initial conditions and aim to reach defined terminal conditions, representing the start and desired end states of the quadrotor during a landing maneuver. These conditions are outlined in Table 4.2. To initialize the SCP algorithm, an initial trajectory is generated using linear interpolation between the initial and final states. This approach provides a starting point sufficiently close to the desired trajectory, enabling the iterative process of refinement to begin effectively.

Initially, we conducted a simulation on the quadrotor modeled with 16 states, not considering the specific rotor dynamics. This phase focuses on the basic flight dynamics and control responses under nominal conditions, providing a baseline understanding of the vehicle's behavior in a simplified scenario and the results are shown in subsection 4.3.1. After establishing a baseline with the initial simulation, we progressed to more complex simulations that incorporate detailed rotor dynamics and higher fidelity models. Specifically, we conducted a comparative simulation between using the 28-state SCP method and using the 28-state GPOPs solver and the results are demonstrated in subsection 4.3.2.

4.3.1 Simulation with 16 states

To benchmark our results and demonstrate the efficacy of our approach, we first solved the most fundamental landing problem with only 12 states that excludes aerodynamic factors, serving as a baseline for our comparative analysis. Then, the 16 states optimization problem is solved. Both the 12-state and 16-state models have been solved using the GPOPS solver, a well-established tool in optimal control [55]. The results obtained from our SCP-based method and the GPOPS solutions are presented and compared. The following subsections detail the results of these simulations, demonstrating the performance of our proposed method in achieving efficient, safe, and practical eVTOL landing trajectories. All simulations are carried out on a laptop with an macOS 64-bit operating system and M1 Pro Processor.

The simulated landing task begins with the vehicle positioned five meters above the vertiport, at a distance of three meters in the x-direction and two meters in the y-direction from the designated landing point. Figure 4.3 below depicts the optimal landing trajectory achieved using the SCP method. Notably, the trajectory shows a smooth landing pattern, guiding the vehicle from its initial state to the final landing condition efficiently and safely.

Figure 4.4 illustrates the efficiency of the SCP method in terms of computational time. Remarkably, the SCP algorithm achieved convergence in just six iterations, with the entire process taking less than three seconds in total. Each iteration was notably fast, averaging around 0.5 seconds. This rapid convergence demonstrates the potential of the SCP method for real-time solution capability for complex dynamic systems. Figure 4.5 presents the convergence process for the objective function, where we can see that the SCP method converges to a stable objective value. Figures 4.6 to 4.8 display the convergence profiles for the x, y, and z components of the vehicle position, respectively. These plots reveal that the solver met the convergence criteria in six iterations.

Figure 4.9 shows the trajectories from GPOPS and the SCP algorithm. It is clear to see that all the three trajectories are similar to each other, and the SCP method indeed finds a solution near the optimal result. Figure 4.10 compares the corresponding control profiles for the 16-state model. The rotor speed curves from the two solvers show very similar trend as expected. In figure 4.11, we present a side-by-side comparison of the thrust curves derived

Table 4.1: Vehicle parameters for simulations

Parameter	Value
Vehicle's mass, m	0.69 kg
Propeller diameter, D	0.1524 m
Atmospheric density, ρ	1.225 kg/m^3
Gravitational acceleration, g	9.81 m/s^2
Moment of inertia about body frame's x-axis, I_x	4.69×10^{-2} kg·m^2
Moment of inertia about body frame's y-axis, I_y	3.58×10^{-2} kg·m^2
Moment of inertia about body frame's z-axis, I_z	6.73×10^{-2} kg·m^2
Roll moment coefficient, C_{roll}	0.0414
Pitch moment coefficient, C_{pitch}	0.0207
Yaw moment coefficient, C_{yaw}	0
Maximum descent airspeed, V_{max}	4 m/s
Maximum net thrust, T_{max}	11 N
Time of flight, t_f	4 s

Table 4.2: Initial and terminal conditions for the landing scenario

Parameter	Value
Initial position, $[x_0, y_0, z_0]$	[3,4,5] m
Initial velocity, $[V_{x0}, V_{y0}, V_{z0}]$	[0,0,0] m/s
Initial orientation, $[\phi_0, \theta_0, \psi_0]$	[0,0,0] deg
Initial angular velocity, $[p, q, r]$	[0,0,0] deg/s
Terminal position, $[x_0, y_0, z_0]$	[0,0,0] m
Terminal velocity, $[V_{x0}, V_{y0}, V_{z0}]$	[0,0,0] m/s
Terminal orientation, $[\phi_0, \theta_0, \psi_0]$	[0,0,0] deg
Terminal angular velocity, $[p, q, r]$	[0,0,0] deg/s

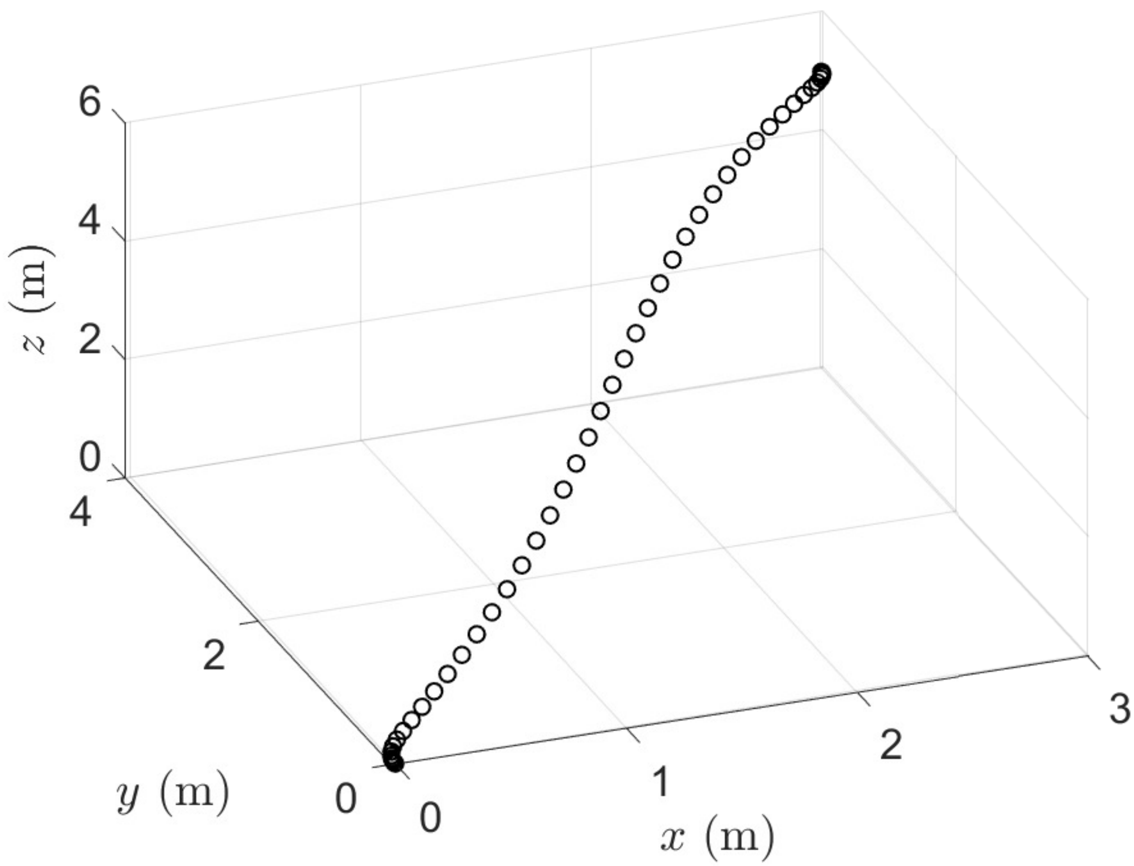


Figure 4.3: Three-dimensional 12-state optimal landing trajectory by SCP.

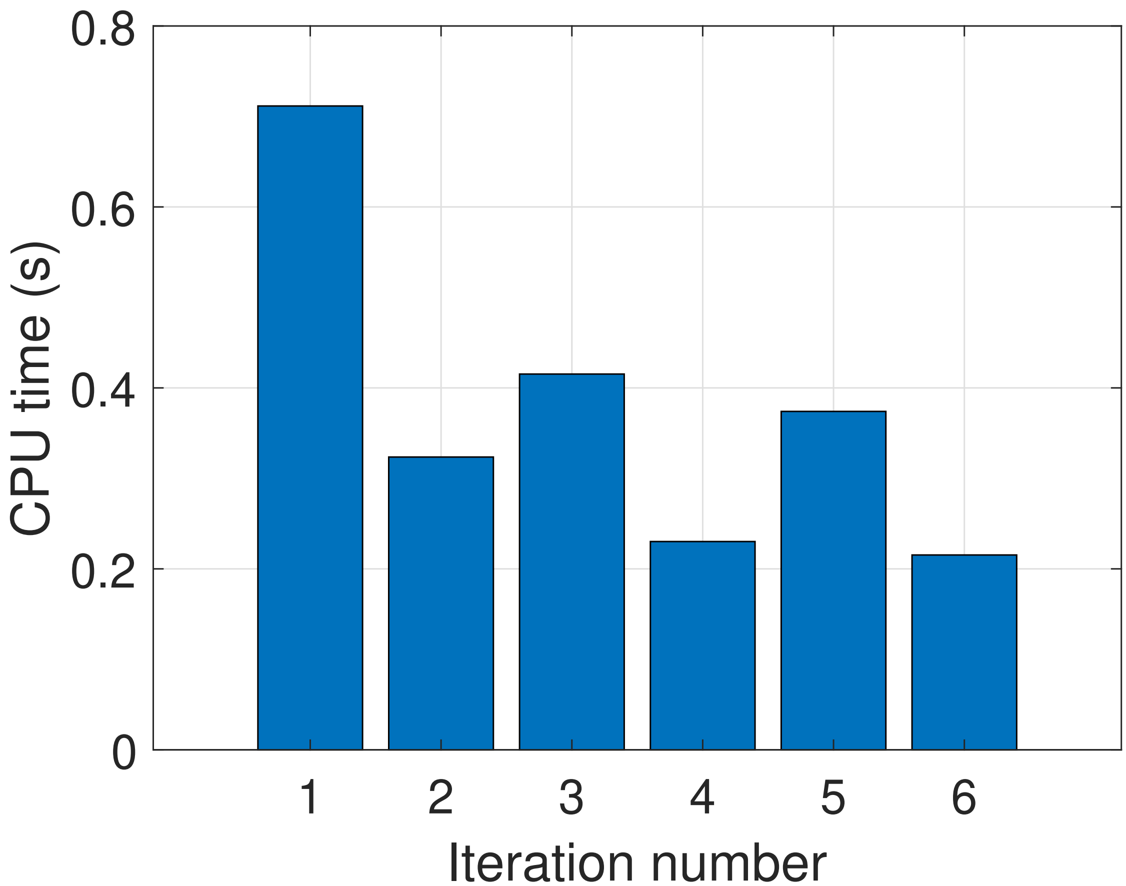


Figure 4.4: Computational time of each subproblem for SCP algorithm.

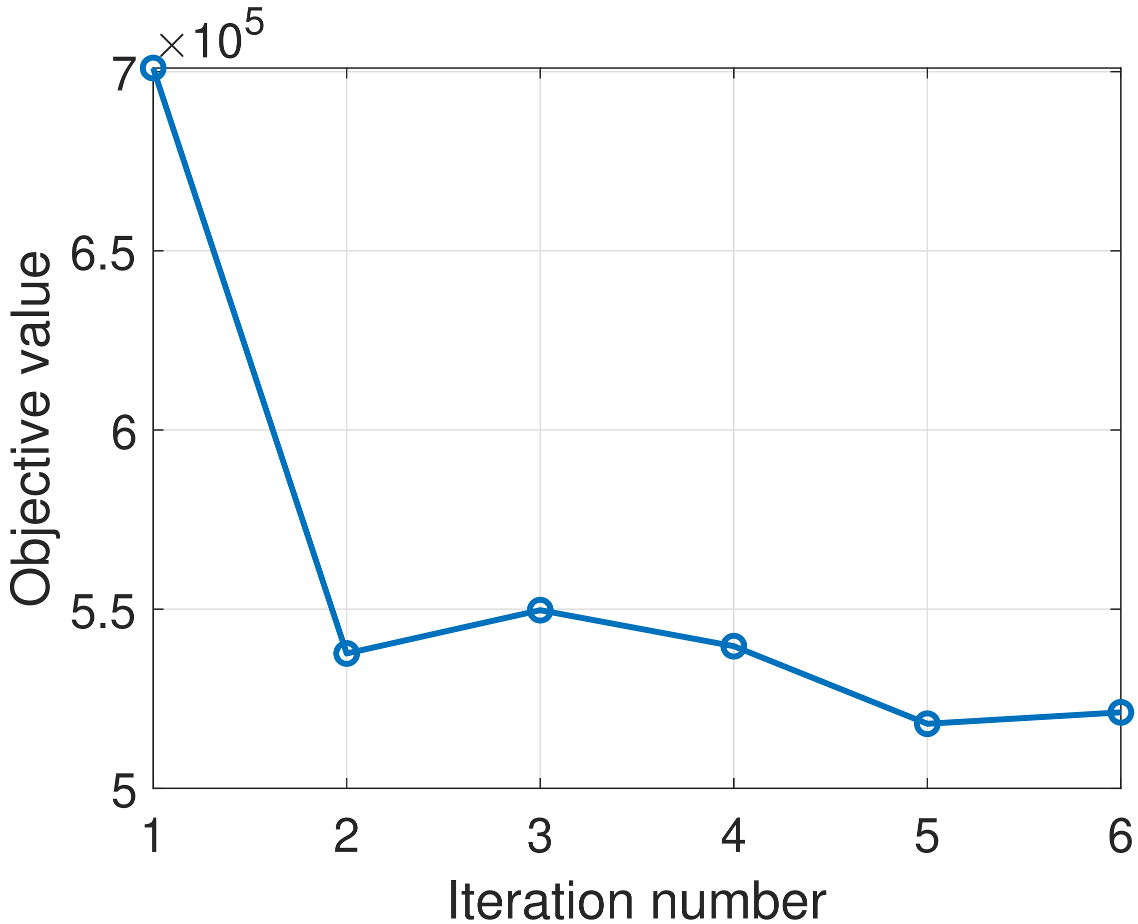


Figure 4.5: SCP convergence profile for objective function.

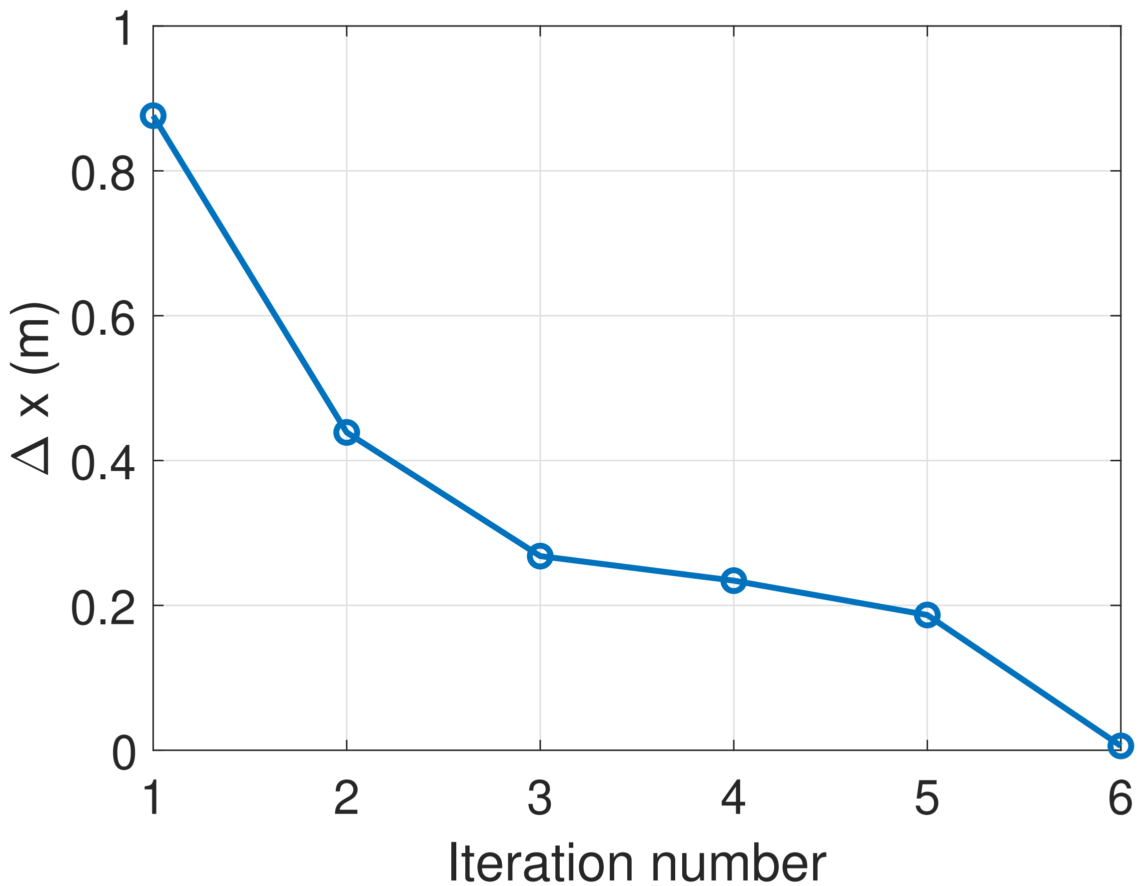


Figure 4.6: SCP convergence profile for x direction.

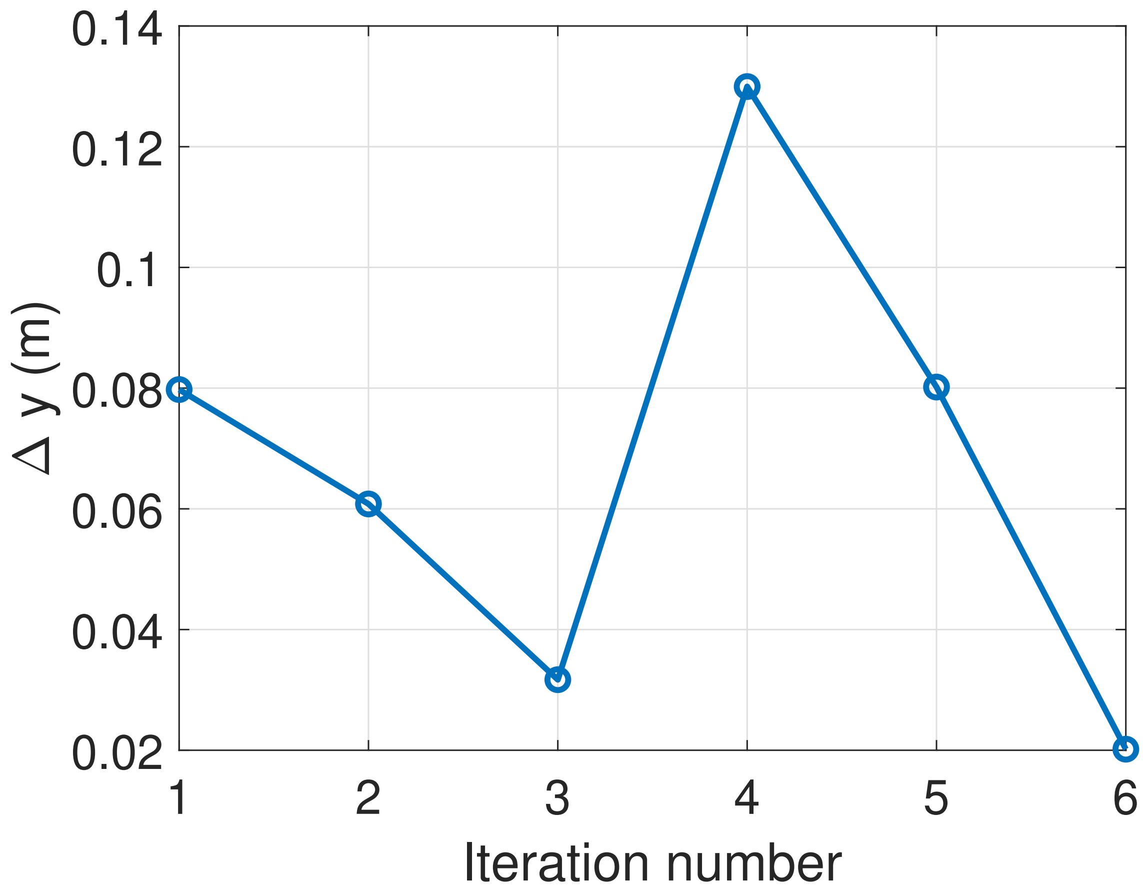


Figure 4.7: SCP convergence profile for y direction.

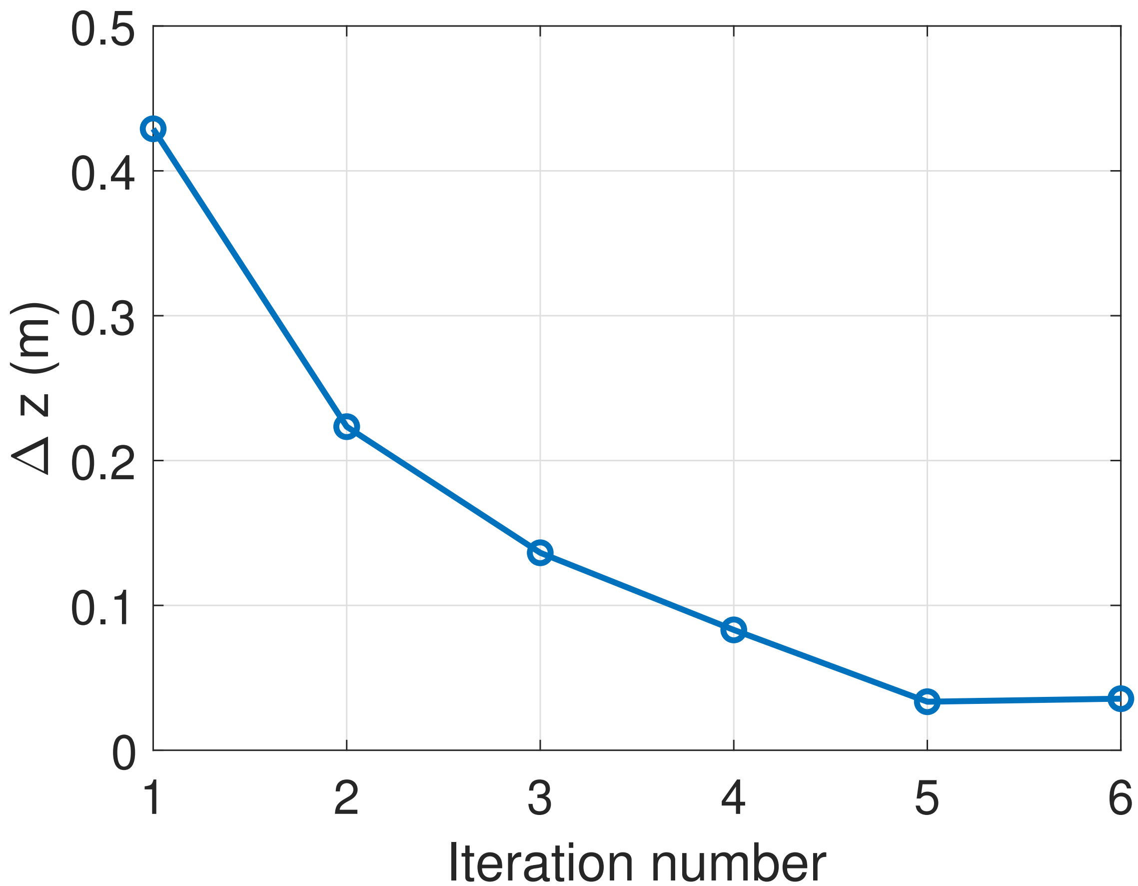


Figure 4.8: SCP convergence profile for z direction.

from the solvers. It is evident from the comparison that the SCP algorithm has converged to a significantly smoother thrust profile. This smoother profile is likely a result of the reduced nonlinearity inherent in the convexification process. The smoother thrust curve achieved by the SCP algorithm may contribute to a more comfortable and safer flight experience, especially in the UAM context where precision and smoothness in trajectory control are the priority. Figure 4.12 presents a comparison of torques in the x, y, and z directions. This plot reveals that the torque profile for the 12-state model exhibits some deviations when compared to the 16-state model, as solved by both GPOPS and SCP. This underscoring the impact of aerodynamics on the system's dynamics, while the latter two models demonstrate a closely aligned trend. Figures 4.13 and 4.14 provide insightful comparisons of velocity and angular position curves, respectively, across the different solutions. Similar to the observations made in the thrust curve comparison, all the three solutions demonstrate closely aligned trends in both velocity and angular position while the solution derived from the SCP method exhibits slightly higher smoothness.

Lastly, Table 4.3 presents a comparison of computation times between the two solvers under different models, further highlighting the efficiency of the SCP method relative to the traditional GPOPS solver. Specifically, for the 12-state problem, the GPOPS solver took eight seconds to converge. However, when the model complexity was elevated to 16 states, the GPOPS solver required significantly more time, approximately 155 seconds, to reach convergence. In contrast, the SCP method demonstrated remarkable efficiency, solving the same 16-state problem in merely three seconds, which is over 50 times faster than GPOPS. This advantage in computational times underscores the superior efficiency of the SCP method, particularly as the problem complexity escalates. These findings show that the SCP method is a suitable option for real-time trajectory optimization.

4.3.2 Simulation with 28 states

To further refine our analysis and enhance the precision of our trajectory optimization, we have expanded the model to include the full 28-state optimization problem that incorporates more complex dynamics and aerodynamic interactions. This extended model enables a more comprehensive assessment of the eVTOL's performance under realistic operating conditions.

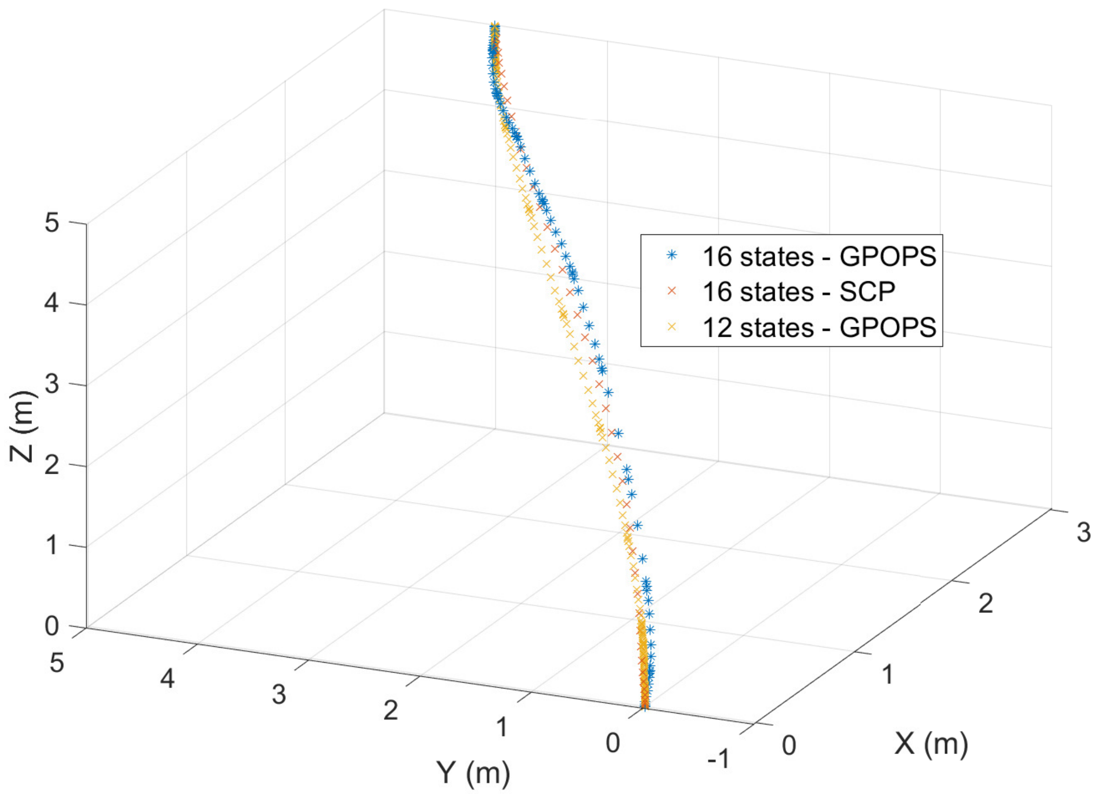


Figure 4.9: Trajectory comparison between GPOPS and SCP.

Table 4.3: Comparison of computational time of different solvers

12 states (GPOPS)	**16 states (GPOPS)**	**16 states (SCP)**
8 s	155 s	3 s

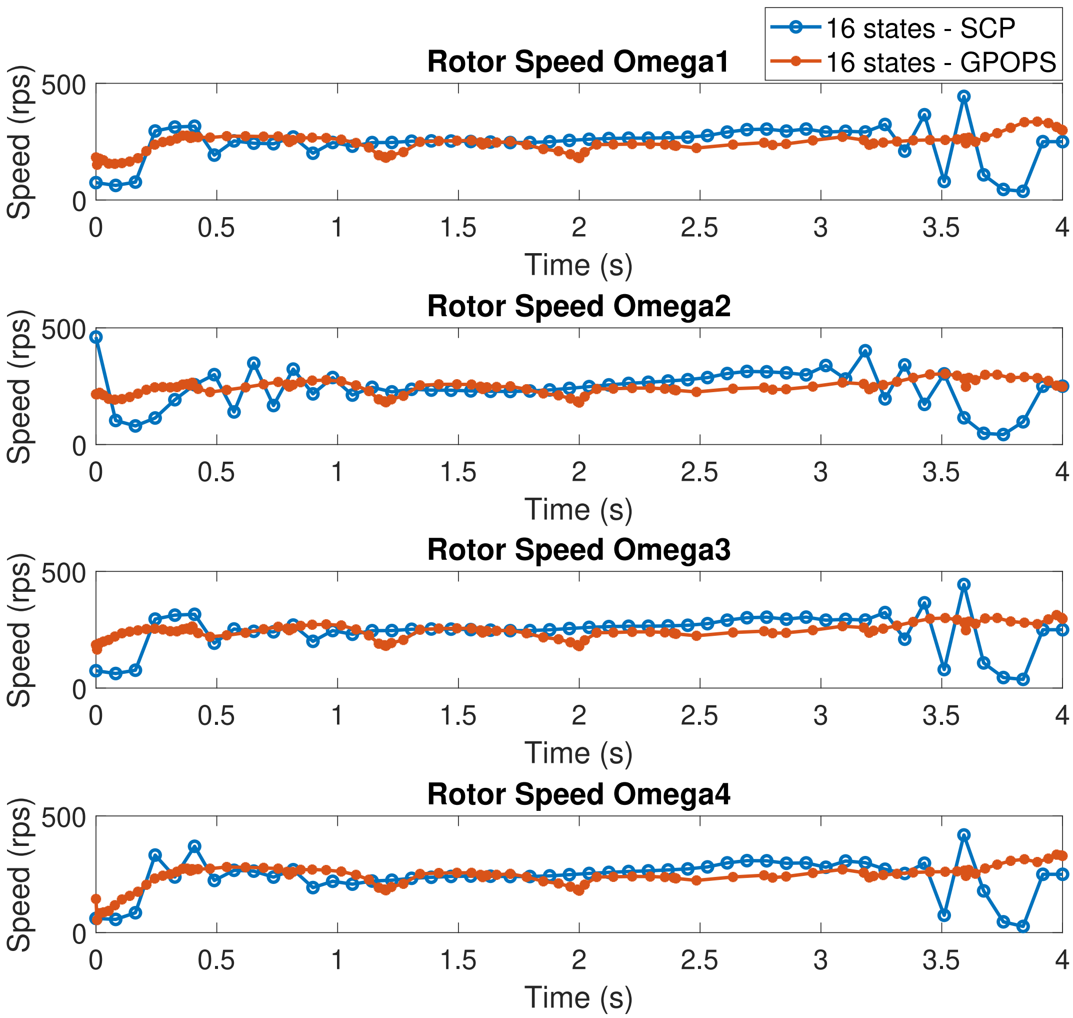


Figure 4.10: Rotor speed comparison between GPOPS and SCP.

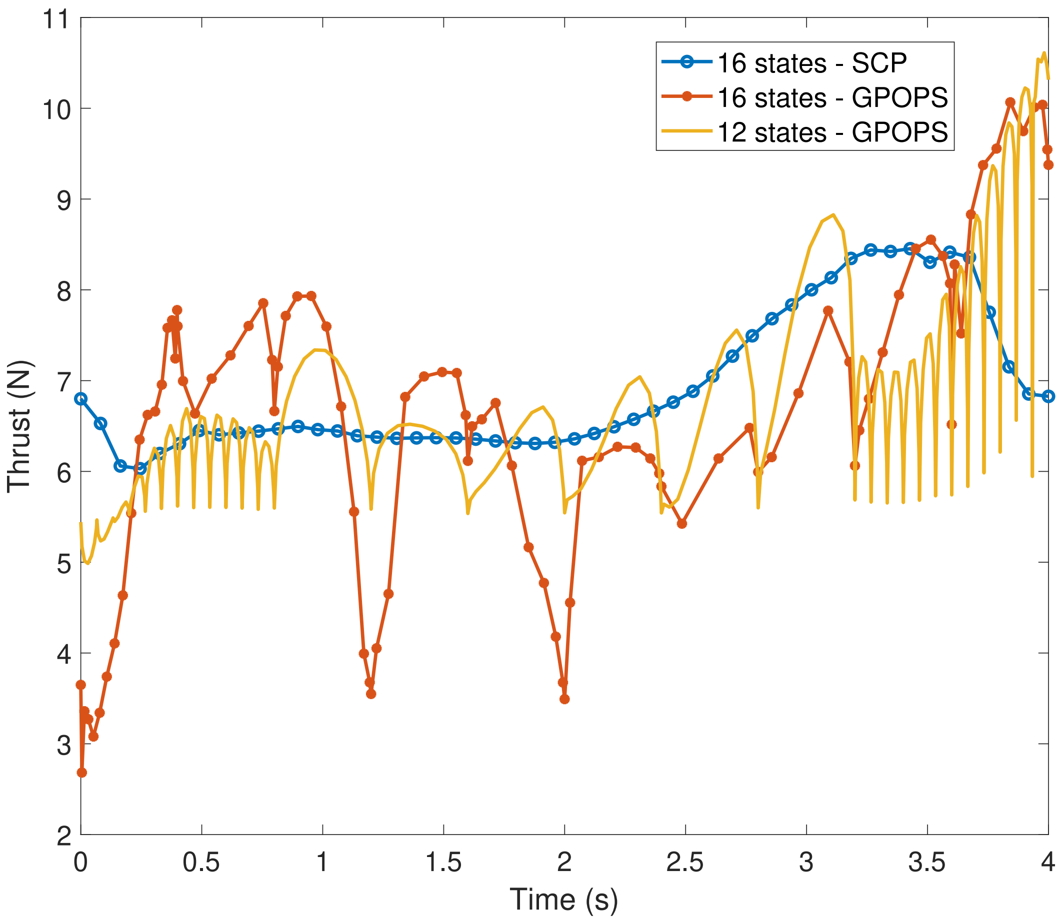


Figure 4.11: Thrust comparison between GPOPS and SCP.

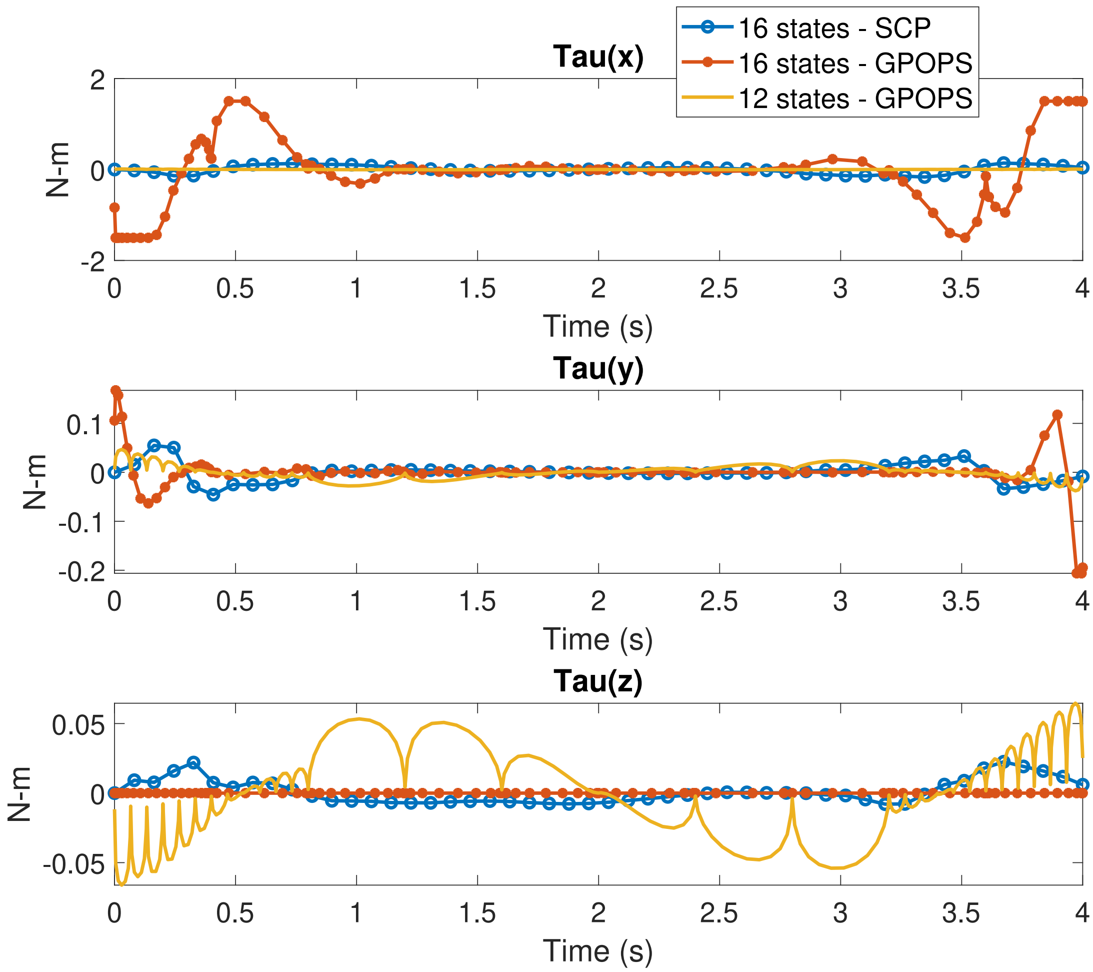


Figure 4.12: Torques comparison between GPOPS and SCP.

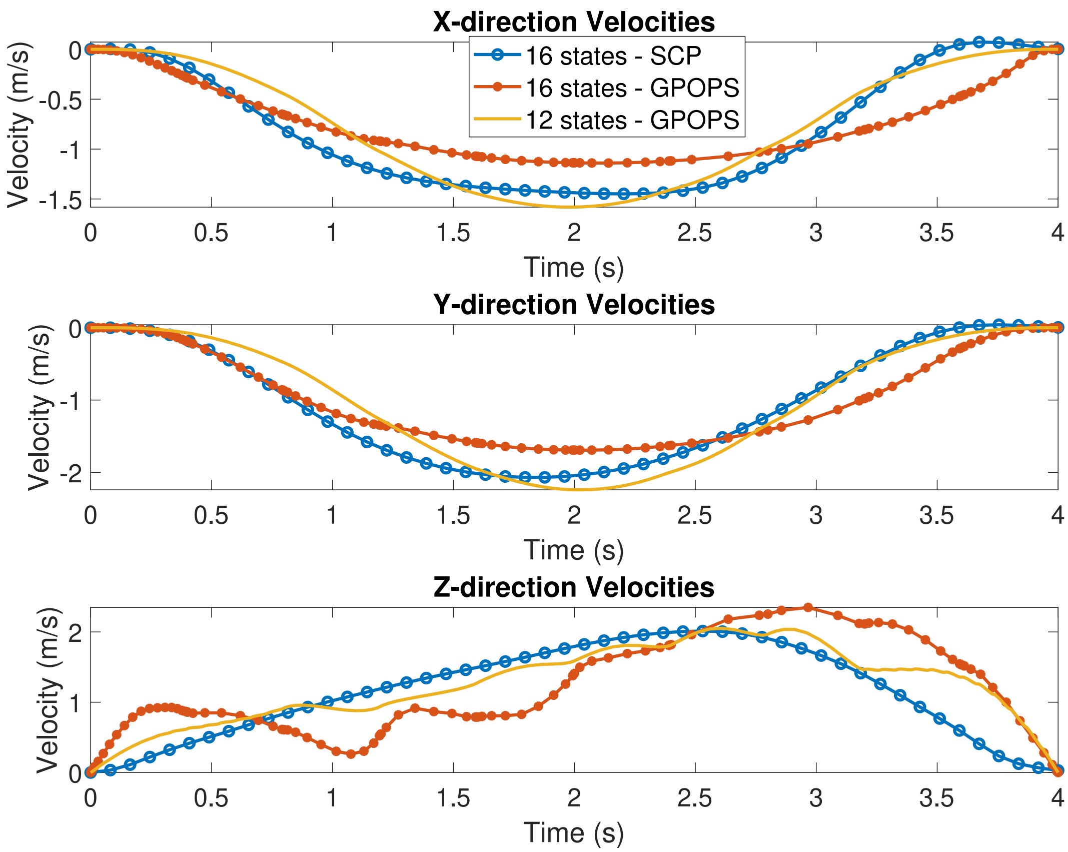


Figure 4.13: Velocity profile comparison between GPOPS and SCP.

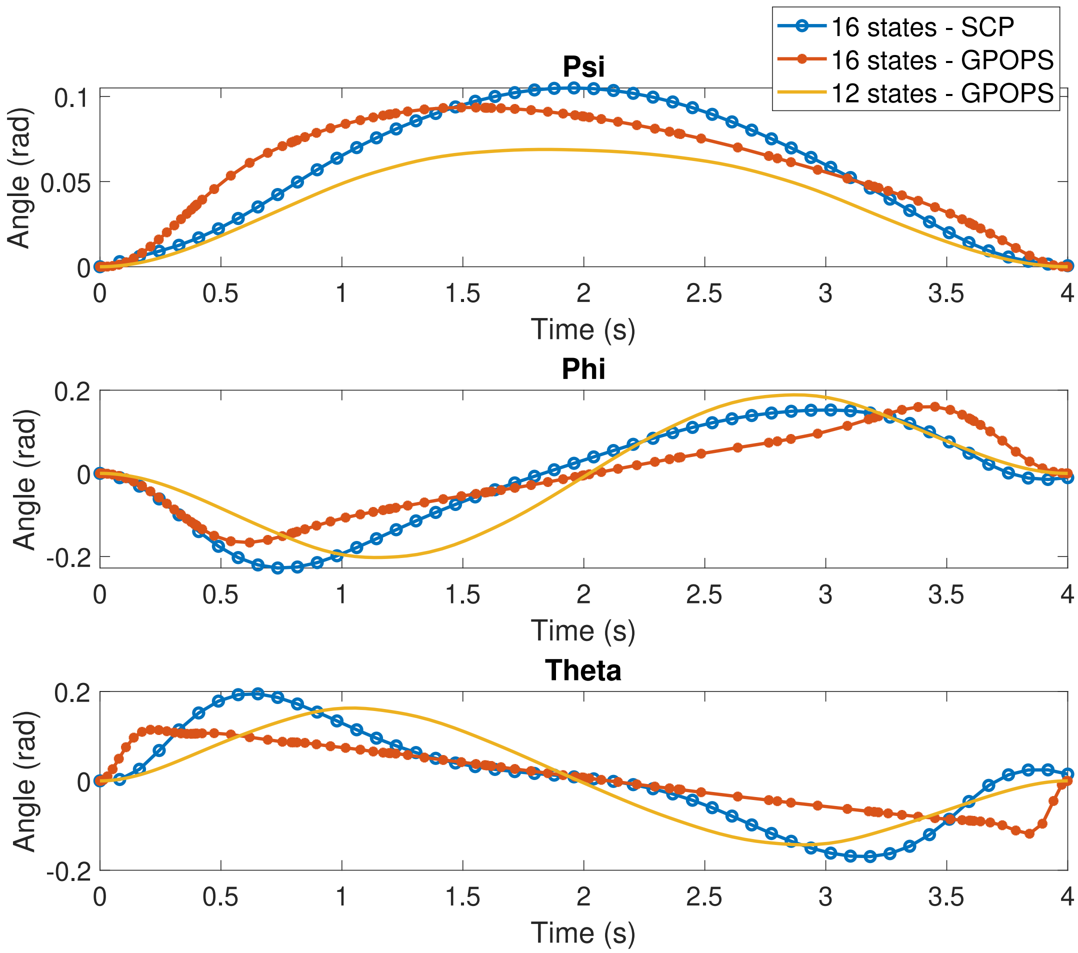


Figure 4.14: Angular position comparison between GPOPS and SCP.

These simulations also serve as a rigorous test of our SCP-based approach, highlighting its capability to manage increased complexity while maintaining computational feasibility on standard hardware.

Remark 4.1: Handling Aerodynamic States in Objective Function

In the trajectory optimization of eVTOL vehicles, accurately guessing initial and final conditions for aerodynamic states can be particularly challenging. These difficulties may lead to scenarios that appear artificially infeasible, compromising the integrity and feasibility of the optimization process. To address this issue, a new approach is adapted by modifying how these boundary conditions are handled within the optimization framework. Instead of setting strict constraints on the aerodynamic boundary conditions, which can be overly restrictive, these states are now penalized in the objective function. This adjustment allows for greater flexibility and avoids the potential pitfalls of infeasibility due to poor initial guesses.

The revised objective function incorporates penalties for deviations from desired boundary states, thus ensuring that the solution respects these conditions without necessitating exact adherence, which might not always be possible due to the complexity and variability of aerodynamic behaviors. The new objective function is defined as follows:

$$J_{\text{new}} = \int_{t_0}^{t_f} \left(\sum_{I=1}^{4} \Omega_I^2(t) + w_v \cdot \text{ess sup} \|E(t)\mathbf{v}(t)\|_1 \right) dt + \sum_{j=1}^{n} \lambda_j \|x_j(t_f) - x_{j,\text{target}}\|^2 \quad (4.44)$$

In this revised formulation:

- The first term, as in the original objective function (4.42), minimizes the sum of the squares of the rotor speeds Ω_I and the essential supremum norm of the aerodynamic forces weighted by w_v.
- The new term added, $\sum_{j=1}^{n} \lambda_j \|x_j(t_f) - x_{j,\text{target}}\|^2$, represents the penalty for the deviation of selected aerodynamic states at the final time t_f from their target values $x_{j,\text{target}}$. Each state's deviation is weighted by a penalty coefficient λ_j, which can be

adjusted to reflect the relative importance of accurately meeting that state's target value at the final time.

This modified objective function is designed to improve the practical feasibility of the optimization problem while still guiding the system toward a trajectory that meets operational and safety criteria. This method strikes a balance between strict adherence to boundary conditions and the flexibility needed to handle the unpredictable nature of aerodynamic states in real-world scenarios.

Figure 4.15 presents a trajectory comparison between the GPOPS and SCP methods for a system with 28 states. The figure clearly demonstrates that both solvers have converged to similar trajectories, indicating that both methods are effective in finding similar optimal solutions. This convergence suggests that the underlying dynamics and constraints are accurately captured by both approaches, leading to robust performance across different solvers.

Figure 4.16 presents the angular position outcomes, where the angles ϕ and θ from both methods align closely, showcasing the efficacy and accuracy of both approaches in simulating angular dynamics. A slight variation is observed in the ψ angle, though the magnitude of this discrepancy remains minimal, indicating the solvers might converge to slightly different local solutions.

Moving to Figure 4.17, the velocity profiles are compared, revealing that both methods track similar trends across the trajectory. Notably, the SCP method produces a smoother velocity curve, suggesting its robustness in handling rapid dynamic changes smoothly. This feature is particularly beneficial for ensuring stable and realistic flight dynamics in real-world eVTOL operations.

Figure 4.18 illustrates the rotor speed profile resulting from the two methods. It is evident from the figure that the SCP method produces a smoother control profile compared to GPOPS. This smoother profile can likely be attributed to the reduced nonlinearity introduced during the convexification process inherent to the SCP method. The convexification reduces the complexity of the system dynamics, leading to a more regularized and gradual control input, which is beneficial in maintaining stability and efficiency in practical applications.

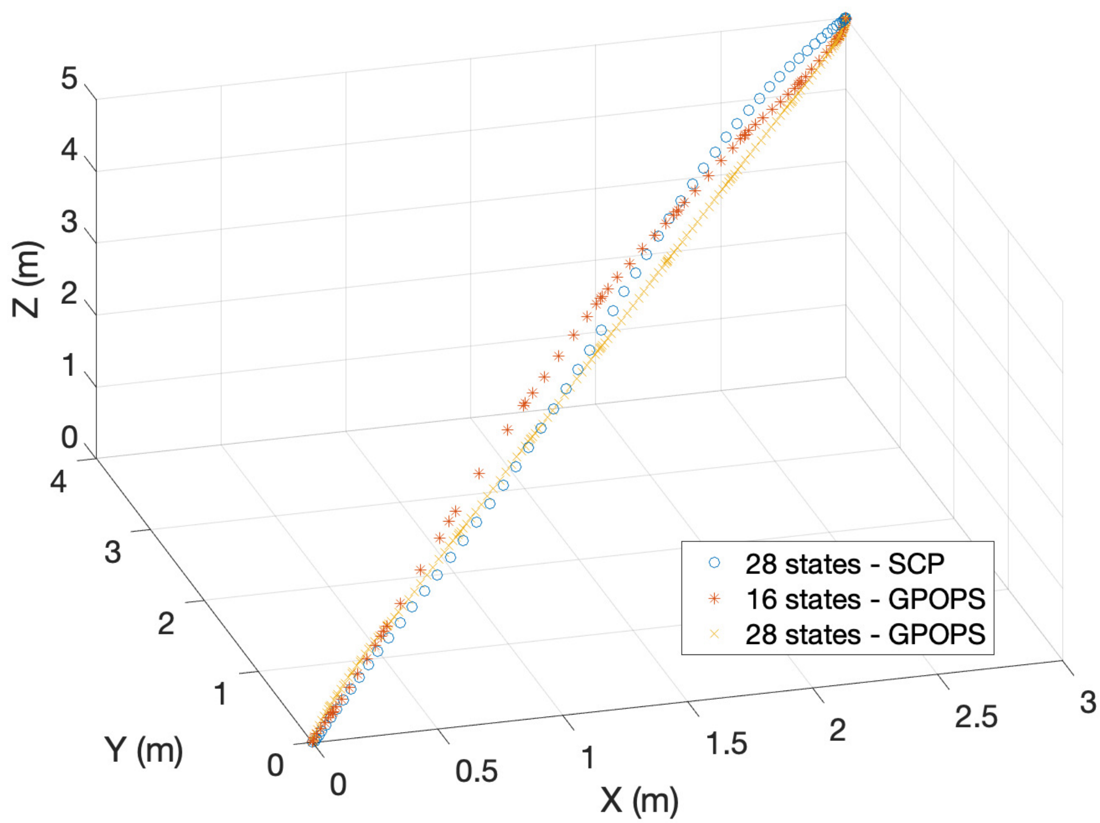


Figure 4.15: Trajectory comparison between GPOPS and SCP for 28 states.

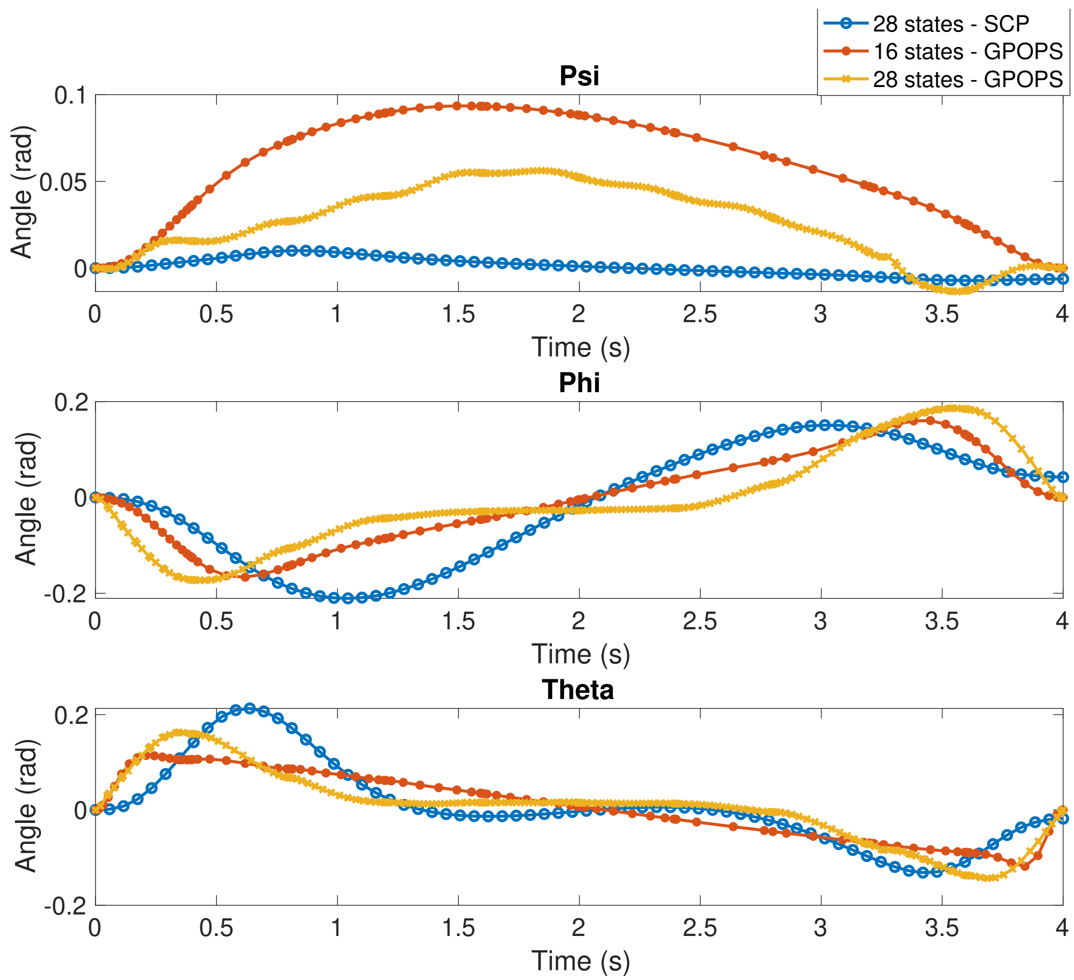


Figure 4.16: Angular position comparison between GPOPS and SCP for 28 states.

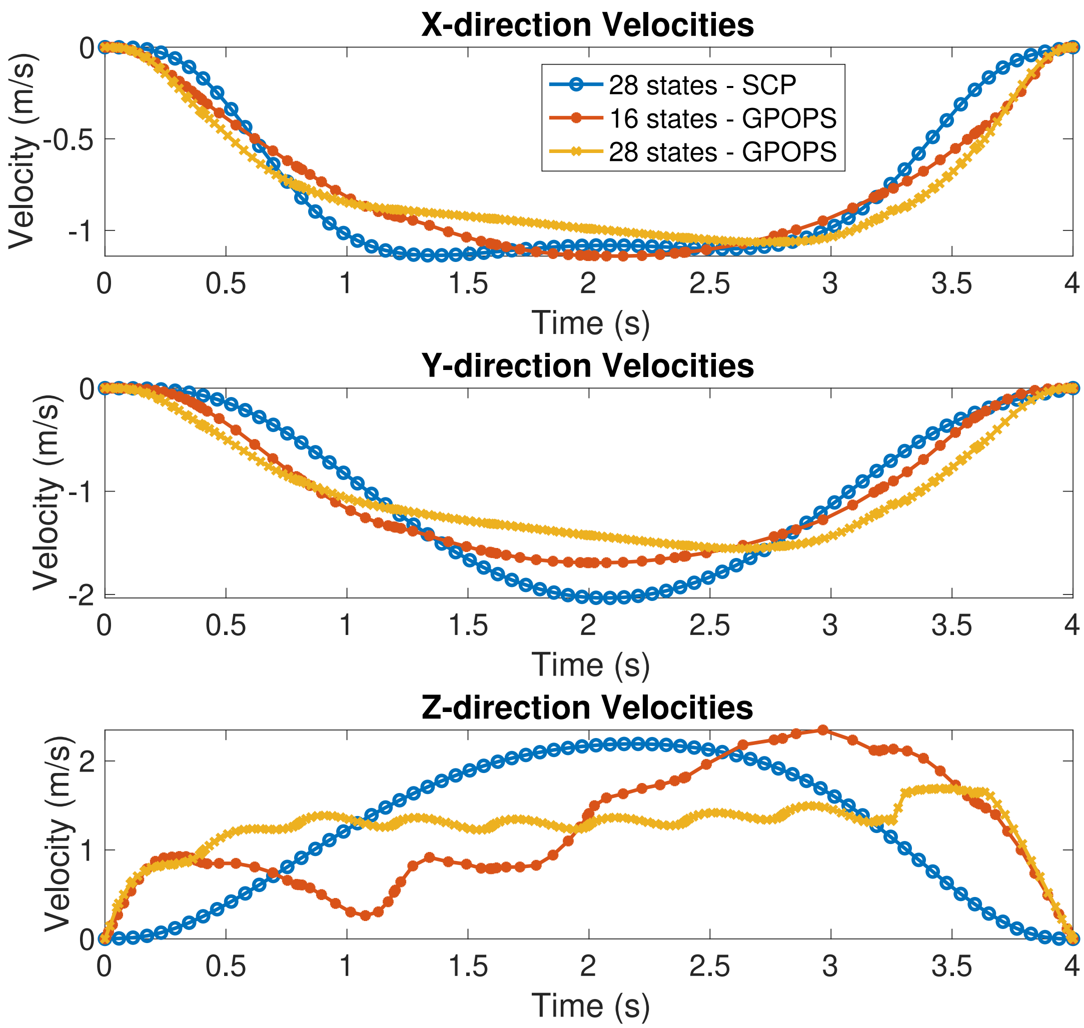


Figure 4.17: Velocity comparison between GPOPS and SCP for 28 states.

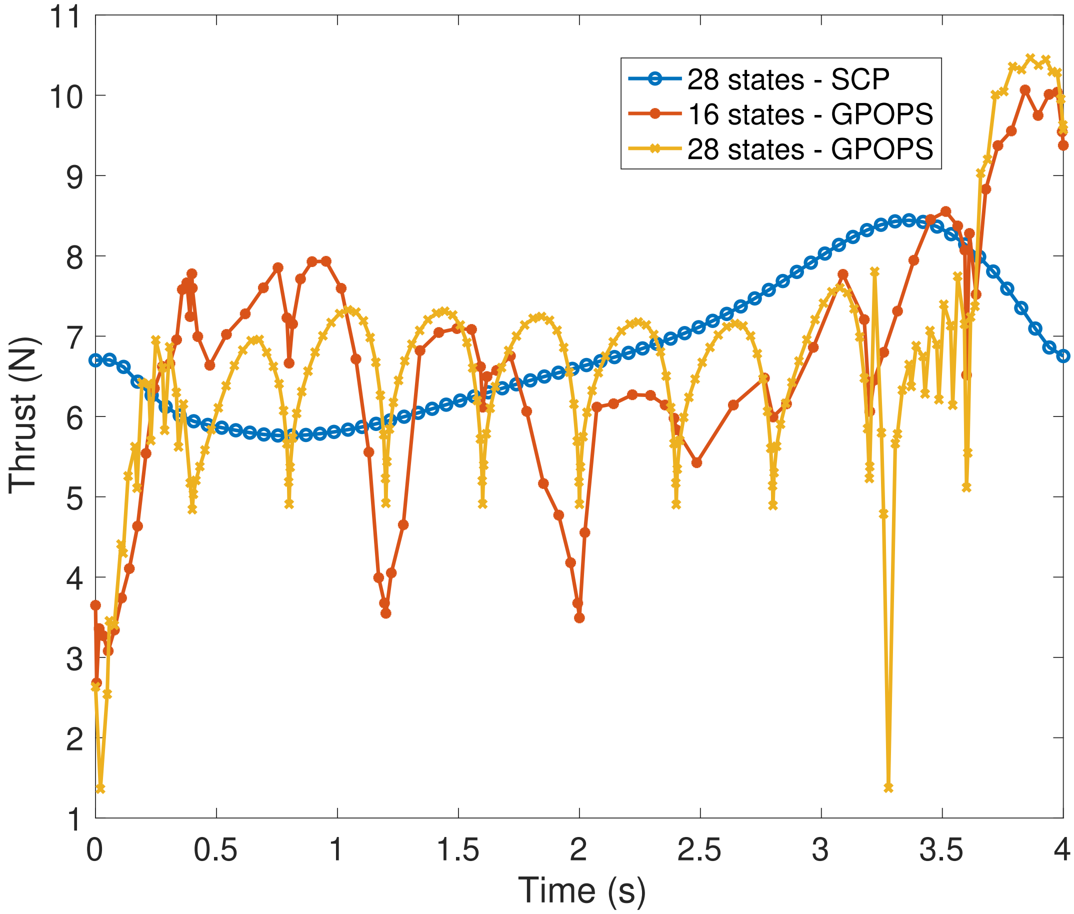


Figure 4.18: Thrust comparison between GPOPS and SCP for 28 states.

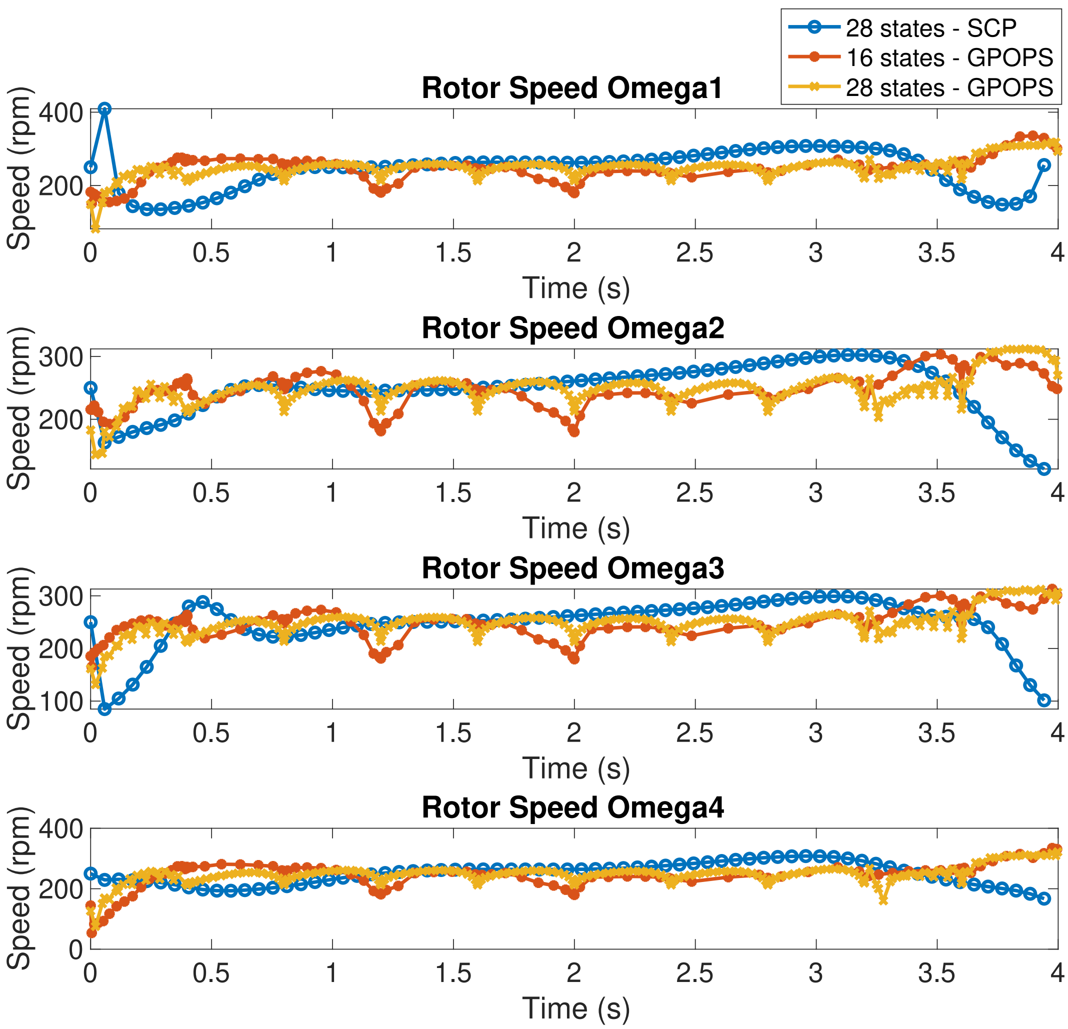


Figure 4.19: Rotor speed comparison between GPOPS and SCP for 28 states.

Lastly, Figure 4.19 examines the thrust outputs, where the smoother control response generated by the SCP method is again evident. The thrust profile via SCP exhibits less variability and a more refined control strategy, contrasting with the slightly more abrupt changes observed with GPOPS. This smoother thrust curve by SCP aligns with the control outputs, potentially translating to more efficient and controlled eVTOL maneuvers.

Remark 4.2: Jitters in GPOPS solution In attempting to address the 28 states problem, the GPOPS solver exhibited significant challenges in achieving convergence, demonstrating noticeable jitters in the resultant outputs. Notably, GPOPS employs an hp-adaptive discretization approach using Legendre-Gauss-Radau (LGR) points. This discretization strategy, while robust for a variety of problems, enforces boundary conditions only at one end of the mesh interval. Such a configuration can contribute to sinusoidal or oscillatory behaviors in the control curves, particularly in highly complex models like the 28 states scenario we investigated.

4.4 Summary

Vehicle landing problems under different scenarios receive increasing attention from researchers as the potential environmental factors can significantly affect the aerodynamic behavior of the eVTOL vehicle. Therefore, in this study, we investigate the possibility of combining the rotor aerodynamic model with the flight dynamics to formulate an optimal control problem for high-fidelity landing trajectory optimization. This integration is crucial in urban environments where precision and reliability in landing can be heavily affected by the complex and dynamic nature of these settings. In this research, we have shown that the SCP method can outperforms traditional nonlinear programming approaches in terms of computational efficiency. This improvement is a critical step toward real-time solution capabilities, a key requirement for the practical deployment of autonomous eVTOL vehicles in urban airspace.

Chapter 5

Aim III: Develop a convex approach to merging trajectory optimization of eVTOL vehicles

The merging control of UAM vehicles flying along different airways or air corridors would become a bottleneck in the development and implementation of future UAM systems. In this work, we consider the merging control of two eVTOL vehicles and formulate an optimal control problem that incorporates the flight dynamics of the vehicle and collision avoidance constraint. Two objective functions, i.e., minimum flight time and minimum control cost, are considered in the problem formulation. The problems are first solved using a general-purpose optimal control solver (GPOPS). Then, we introduce several new control variables to reduce the nonlinearity of the dynamic system, apply a lossless convexification technique to relax the resulting nonconvex control constraint, linearize the nonlinear terms in the dynamics, and develop a sequential convex programming method as an alternative approach to the UAM merging control problem considered in this study.

5.1 Problem Formulation

Different from the ground vehicle merging control problem that has been usually formulated as a one-dimensional problem, air vehicles operate in a three-dimensional space and are more complex to model. In this research, we model the motion of air vehicles in both horizontal and vertical dimensions. We propose to use a decentralized framework to solve a merging control problem of two vehicles. In this problem setup, one vehicle is assumed to fly in the main corridor with a certain fixed altitude and constant velocity while the other vehicle approaches and merges into the main corridor from a lower altitude. Unlike ground vehicles that can only drive on the roads and have a fixed merging point, eVTOL vehicles can take any possible path from the initial flight condition to the main corridor with a flexible merging location. The merging problem is illustrated by Fig. 5.1 below.

5.1.1 System Dynamics

The EHang 184 quad-copter eVTOL vehicle [1] is used as a reference vehicle model for this research. The formulated problem as well as the optimization methods developed in this research can be easily adapted to other eVTOL vehicles such as CityAirbus, Airbus Vahana, and Volocopter 2X, through slight modifications. In this research, we assume that the eVTOL vehicles take a geodesic path between the initial and final positions, and only the motion in the vertical plane is considered. The dynamic model used in this research is adapted from [59]. The equations of motion are as follows:

$$\dot{x} = V_x \tag{5.1}$$

$$\dot{z} = V_z \tag{5.2}$$

$$\dot{V}_x = \frac{T\sin\theta}{m} - \frac{D_x}{m} \tag{5.3}$$

$$\dot{V}_z = \frac{T\cos\theta}{m} - \frac{D_z}{m} - g \tag{5.4}$$

where x represents the along-track distance and z represents the altitude, which together composes the position vector of the vehicle. V_x and V_z represent the horizontal and vertical components of the vehicle speed, respectively. T is the net thrust, m is the vehicle mass, g is the gravitational acceleration that is assumed to be constant, and θ is the rotor tip-path-plane pitch angle. The state vector and the control vector of the system are defined

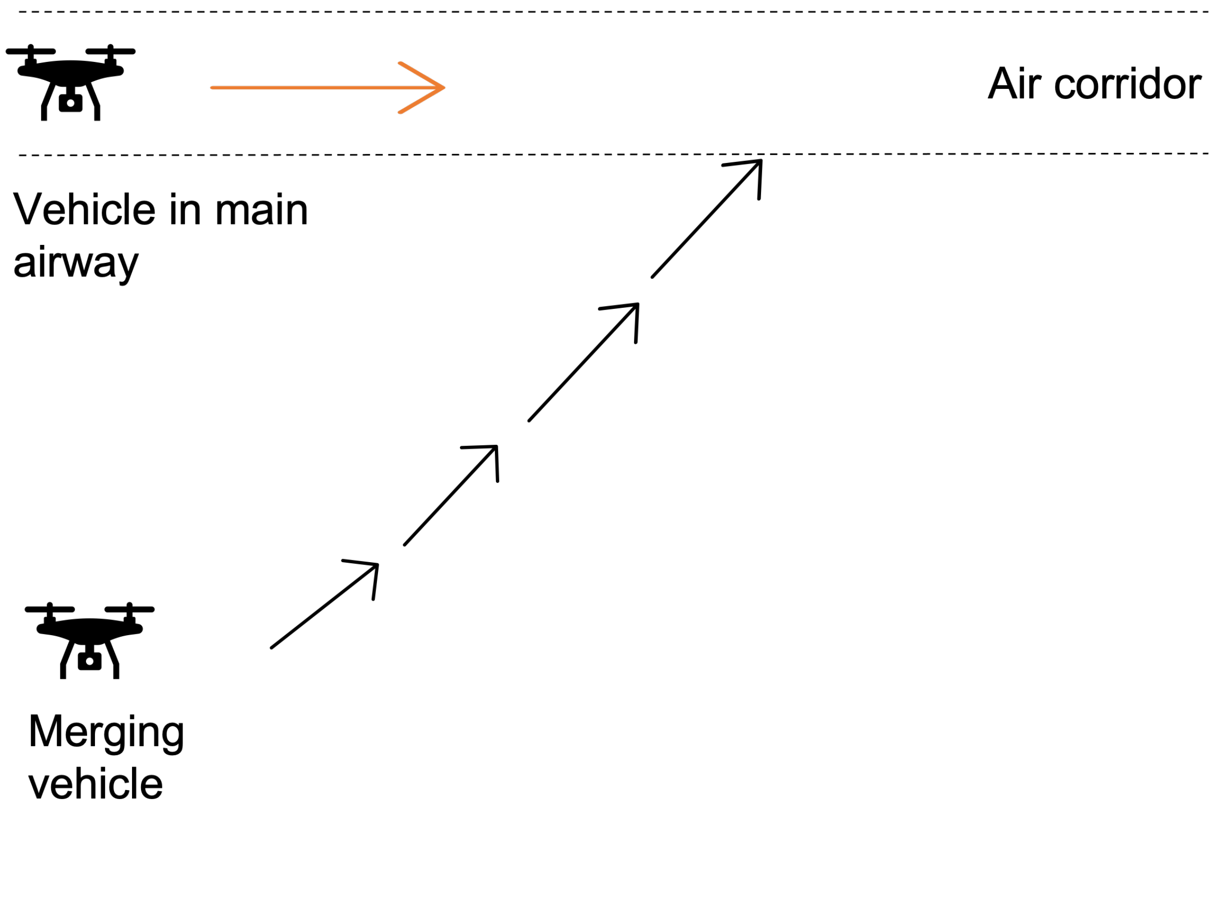


Figure 5.1: Merging control of two eVTOL vehicles.

as:

$$\mathbf{y} = [x, z, V_x, V_z]^T \text{ and } \mathbf{u} = [T, \theta]^T \tag{5.5}$$

The horizontal (D_x) and vertical (D_z) components of the aerodynamic drag can be calculated from the equations below:

$$D_x = \frac{\rho V_x^2 C_D S_x}{2} \text{ and } D_z = \frac{\rho V_z^2 C_D S_z}{2} \tag{5.6}$$

where ρ is the atmospheric density, C_D is the aerodynamic drag coefficient, and S_x and S_z are the reference front and top flat plate areas of the fuselage, respectively. For simplicity, all these four variables are assumed to be constant in this research. Thus, the drag force D_x and D_z are only dependent on the velocity components.

5.1.2 Flight Constraints

To ensure the safety of the flight as well as the passenger's comfortability, the eVTOL vehicles have to meet various constraints. First, we impose the initial and terminal conditions for the merging vehicle by introducing the following boundary conditions:

$$\mathbf{y}(t_0) = \mathbf{y}_0 \tag{5.7}$$

$$\mathbf{y}(t_f) = \mathbf{y}_f \tag{5.8}$$

In this study, the along-track distance x of the final state is free to optimize. Then, we define the maximum distance to the merging point and maximum altitude as follows:

$$0 \leq x \leq x_{max} \tag{5.9}$$

$$0 \leq z \leq z_{max} \tag{5.10}$$

Meanwhile, the maximum airspeed, maximum thrust, and the vehicle's maximum pitch angle are restricted and specified below:

$$\sqrt{V_x^2 + V_z^2} \leq V_{max} \tag{5.11}$$

$$0 \leq T \leq T_{max} \tag{5.12}$$

$$|\theta| \leq \theta_{max} \tag{5.13}$$

Moreover, for the merging control problem, vehicle safety is the top concern. Therefore, we enforce a collision avoidance constraint on the vehicle by applying the following inequality constraint:

$$\sqrt{(x - x_2)^2 + (z - z_2)^2} \geq d \tag{5.14}$$

where x_2 and z_2 denote the position of the vehicle operating on the main air corridor, and d is a predefined safe distance between two vehicles.

Remark 1: In the context of this research, "decentralized" refers to the ability of the merging vehicle to perform trajectory optimization independently, based on its own optimal control goals. While the preliminary research has modeled vehicles on the main airway as flying at a constant speed, the merging vehicle autonomously adjusts its trajectory. For future work, we can enhance the realism of this model by introducing random fluctuations in speed or employing simple aerodynamics for more realistic simulations of all vehicles involved. This approach underscores the decentralized nature by focusing on individual vehicle autonomy within the merging process.

5.1.3 Performance Index and Optimal Control Problem

In this research, we formulate the considered merging control problem as an optimal control problem, which is classified into two cases based on different performance indexes including the minimization of an individual vehicle's total thrust:

$$\text{Minimize} \quad J = \int_{t_0}^{t_f} T dt \tag{5.15}$$

and minimization of individual vehicle's final time:

$$\text{Minimize} \quad J = t_f \tag{5.16}$$

With two different objective functions and constraints, the optimal control problem is formulated as:

Problem 1:

$$\text{Minimize:} \quad (5.15) \text{ or } (5.16)$$
$$\text{Subject to: } (5.1), (5.2), (5.3), (5.4), (5.5),$$
$$(5.6), (5.7), (5.8), (5.9), (5.10), (5.11), (5.12), (5.13), (5.14)$$

Overall, we want to minimize the objective function (5.15) or (5.16) while satisfying the system dynamics (5.1)-(5.4), the boundary conditions (5.7) and (5.8), the inequality state and control constraints (5.9)-(5.13), and the collision avoidance constraint (5.14).

5.2 Methodologies

In this section, we provide technical details about the pseudospectral method and the convex optimization approach applied in this research. Subsection 5.2.1 briefly introduces the pseudospectral method that can be found in the literature. The convexification techniques used in this research are discussed in subsection B, and a sequential convex programming approach is developed in the last part of this section.

5.2.1 Pseudospectral Method

In this research, we first exploit GPOPS [55], a powerful optimal control solver, which builds on a pseudospectral collocation method. The pseudospectral method belongs to the family of direct collocation methods, where the state is approximated by basis functions and the dynamics are collocated at specific points. For the pseudospectral method, the collocation points are usually based on Lagrange polynomials. Originally, Gaussian quadrature collocation points were selected by p methods using a single collocation interval. Convergence of the p method was achieved by increasing the degree of the polynomial approximation. However, this approach may become too computationally expensive in the case of solution trajectories with rapidly changing or discontinuous state variables. To solve this problem, GPOPS solver uses an hp-adaptive method, where h refers to increasing the segment numbers and p refers to changing the polynomial orders [17].

The first step of the pseudospectral method is domain transformation. We want to map the problem from the physical domain, $t \in [t_o, t_f]$, to a computational domain, $\tau \in [-1, 1]$, with the following equations:

$$t = \Gamma(\tau) = \left(\frac{t_f + t_0}{2}\right) + \left(\frac{t_f - t_0}{2}\right)\tau \tag{5.17}$$

$$\tau = \Gamma^{-1}(t) = \frac{2t - (t_f + t_0)}{t_f - t_0} \tag{5.18}$$

After that, the state $\mathbf{Y}(\tau)$ is approximated by using values of $\mathbf{Y}$ at discrete time points and the Lagrange polynomial basis functions Φ_i, which can be written as:

$$\mathbf{Y}(\tau) = \prod_{i=1}^{N+1} \mathbf{y}(\tau_i)\Phi_i(\tau) \tag{5.19}$$

where $\mathbf{y}(\tau)$ is the state matrix, and $\Phi_i(\tau)$ is the Lagrange polynomial basis function defined as:

$$\Phi_i(\tau) = \prod_{j=1, j\neq i}^{N+1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{5.20}$$

From the equation above, the derivatives can be computed by multiplying a differential matrix, $\mathbf{D}$, with the state:

$$\dot{\mathbf{y}} \approx \mathbf{D}\mathbf{y} \tag{5.21}$$

Then, we can rewrite the dynamic constraints into defect equality constraints as follows:

$$\mathbf{h}(\mathbf{y}, \mathbf{u}) = \mathbf{D}\mathbf{y} - \frac{1}{2}\Delta t\, \mathbf{f}(\mathbf{y}, \mathbf{u}) = \mathbf{0} \tag{5.22}$$

where $\mathbf{f}(\mathbf{y}, \mathbf{u})$ represents the dynamic equations matrix, Δt is the time step, and $\mathbf{h}(\mathbf{y}, \mathbf{u})$ are the resulting equality constraints. Finally, the original optimal control problem is transformed into a nonlinear programming (NLP) problem, which can be solved by NLP solvers such as IPOPT or SNOPT.

5.2.2 Convex Optimization

From Problem 1, we recognize that the system dynamic equations are nonlinear, which will deteriorate the convergence of the NLP method. Additionally, the highly coupled control and state variables usually can lead to high-frequency jitters in the control profiles [80]. To address these issues, we first introduce the following new variables:

$$u_1 = T\sin\theta,\ u_2 = T\cos\theta,\ \text{and } u_3 = T \tag{5.23}$$

Along with the above new variables, a new constraint should be imposed:

$$u_1^2 + u_2^2 = u_3^2 \tag{5.24}$$

Then, the original system dynamics in (5.9)-(5.12) become:

$$\dot{x} = V_x \tag{5.25}$$

$$\dot{z} = V_z \tag{5.26}$$

$$\dot{V}_x = \frac{u_1}{m} - \frac{D_x}{m} \tag{5.27}$$

$$\dot{V}_z = \frac{u_2}{m} - \frac{D_z}{m} - g \tag{5.28}$$

and the objective function (5.15) takes the following new form:

$$J = \int_{t_0}^{t_f} u_3 dt \tag{5.29}$$

The boundary conditions and inequality constraints on the states remained the same as before; however, a set of inequality constraints has to be enforced on the new controls to ensure that (5.12) and (5.13) are met:

$$u_3 \sin(-\theta_{max}) \leq u_1 \leq u_3 \sin(\theta_{max}) \tag{5.30}$$

$$u_3 \cos(\theta_{max}) \leq u_2 \leq u_3 \cos(0) \tag{5.31}$$

$$0 \leq u_3 \leq T_{max} \tag{5.32}$$

Notice that the feasible set of the new equality constraint (5.24) represents the surface of a cone, which is nonconvex. However, we can apply a relaxation technique to convert the original constraint to an inequality constraint below:

$$u_1^2 + u_2^2 \leq u_3^2 \tag{5.33}$$

Along with the bounded u_3 in (5.32), (5.32) and (5.33) constitute a solid convex cone defined by two convex constraints.

Other than the nonconvex dynamic system, the collision avoidance constraint defined by (5.14) is also known to be nonconvex. To solve this problem we propose to use a different approach to impose the constraint as follows:

$$|x - x_2| + |z - z_2| \geq d^* \tag{5.34}$$

where d^* is the L_1 norm distance between the two vehicles. Finally, the nonconvex terms in the left-hand sides of the nonlinear dynamic system defined by (5.25)-(5.28) need also to be convexified. However, for this problem, to account for the free final time in the procedure, we defined a new variable τ over a fixed domain $[0, 1]$ to replace time t. The time dilation σ between t and τ is defined as:

$$\sigma = \frac{dt}{d\tau} = t_f - t_0 \tag{5.35}$$

and the objective functions are changed to:

$$J = \int_0^1 \sigma u_3 d\tau \tag{5.36}$$

and

$$J = \int_0^1 \sigma d\tau \tag{5.37}$$

To convexify the equations of motion, we apply a successive linearization method to replace the nonlinear equation with a first-order Taylor series expansion around a reference solution that will be updated sequentially as detailed in the next subsection. With the linear approximation method the dynamic equations become:

$$\mathbf{y}' = \frac{d\mathbf{y}}{d\tau} = \sigma \mathbf{f}(\mathbf{y}, \mathbf{u}, \tau) \approx A\mathbf{y} + B\mathbf{u} + F\sigma + c \tag{5.38}$$

where A, B, F, and c are obtained from below:

$$A = \bar{\sigma}\frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \tau) \tag{5.39}$$

$$B = \bar{\sigma}\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \tau) \tag{5.40}$$

$$F = \mathbf{f}(\bar{\mathbf{y}}, \bar{\mathbf{u}}, \tau) = \mathbf{f}(\bar{\mathbf{y}}) + B\bar{\mathbf{u}} \tag{5.41}$$

$$c = -(A\bar{\mathbf{y}} + B\bar{\mathbf{u}}) \tag{5.42}$$

By applying the trapezoidal rule, we obtain the current state Y_i from the previous step Y_{i-1}:

$$\begin{aligned} \mathbf{Y}_i = \mathbf{Y}_{i-1} + \frac{\Delta t}{2}[&(A_{i-1}^{k-1}\mathbf{Y}_{i-1} + B\mathbf{u}_{i-1} + \sigma(\mathbf{f}_{i-1}^{k-1} + B\mathbf{u}_{i-1}^{k-1}) - (A_{i-1}^{k-1}\mathbf{y}_{i-1}^{k-1} + B\mathbf{u}_{i-1}^{k-1})) \\ &+(A_i^{k-1}\mathbf{Y}_i + B\mathbf{u}_i + \sigma(\mathbf{f}_i^{k-1} + B\mathbf{u}_i^{k-1}) - (A_i^{k-1}\mathbf{y}_i^{k-1} + B\mathbf{u}_i^{k-1}))] \end{aligned} \tag{5.43}$$

Lastly, a trust-region constraint is added to the problem to improve the convergence of the proposed sequential optimization process:

$$|\mathbf{y}(\tau) - \mathbf{y}^*(\tau)| \leq \delta \tag{5.44}$$

where δ is a fixed trust-region radius and $\mathbf{y}^*$ is the reference states obtained from the last iteration of the sequential optimization algorithm. Now, a relaxed optimal control problem is formulated as follows:

Problem 2:

$$\text{Minimize:} \quad (5.36) \text{ or } (5.37)$$

$$\text{Subject to: } (5.7), (5.8), (5.9), (5.10), (5.11),$$

$$(5.34), (5.30), (5.31), (5.32), (5.33), (5.43), (5.44)$$

5.2.3 Sequential Convex Programming

In this study, a standard Sequential Convex Programming (SCP) framework was applied, an approach to iteratively tackle local convex optimization problems, which provides approximate solutions to the inherently nonconvex merging control problem (referred to as Problem 1). This framework was also employed to facilitate a comparative analysis with outcomes derived using the pseudospectral method implemented in the GPOPS solver. The SCP methodology used in this study includes several key steps:

1. Initialize the iteration index $k = 0$ and define the initial states $\mathbf{y}(t_0)$. Then insert the initial conditions to the equations of motion to obtain an initial trajectory $\mathbf{y}^{(0)}$ that is generated by linear interpolation between the initial and final states with a specific control. The initial guess for the time variable σ is obtained from GPOPS solution. Set $k = k + 1$.

2. For $k > 0$, parameterize a convex subproblem (i.e., Problem 2) using the solution from the previous iteration and solve this subproblem to find a solution pair $[\mathbf{y}^{(k)}, \mathbf{u}^{(k)}, \sigma^{(k)}]$ at the current step.

3. Check the convergence condition

$$\sup |\mathbf{y}^{(k)}(\tau) - \mathbf{y}^{(k-1)}(\tau)| \leq \varepsilon \tag{5.45}$$

   where ε is a preset tolerance vector and $\mathbf{y}(\tau)$ defined as the state values at time step τ. If the condition is satisfied, the algorithm moves to step 4; otherwise set $k = k + 1$ and go back to step 2.

4. The algorithm is converged and a solution for the problem is found to be $[\mathbf{y}^{(k)}, \mathbf{u}^{(k)}, \sigma^{(k)}]$.

5.3 Numerical Simulation

In this section, we use the EHang 184 eVTOL vehicle model for all the simulations as in Chapter 3. The vehicle parameters are listed in Table 5.1 below, and the initial and terminal merging conditions are shown in Table 5.2. For all the simulations we assume that the eVTOL aircraft operate on the main fly corridor with a constant altitude of 500 m and a fixed along-track velocity of 20 m/s. The merging vehicle starts with a cruising speed of 23 m/s in the along-track direction and 3 m/s in the vertical direction at an altitude of 400 m. We assume that the merging vehicle receives all the state information about the vehicle operating on the main corridor. All the simulations are performed on a desktop with a 64-bit operating system and an AMD Ryzen 7 1800X Eight-Core Processor.

For the first simulation, we consider the total thrust minimization problem. The GPOPS solver can successfully solve the problem within five seconds. Figures 2 and 3 below show the resulting position and speed profiles of the merging vehicle from the simulation. From the plots, we can observe that for the vehicle to save control effort, the merging vehicle will reduce the speed and merge after the cruising vehicle passes while maintaining a safe collision avoidance distance during the entire flight. Figure 4 below shows the result comparison between the GPOPS solver and SCP algorithm. We can see that for this scenario the two methods converged to different curves with very close final flight times, which indicates that there may exist multiple locally optimal solutions due to the nonconvexity of the problem. In Figs. 5 and 6, we compared the pitch angle profiles and the thrust profiles from both solvers. We can see that the SCP method converged to a smoother control profile than the GPOPS solver.

For the second simulation scenario, we optimize the final time of the merging vehicle. In this case, we expect the merging vehicle to fly as fast as possible to merge into the main corridor. The problem is solved by both SCP and GPOPS. The simulation results are shown in Figures 7-11. Aligned with our expectations, the merging eVTOL vehicle flies at a much higher speed than the previous task and merges in front of the cruising vehicle while

Table 5.1: Vehicle parameters for simulations

Parameter	Value
Vehicle's mass, m	240 kg
Reference front plate area, S_x	2.11 m^2
Reference top plate area, S_z	1.47 m^2
Drag coefficient, C_D	1
Atmospheric density, ρ	1.225 kg/m^3
Gravitational acceleration, g	9.81 m/s^2
Maximum along-track distance, x_{max}	2,000 m
Maximum altitude, z_{max}	500 m
Maximum airspeed, V_{max}	30 m/s
Maximum net thrust, T_{max}	4,800 N
Maximum rotor tip-path-plane pitch angle, θ_{max}	30 deg
Time of flight, t_f	10-200 s

Table 5.2: Initial and terminal conditions for merging vehicle

Parameter	Value
Initial along-track distance, x_0	0 m
Initial altitude, z_0	400 m
Initial along-track airspeed, V_{x0}	23 m/s
Initial vertical airspeed, V_{z0}	3 m/s
Terminal along-track distance, x_f	200 - 2,000 m
Terminal altitude, z_f	500 m
Terminal along-track airspeed, V_{xf}	20 m/s
Terminal vertical airspeed, V_{zf}	0 m/s
Collision avoidance distance, d	50 m

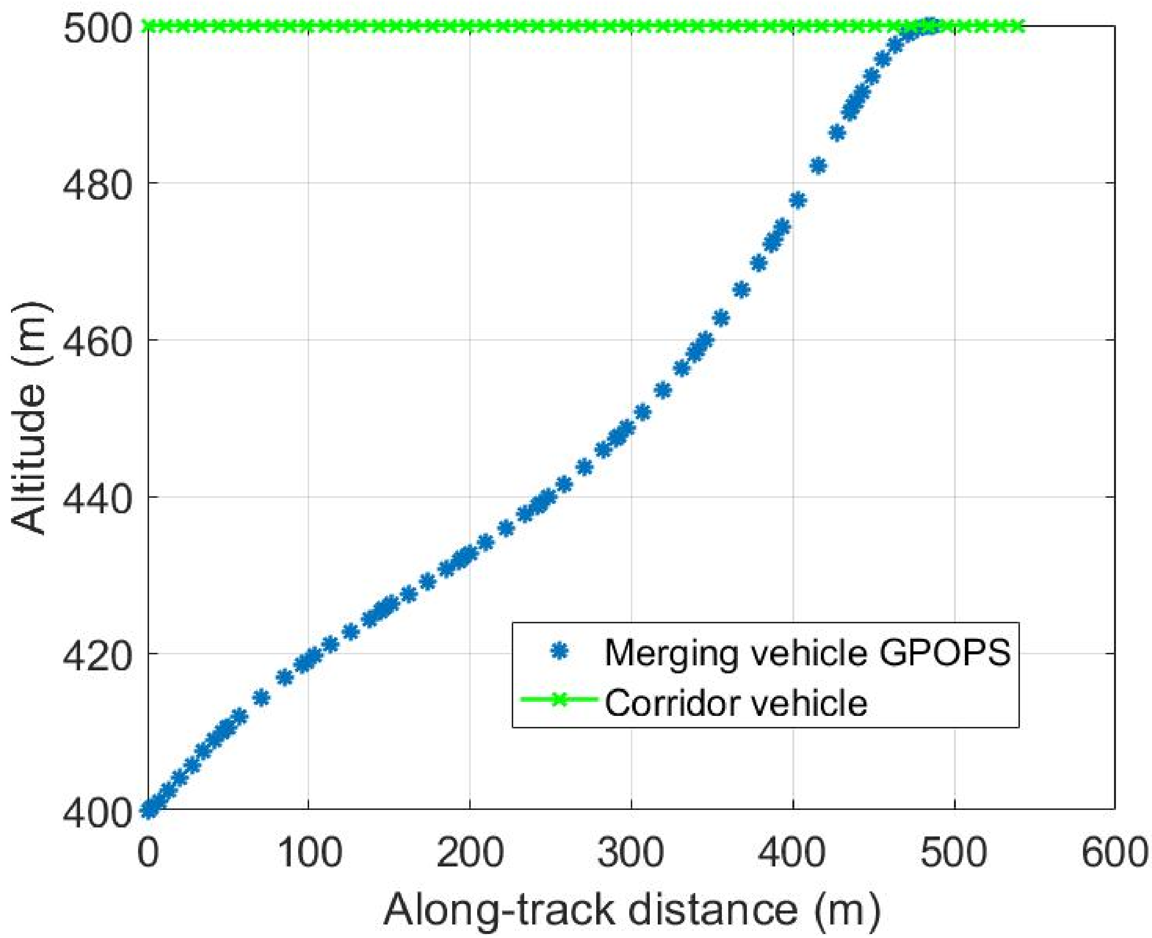


Figure 5.2: Trajectories of two merging vehicles for control-effort minimization.

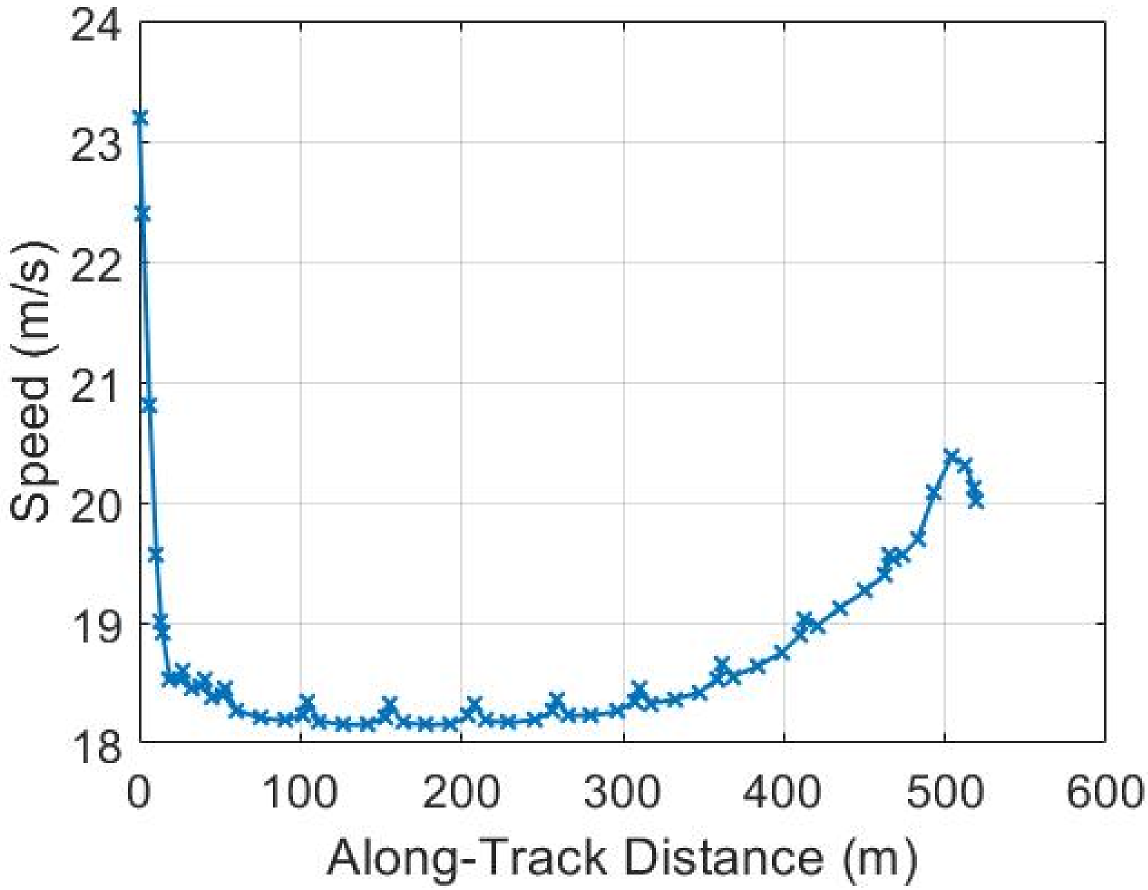


Figure 5.3: Merging vehicle speed profile by GPOPS for control-effort minimization.

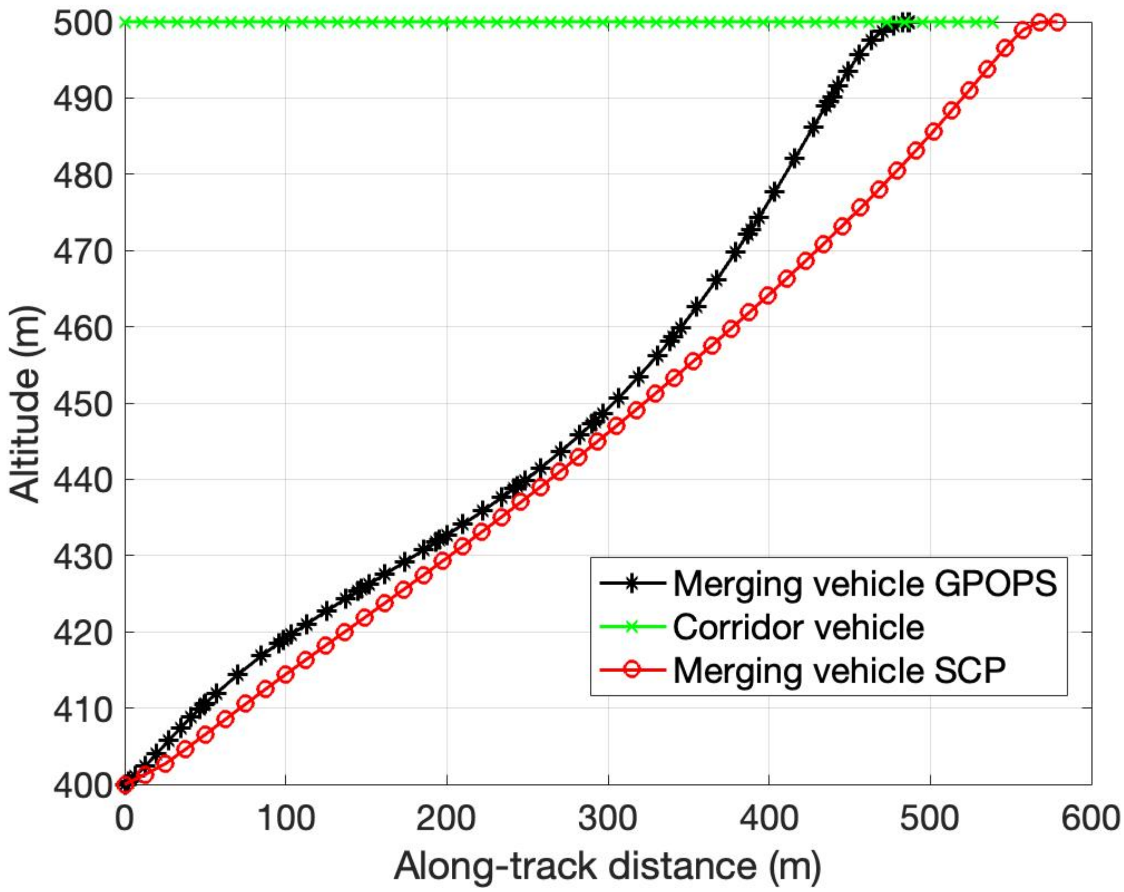


Figure 5.4: Comparison of trajectories between GPOPS and SCP for control-effort minimization.

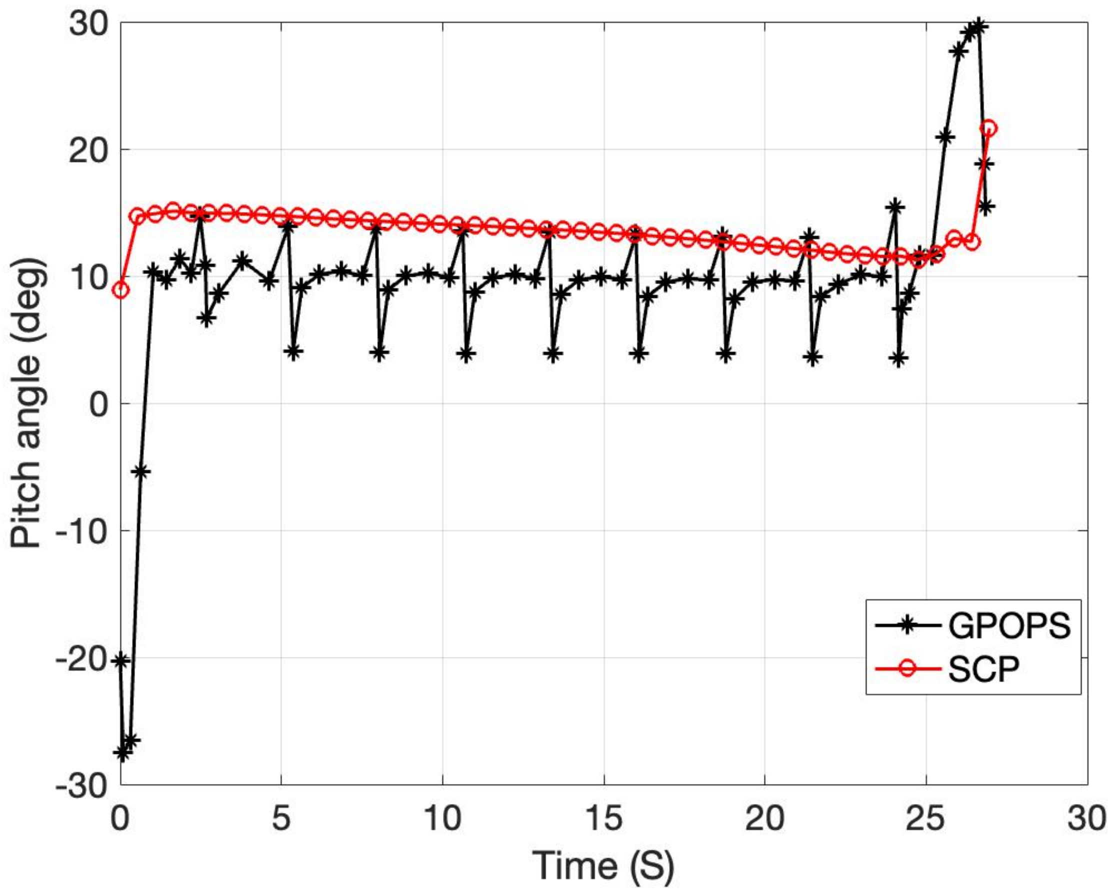


Figure 5.5: Comparison of pitch angle profiles between GPOPS and SCP for control-effort minimization.

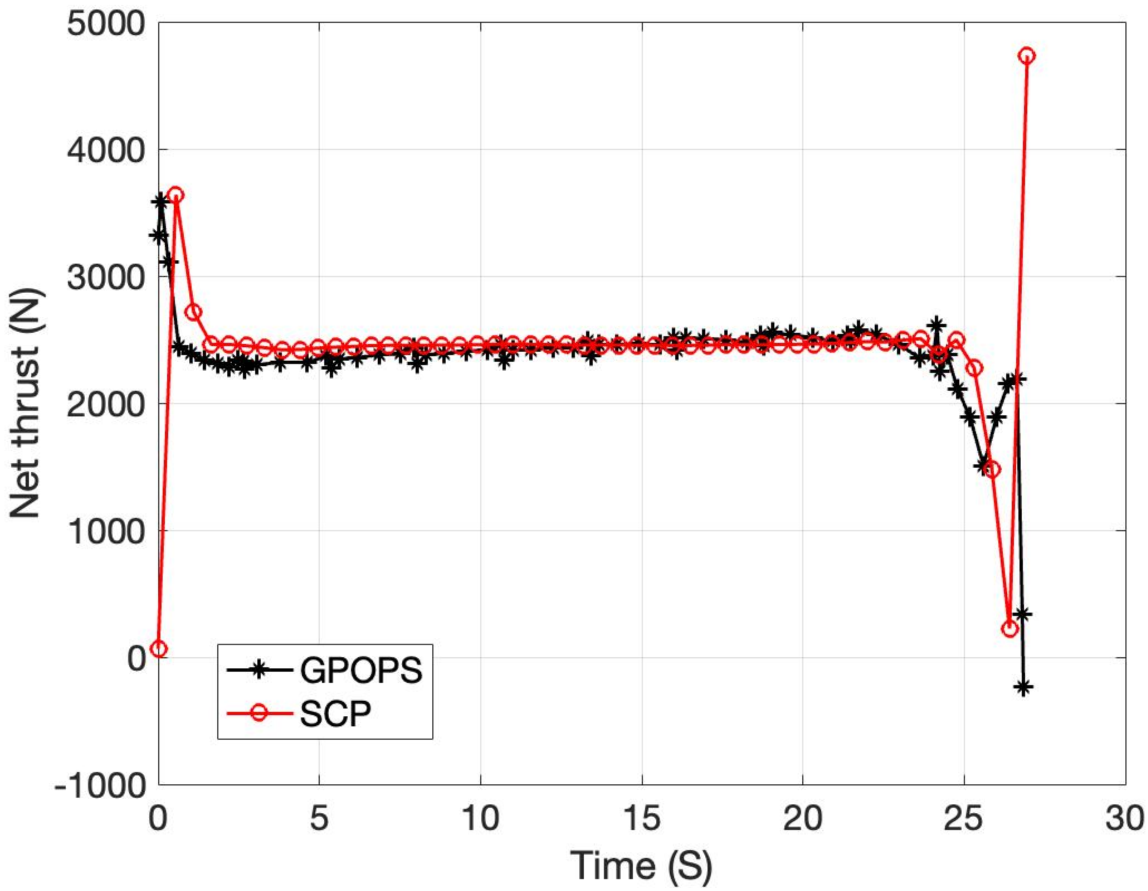


Figure 5.6: Comparison of thrust profiles between GPOPS and SCP for control-effort minimization.

satisfying all the predefined constraints. We can notice that both methods converge to a similar result. For the SCP method, it only took three iterations and less than one second to converge while the GPOPS solver took two seconds. Clearly, the SCP method converged to a smoother velocity profile. However, we observe some high-frequency jitters in the thrust curve that may be caused by the small number of collocation points we used to approximate the vehicle trajectory.

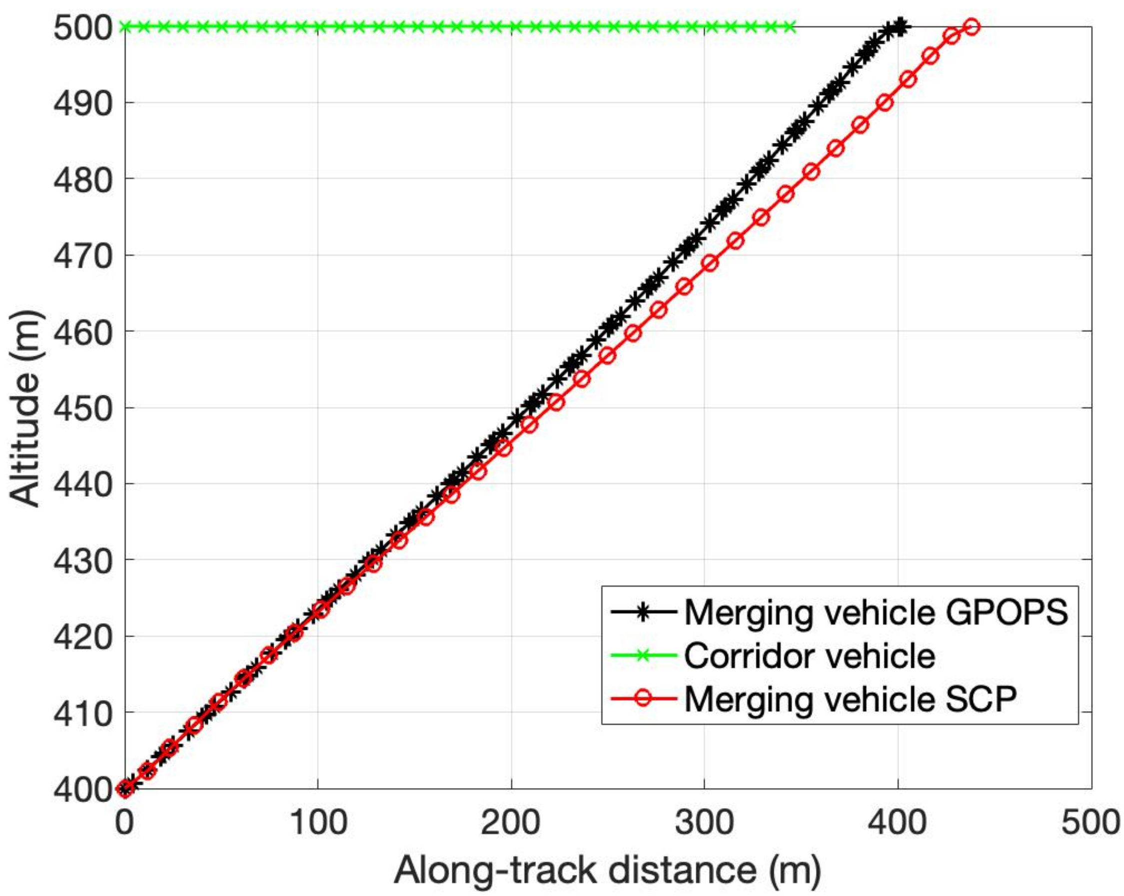


Figure 5.7: Comparison of merging trajectories from SCP and GPOPS for flight-time minimization.

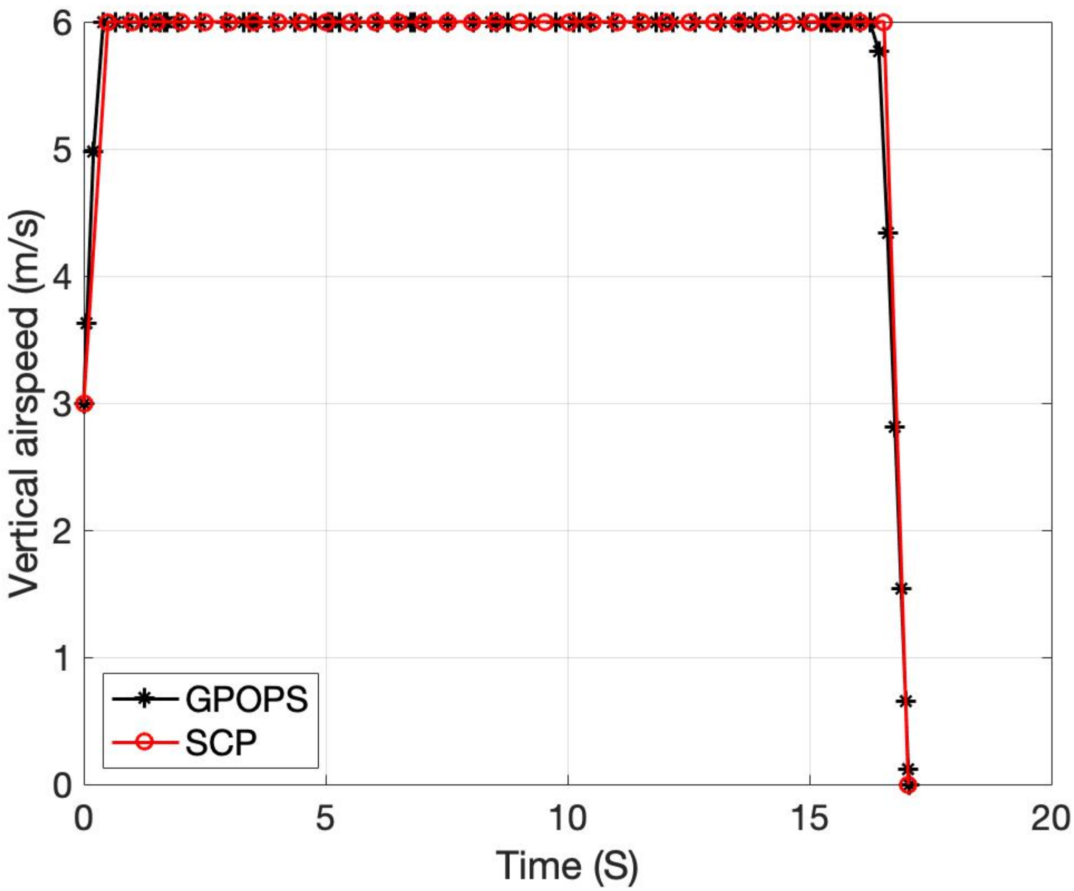


Figure 5.8: Comparison of vertical speed profiles from SCP and GPOPS for flight-time minimization.

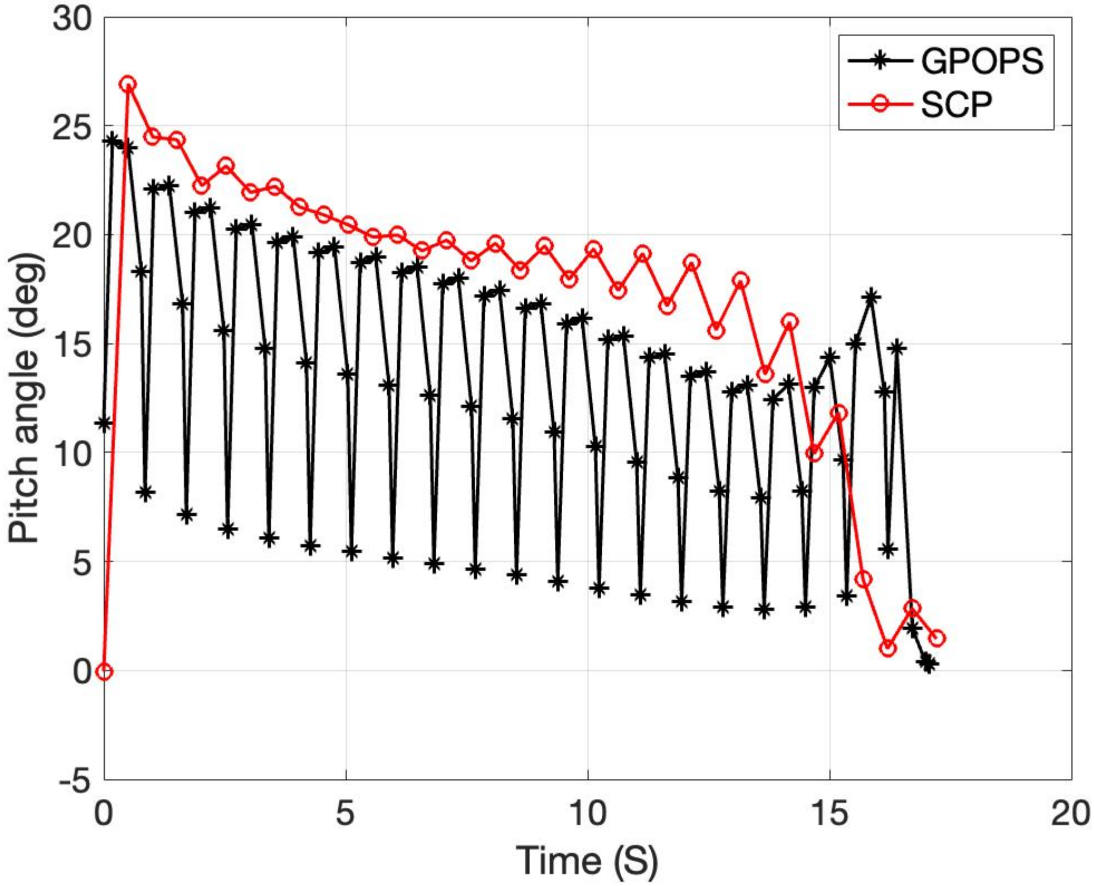


Figure 5.9: Comparison of pitch angle profiles from SCP and GPOPS for flight-time minimization.

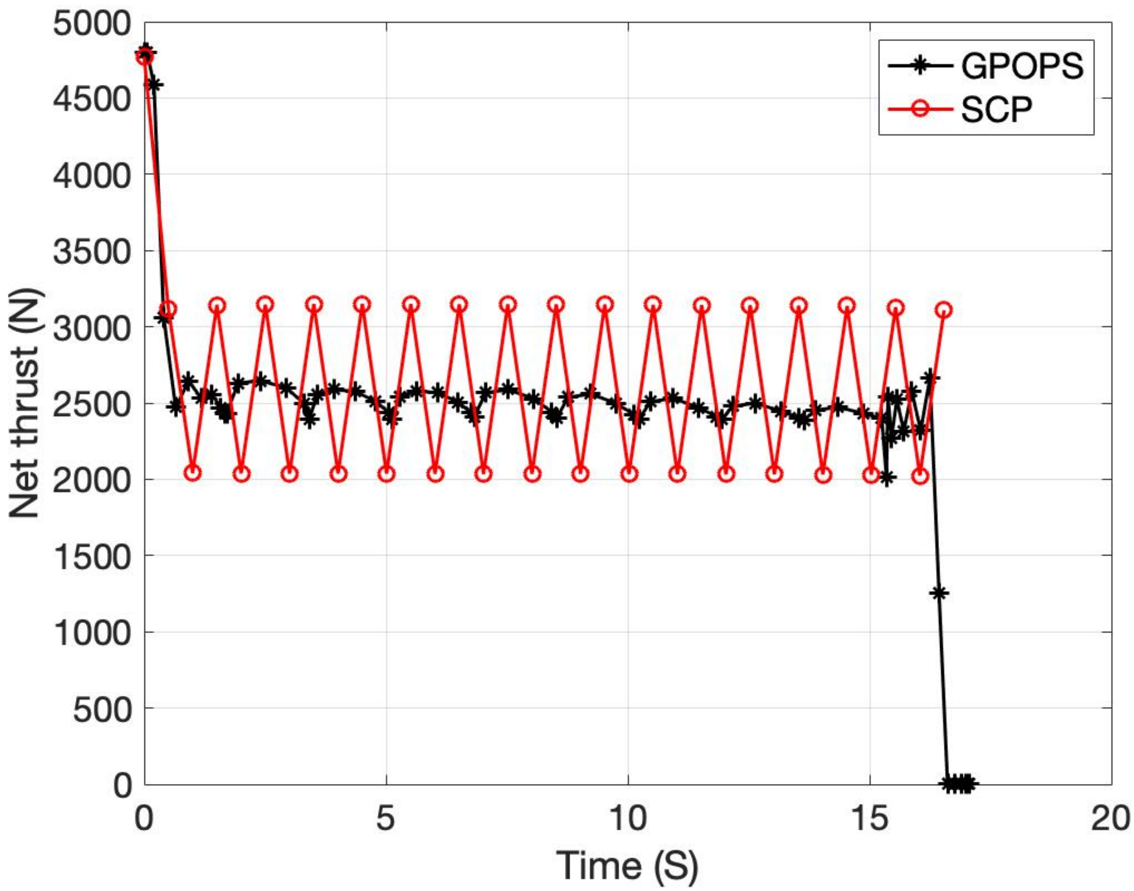


Figure 5.10: Comparison of thrust profiles from SCP and GPOPS for flight-time minimization.

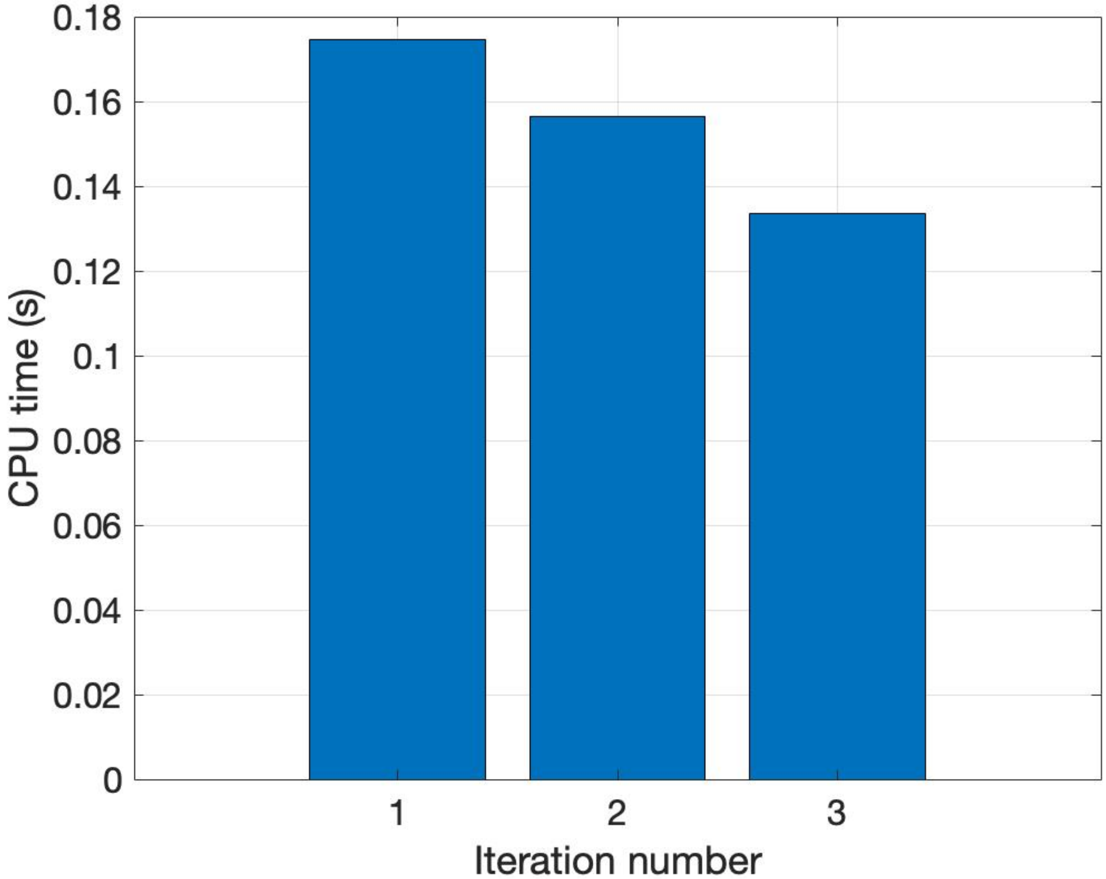


Figure 5.11: Time cost of SCP for flight-time minimization.

5.4 Summary

In summary, we proposed a decentralized optimal control approach to the merging coordination of eVTOL vehicles for urban air mobility. The nonconvex terms in the problem have been modified by different approaches and solved by the SCP method. Simulation results have proved that both the pseudospectral method and the sequential convex programming method can successfully obtain the optimization solution and have the potential to solve the problem in real time for onboard applications.

For future research, we aim to scale our approach from handling two vehicles to managing multiple vehicles in a merging scenario. To achieve this, we plan to adapt a rule-based approach that follows a 'First Come, First Serve' rule. This strategy will primarily focus on optimizing the trajectories of vehicles that are closest to the merging point, thereby simplifying the complexity involved in coordinating multiple vehicles simultaneously.

Furthermore, in the current research, I assumed that vehicles operating on the main airway fly at a constant speed. Moving forward, I plan to introduce random fluctuations in speed or adopt simple aerodynamic equations to consider the aerodynamic effects on the vehicles. This will allow for a more realistic representation of vehicle behavior under varying flight conditions and enhance the robustness of our merging strategy against aerodynamic uncertainties.

We also plan to refine our approach by incorporating a power model into our objective functions. Specifically, we will consider using a simple aerodynamic model that utilizes rotor speed as a control input. This adjustment will allow us to directly calculate the power consumption from the rotor speed, offering a more precise and practical measure for optimizing vehicle performance. This consideration will be discussed in more detail in the dissertation to explore its potential benefits and implementation strategies in air vehicle merging scenarios.

Chapter 6

Conclusion and Future Work

This dissertation has presented a thorough investigation into the optimization of eVTOL vehicle operations within the framework of urban air mobility, with a focus on trajectory optimization and merging coordination. This dissertation primarily focuses on utilizing convex approaches to solve advanced air mobility trajectory optimization problems in real-time for densely populated urban environments. From our simulation results, the SCP algorithm typically requires 4-8 iterations to converge. On our personal laptop, it usually takes about 0.5-1 seconds to perform each iteration. However, with proper optimization and execution using a more efficient language like C++, the computational time is expected to be reduced to less than 0.2 seconds per iteration. With these improvements, the autonomous system should be able to update its trajectory every second, thereby enabling real-time flight trajectory optimization. This performance enhancement will be critical for practical deployment in dynamic aerial environments where rapid response is essential.

The chapter 3 presents a multi-phase trajectory optimization problem, transitioning from cruise to landing phases, while the chapter 4 advances this discussion by incorporating high-fidelity models that consider aerodynamic effects. Throughout, we employed advanced optimization techniques, particularly the SCP method, to address complex problems often characterized by nonconvex terms. These methods have been rigorously tested through simulations, proving their efficacy in achieving optimal control of eVTOL vehicles, ensuring safety, and adhering to operational constraints.

Further, we proposed a decentralized optimal control approach for the merging coordination of multiple eVTOLs in chapter 5. This problem addresses the practical challenges of multiple aerial vehicles in shared airspace, a critical aspect as urban air mobility systems.

Overall, we proved that the SCP method can provide solutions that are viable for real-time, onboard applications. Looking ahead, several areas of potential research can further extend the contributions of this dissertation.

First of all, for the chapter 3, we will improve the convergence of our SCP method using various techniques such as the line-search and trust-region methods. More complicated CONOPs with higher-fidelity vehicle models will also be addressed in the future (maybe beyond this dissertation).

Second, for the chapter 4, we are now focused on enhancing the SCP algorithm to improve both the quality of solutions and computational speed. Moving forward, we will present more results from the convex optimization approach and conduct additional landing scenarios to further demonstrate the effectiveness of the proposed methods.

Lastly, for the chapter 5, we aim to scale our approach from handling two vehicles to managing multiple vehicles in a merging scenario. To achieve this, we plan to adapt a rule-based approach that follows a 'First Come, First Serve' rule. This strategy will primarily focus on optimizing the trajectories of vehicles that are closest to the merging point, thereby simplifying the complexity involved in coordinating multiple vehicles simultaneously. Furthermore, in the current research, we assumed that vehicles operating on the main airway fly at a constant speed. Moving forward, we plan to introduce random fluctuations in speed or adopt simple aerodynamic equations to consider the aerodynamic effects on the vehicles. This will allow for a more realistic representation of vehicle behavior under varying flight conditions and enhance the robustness of our merging strategy against aerodynamic uncertainties. We also plan to refine our approach by incorporating a power model into our objective functions. Specifically, we will consider using a simple aerodynamic model that utilizes rotor speed as a control input. This adjustment will allow us to directly calculate the power consumption from the rotor speed, offering a more precise and practical measure for optimizing vehicle performance.

Bibliography

[1] (2018). Ehang 184 Autonomous Aerial Vehicle (AAV). http://www.aerospace-technology.com/projects/ehang-184-autonomous-aerial-vehicle-aav. [Accessed: 17-May-2021]. 113

[2] Abbas, M. A., Milman, R., and Eklund, J. M. (2017). Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. *Canadian journal of electrical and computer engineering*, 40(1):12–22. 16

[3] Achterberg, T. (2019). What's new in gurobi 9.0. *Webinar Talk url: https://www. gurobi. com/wp-content/uploads/2019/12/Gurobi-90-Overview-Webinar-Slides-1. pdf*, 5(9):97–113. 21

[4] Acikmese, B. and Ploen, S. R. (2007). Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366. 21

[5] AlKaitoob, R., Mihaylova, L., and Papazov, E. (2022). The smart city vision: Cutting-edge business projects in dubai. *Real Estate Property & Business*, 6(4). 10

[6] Andersen, E. D. and Andersen, K. D. (2000). The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer. 21

[7] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E. (2020a). Autonomous upper stage guidance using convex optimization and model predictive control. In *ASCEND 2020*, page 4268. 21

[8] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E. (2020b). Convex optimization of launch vehicle ascent trajectory with heat-flux and splash-down constraints. In *AAS/AIAA Astrodynamics Specialist Conference*, Virtual Lake Tahoe. 21

[9] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E. (2021). Convex approach to three-dimensional launch vehicle ascent trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 44(6):1116–1131. 21, 27

[10] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E. (2022). Convex optimization of launch vehicle ascent trajectory with heat-flux and splash-down constraints. *Journal of Spacecraft and Rockets*, 59(3):900–915. 27, 36

[11] Bosson, C. and Lauderdale, T. A. (2018). Simulation evaluations of an autonomous urban air mobility network management and separation service. In *2018 Aviation Technology, Integration, and Operations Conference*, page 3365. 5

[12] Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press. 3, 20, 40

[13] Brittain, M. and Wei, P. (2019). Autonomous air traffic controller: A deep multi-agent reinforcement learning approach. *arXiv preprint arXiv:1905.01303*. 5

[14] Brittain, M. W., Yang, X., and Wei, P. (2021). Autonomous separation assurance with deep multi-agent reinforcement learning. *Journal of Aerospace Information Systems*, 18(12):890–905. 15

[15] Cao, P., Hwang, J. T., Bewley, T., and Kuester, F. (2022). Mission-oriented trajectory optimization for search-and-rescue multirotor uavs in cluttered and gps-denied environments. In *AIAA AVIATION 2022 Forum*, page 3999. 8, 75

[16] Chauhan, S. S. and Martins, J. R. (2020). Tilt-wing evtol takeoff trajectory optimization. *Journal of aircraft*, 57(1):93–112. 16

[17] Darby, C. L., Hager, W. W., and Rao, A. V. (2011). An hp-adaptive pseudospectral method for solving optimal control problems. *Optimal Control Applications and Methods*, 32(4):476–502. 24, 118

[18] Deniz, S. and Wang, Z. (2022). A multi-agent reinforcement learning approach to traffic control at future urban air mobility intersections. In *AIAA SCITECH 2022 Forum*, page 1509. 70

[19] Deniz, S., Wu, Y., Shi, Y., and Wang, Z. (2022). A multi-agent reinforcement learning approach to traffic control at merging point of urban air mobility. In *AIAA AVIATION 2022 Forum*, page 3912. 70

[20] Deniz, S., Wu, Y., Shi, Y., and Wang, Z. (2023). Autonomous landing of evtol vehicles via deep q-networks. In *AIAA AVIATION 2023 Forum*. 70

[21] Domahidi, A., Chu, E., and Boyd, S. (2013). Ecos: An socp solver for embedded systems. In *2013 European Control Conference (ECC)*, pages 3071–3076. IEEE. 42

[22] Duchi, J. (2018). Sequential Convex Programming. `https://stanford.edu/class/ee364b/lectures/seq_notes.pdf`. 21

[23] Elevate, U. (2016). Fast-forwarding to the future of on-demand, urban air transportation [r/ol]. 7

[24] Erzberger, H. (2005). *Automated conflict resolution for air traffic control*. [National Aeronautics and Space Administration], Ames Research Center. 5

[25] Falcone, P., Tufo, M., Borrelli, F., Asgari, J., and Tseng, H. E. (2007). A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *2007 46th IEEE Conference on Decision and Control*, pages 2980–2985. IEEE. 16

[26] Garrow, L. A., German, B. J., and Leonard, C. E. (2021). Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research. *Transportation Research Part C: Emerging Technologies*, 132:103377. 1

[27] Goyal, R., Reiche, C., Fernando, C., Serrao, J., Kimmel, S., Cohen, A., and Shaheen, S. (2018). Urban air mobility (uam) market study. Technical report. 9

[28] Grant, M., Boyd, S., and Ye, Y. (2009). Cvx users' guide. *online: http://www. stanford. edu/boyd/software. html*. 21

[29] Gutierrez, J., Neogi, N. A., Kaeli, D., and Dill, E. T. (2022). A high-performance computing gnss-aware path planning algorithm for safe urban flight operations. In *AIAA AVIATION 2022 Forum*, page 3461. 3

[30] Hill, B. P., DeCarme, D., Metcalfe, M., Griffin, C., Wiggins, S., Metts, C., Bastedo, B., Patterson, M. D., and Mendonca, N. L. (2020). Uam vision concept of operations (conops) uam maturity level (uml) 4. 9, 10, 11

[31] Jiang, Z. and Pakmehr, M. (2022). Model predictive control for distributed electric propulsion of evtol vehicles: A preliminary design. In *AIAA SCITECH 2022 Forum*, page 0878. 16

[32] Kim, J.-H., Sukkarieh, S., and Wishart, S. (2003). Real-time navigation, guidance, and control of a uav using low-cost sensors. In *Field and Service Robotics*, pages 299–309. Springer. 15

[33] KIRKLAND, W. (2021). Inrix global traffic scorecard. 7

[34] Kleinbekman, I. C., Mitici, M., and Wei, P. (2020). Rolling-horizon electric vertical takeoff and landing arrival scheduling for on-demand urban air mobility. *Journal of Aerospace Information Systems*, 17(3):150–159. 13

[35] Kleinbekman, I. C., Mitici, M. A., and Wei, P. (2018). evtol arrival sequencing and scheduling for on-demand urban air mobility. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–7. IEEE. 13, 15

[36] Konnai, R., Scherer, R. C., Peplinski, A., and Ryan, K. (2017). Whisper and phonation: Aerodynamic comparisons across adduction and loudness. *Journal of Voice*, 31(6):773–e11. 10

[37] Leishman, G. J. (2006). *Principles of helicopter aerodynamics with CD extra*. Cambridge university press. 72

[38] Lenhart, S. and Workman, J. T. (2007). *Optimal control applied to biological models*. Chapman and Hall/CRC. 16

[39] Liu, C. H., Chen, Z., Tang, J., Xu, J., and Piao, C. (2018). Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, 36(9):2059–2070. 4

[40] Liu, X. and Lu, P. (2014). Solving nonconvex optimal control problems by convex optimization. *Journal of Guidance, Control, and Dynamics*, 37(3):750–765. 20

[41] Liu, X., Lu, P., and Pan, B. (2017). Survey of convex optimization for aerospace applications. *Astrodynamics*, 1(1):23–40. 20

[42] Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE. 42

[43] Manthey, S., González, M., and Terzidis, O. (2024). Volocopter (germany). In *Cases on STEM Entrepreneurship*, pages 87–101. Edward Elgar Publishing. 13

[44] Marconi, L., Isidori, A., and Serrani, A. (2002). Autonomous vertical landing on an oscillating platform: an internal-model based approach. *Automatica*, 38(1):21–32. 5, 70

[45] Marshall, J. A., Sun, W., and L'Afflitto, A. (2021). A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems. *Annual Reviews in Control*, 52:390–427. 15

[46] Marzouk, O. A. (2022). Urban air mobility and flying cars: Overview, examples, prospects, drawbacks, and solutions. *Open Engineering*, 12(1):662–679. 11

[47] McDonald, L., Wu, Y., Deniz, S., and Wang, Z. (2022). Real-time generation of comfort-optimal flight trajectories for urban air mobility missions. In *AIAA SCITECH 2022 Forum*, page 2157. 70

[48] McDonald, S. T. and Wang, Z. (Dallas, Texas, 2019). Real-time optimal trajectory generation for uav to rendezvous with an aerial orbit. In *AIAA Aviation Forum*. 21

[49] Mueller, M. W. and D'Andrea, R. (2014). Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 45–52. IEEE. 13

[50] Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media. 30, 32

[51] Ngo, T. D. and Sultan, C. (2016). Model predictive control for helicopter shipboard operations in the ship airwakes. *Journal of Guidance, Control, and Dynamics*, 39(3):574–589. 5, 70

[52] Oleksak, K., Wu, Y., Abella, M., Wang, Z., and Gan, H. (2021). Trajectory optimization of unmanned aerial vehicles for wireless communication with ground terminals. In *AIAA Scitech 2021 Forum*, page 0709. 6, 21

[53] Ormiston, R. A. (2016). Revitalizing advanced rotorcraft research—and the compound helicopter: 35th ahs alexander a. nikolsky honorary lecture. *Journal of the American Helicopter Society*, 61(1):1–23. 11

[54] Patterson, M. (2021). Advanced air mobility (aam): An overview and brief history. In *Transportation Engineering and Safety Conference*. 1

[55] Patterson, M. A. and Rao, A. V. (2014). Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1–37. 19, 89, 117

[56] Pitt, D. M. (1980). *Rotor dynamic inflow derivatives and time constants from various inflow models*. Washington University in St. Louis. 75

[57] Polvara, R., Patacchiola, M., Sharma, S., Wan, J., Manning, A., Sutton, R., and Cangelosi, A. (2017). Autonomous quadrotor landing using deep reinforcement learning. *arXiv preprint arXiv:1709.03339*. 70

[58] Pradeep, P., Lauderdale, T. A., Chatterji, G. B., Sheth, K., Lai, C. F., Sridhar, B., Edholm, K.-M., and Erzberger, H. (2020). Wind-optimal trajectories for multirotor evtol aircraft on uam missions. In *AIAA AVIATION 2020 FORUM*, page 3271. 16

[59] Pradeep, P. and Wei, P. (2019). Energy-efficient arrival with rta constraint for multirotor evtol in urban air mobility. *Journal of Aerospace Information Systems*, 16(7):263–277. 4, 13, 15, 25, 70, 113

[60] Prevot, T., Rios, J., Kopardekar, P., Robinson III, J. E., Johnson, M., and Jung, J. (2016). Uas traffic management (utm) concept of operations to safely enable low altitude flight operations. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, page 3292. 7

[61] Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528. 16, 18

[62] Rao, A. V. (2014). Trajectory optimization: a survey. In *Optimization and optimal control in automotive systems*, pages 3–21. Springer. 3

[63] Renegar, J. (2001). *A mathematical view of interior-point methods in convex optimization*. SIAM. 21

[64] Ross, I. M. and Karpenko, M. (2012). A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182–197. 19

[65] Sanfourche, M., Delaune, J., Le Besnerais, G., De Plinval, H., Israel, J., Cornic, P., Treil, A., Watanabe, Y., and Plyer, A. (2012). Perception for uav: Vision-based navigation and environment modeling. *Aerospace Lab*, (4):p–1. 15

[66] Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. In *2001 European control conference (ECC)*, pages 2603–2608. IEEE. 5

[67] Smedt, D. d., Bronsvoort, J., and McDonald, G. (2013). Controlled time of arrival feasibility analysis. 4

[68] Spencer, C. T. (2024). Development, modeling, identification, and control of tilt-rotor evtol aircraft. 13

[69] Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K.-D., Kaiser, J., and Plötner, K. O. (2020). An overview of current research and developments in urban air mobility–setting the scene for uam introduction. *Journal of Air Transport Management*, 87:101852. 2, 7

[70] Sun, S., Cioffi, G., de Visser, C., and Scaramuzza, D. (2021). Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events. *IEEE Robotics and Automation Letters*, 6(2):580–587. 13

[71] Sussmann, H. J. and Willems, J. C. (1997). 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, 17(3):32–44. 16

[72] Szmuk, M. and Acikmese, B. (2018). Successive convexification for 6-dof mars rocket powered landing with free-final-time. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 0617. 21

[73] Thipphavong, D. P., Apaza, R., Barmore, B., Battiste, V., Burian, B., Dao, Q., Feary, M., Go, S., Goodrich, K. H., Homola, J., et al. (2018). Urban air mobility airspace integration concepts and considerations. In *2018 Aviation Technology, Integration, and Operations Conference*, page 3676. 9, 13

[74] Topputo, F. and Zhang, C. (2014). Survey of direct transcription for low-thrust space trajectory optimization with applications. In *Abstract and Applied Analysis*, volume 2014. Hindawi. 20

[75] Vascik, P. D. (2017). *Systems-level analysis of On Demand Mobility for aviation*. PhD thesis, Massachusetts Institute of Technology. 1

[76] Von Stryk, O. and Bulirsch, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37(1):357–373. 3, 18

[77] Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57. 18

[78] Wang, M., Diepolder, J., Zhang, S., Söpper, M., and Holzapfel, F. (2021a). Trajectory optimization-based maneuverability assessment of evtol aircraft. *Aerospace Science and Technology*, 117:106903. 8

[79] Wang, Z. (2024). A survey on convex optimization for guidance and control of vehicular systems. *Annual Reviews in Control*, 57:100957. 20

[80] Wang, Z. and Grant, M. J. (2017). Constrained trajectory optimization for planetary entry via sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 40(10):2603–2615. 6, 38, 119

[81] Wang, Z. and Lu, Y. (2020a). Improved sequential convex programming algorithms for entry trajectory optimization. *Journal of Spacecraft and Rockets*, 57(6):1373–1386. 21

[82] Wang, Z. and Lu, Y. (2020b). Improved sequential convex programming algorithms for entry trajectory optimization. *Journal of Spacecraft and Rockets*, 57(6):1373–1386. 34

[83] Wang, Z. and McDonald, S. T. (2020). Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles. *Aerospace Science and Technology*, 99. 21

[84] Wang, Z. and McDonald, S. T. (Dallas, Texas, 2019). Optimal rendezvous of unmanned aerial and ground vehicles via sequential convex programming. In *AIAA Aviation Forum*. 21

[85] Wang, Z., Wei, P., and Sun, L. (2021b). Optimal cruise, descent, and landing of evtol vehicles for urban air mobility using convex optimization. In *AIAA SciTech 2021 Forum*, page 0577. 6, 23, 28, 70

[86] Wang, Z., Wu, Y., and Huang, D. (2023). Optimal landing control of evtol vehicles using ode-based aerodynamic model. In *AIAA SCITECH 2023 Forum*, page 1742. 70

[87] Wu, Y., Deniz, S., Shi, Y., and Wang, Z. (2022a). A convex optimization approach to real-time merging control of evtol vehicles for future urban air mobility. In *AIAA AVIATION 2022 Forum*, page 3319. 70

[88] Wu, Y., Deniz, S., Shi, Y., Wang, Z., and Huang, D. (2023). Precision landing trajectory optimization for evtol vehicles with high-fidelity aerodynamic models. In *AIAA AVIATION 2023 Forum*, page 3409. 70

[89] Wu, Y., Wang, Z., Benedikter, B., and Zavoli, A. (2022b). A convex approach to multiphase trajectory optimization of evtol vehicles for urban air mobility. In *AIAA SCITECH 2022 Forum*, page 2159. 4, 16, 24

[90] Wu, Y., Wang, Z., Benedikter, B., and Zavoli, A. (2022c). A convex approach to multiphase trajectory optimization of evtol vehicles for urban air mobility. In *AIAA SCITECH 2022 Forum*, page 2159. 70

[91] Xiang, S., Xie, A., Ye, M., Yan, X., Han, X., Niu, H., Li, Q., and Huang, H. (2023). Autonomous evtol: A summary of researches and challenges. *Green Energy and Intelligent Transportation*, page 100140. 8

[92] Yomchinda, T., Horn, J., and Langelaan, J. (2011). Flight path planning for descent-phase helicopter autorotation. In *AIAA Guidance, Navigation, and Control Conference*, page 6601. 25

[93] Zhang, S., Li, Y., and Dong, Q. (2022). Autonomous navigation of uav in multi-obstacle environments based on a deep reinforcement learning approach. *Applied Soft Computing*, 115:108194. 3

[94] Zhao, D.-J. and Song, Z.-Y. (2017). Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. *Acta Astronautica*, 137:60–69. 27

Vita

Yufei Wu was born and raised in Beijing, China. After earning his bachelor's degree in Mechanical Engineering from the University of California Davis, Yufei decided to persuade on a career in mechanical and autonomous systems engineering. Motivated by a relentless pursuit of knowledge and a desire to push the boundaries of current technological capabilities, Yufei subsequently enrolled in the Master of Science program in Mechanical Engineering at Stevens Institute of Technology, where his research focused on wearable robotics—a pivotal area at the intersection of biomechanics and technology. Throughout his master's program, Yufei dedicated himself to intense study and research, culminating in significant contributions to the field of smart footwear technology through the use of advanced machine learning models.

Building on the robust foundation established during his master's studies, Yufei decided to advance his research and professional career by pursuing a Ph.D. in Mechanical Engineering at the University of Tennessee Knoxville. Under the guidance of his advisor Zhenbo Wang and through numerous challenges inherent in doctoral research, Yufei has been exploring and expanding the boundaries of autonomous systems and control technologies. His work has notably advanced the field of trajectory optimization for autonomous drones and eVTOL vehicles, utilizing sophisticated convex optimization techniques to enable real-time operational capabilities. Yufei's hard work paid off when he was awarded Tennessee Fellowship for Graduate Excellence. He is incredibly grateful for all the support from his parents, advisor as he begins her new career.