

Convex Approach to Real-Time Multiphase Trajectory Optimization for Urban Air Mobility

Yufei Wu,^{*} Sabrullah Deniz,[†] Yang Shi,[‡] and Zhenbo Wang[§] 

University of Tennessee, Knoxville, Tennessee 37996

<https://doi.org/10.2514/1.D0348>

Electric vertical takeoff and landing technology has emerged as a promising solution to alleviate ground transportation congestion. However, the limited capacity of onboard batteries enforces hard time constraints for vehicle operations in a complex urban environment. Therefore, a constrained, real-time trajectory optimization method is necessary to enable safe and energy-efficient vehicle flight. Unfortunately, most existing trajectory optimization methods suffer from low computational efficiency, unpredictable convergence processes, and strong dependence on a good initial trajectory. In this paper, we employ a successive convex programming approach to solve the multiphase minimum-energy-cost electric vertical takeoff and landing vehicle trajectory optimization problem for urban air mobility applications. The main contribution is the development and validation of two novel convexification methods that find approximate optimal solutions to the multiphase nonlinear trajectory optimization problem in real time by solving a sequence of convex subproblems. Specifically, the first method transforms the original nonlinear problem into a sequence of second-order cone programming problems through a convenient change of variables and lossless convexification, while the second approach achieves a similar goal without the need for control convexification. The resulting convex subproblems can be solved reliably in real time by advanced interior point methods. The proposed methods are demonstrated through numerical simulations of two different urban air mobility scenarios and compared with the solution obtained from GPOPS-II, a state-of-the-art general-purpose optimal control solver. Our proposed sequential convex programming methods can obtain near-optimal solutions with faster computational speed than GPOPS-II.

Nomenclature

C_D	=	drag coefficient
D	=	drag force, N
g	=	gravitational acceleration, m/s ²
m	=	vehicle's mass, kg
N	=	number of phase node
p	=	phase number
S_x	=	reference front plate area, m ²
S_z	=	reference top plate area, m ²
T	=	net thrust, N
t_f	=	time of flight, s
u_1, u_2, u_3	=	new control variables
V_x	=	along-track airspeed, m/s
V_z	=	vertical airspeed, m/s
x	=	along-track distance, m
y	=	state vector
z	=	altitude, m
δ	=	trust-region radius
ϵ	=	tolerance
θ	=	rotor tip-path-plane pitch angle, deg
ρ	=	atmospheric density, kg/m ³
σ	=	free-time variable

I. Introduction

TODAY, the world's population has more than doubled since the 1960s, which has generated a heavy demand on surface transportation. In 2019, Americans spent an average of more than 99 h each year in road traffic, and in the most traffic-congested city, Bogota, Colombia, every driver on average wastes more than 191 h every year in traffic [1]. So, people have begun to reimagine and explore advanced air transportation ideas with the aim of creating novel solutions to surface traffic congestion problems. Fortunately, with the rapid development of battery and aviation technologies, electric vertical takeoff and landing (eVTOL) aircraft are becoming a promising alternative solution to road-traffic congestion problems by utilizing the three-dimensional (3D) urban space to better meet the mobility demand. In 2020, NASA published a report on the urban air mobility (UAM) vision concept of operations [2], where the UAM research is divided into five main areas, including aircraft development, airspace design, individual aircraft operations, airspace operations management, and community integration. In this paper, we focus on the safe and efficient operations of individual eVTOL aircraft.

Compared to traditional rotorcraft, such as helicopters and multi-copters, eVTOL vehicles are potentially more energy-efficient, flexible, and generate much less noise within the operating environment [3]. In spite of the challenges in design, manufacturing, operational management, and flight control, eVTOL vehicles are highly desirable under the envisioned UAM scenarios. However, due to the limits of current technology and the complicated infrastructure development process, eVTOL operations suffer from limited battery endurance and vertiport capacity during early UAM operations [4–6].

The main idea behind the UAM concept is to mitigate ground traffic congestion by taking advantage of the 3D National Airspace System (NAS) with minimal additional workload and burden on air traffic controllers. Therefore, to fulfill the limited vertiport capacity and limited battery life, the eVTOL operations should follow a four-dimensional (4D) trajectory, which includes the 3D trajectory parameterized by time. Also, a time of arrival (RTA) constraint is usually required and enforced for a given mission scenario under a specific flight schedule [7].

A general autonomous flight control system consists of several subsystems, including navigation and perception, mission planning,

Presented as Paper 2022-2159 at the AIAA SciTech Forum, San Diego, CA, and Virtual, January 3–7, 2022; received 5 October 2022; accepted for publication 23 June 2024; published online XX epubMonth XXXX. Copyright © 2024 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2380-9450 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

^{*}Ph.D. Candidate, Department of Mechanical, Aerospace, and Biomedical Engineering; ywu86@vols.utk.edu. Student Member AIAA.

[†]Graduate Student, Department of Mechanical, Aerospace, and Biomedical Engineering; sdeniz@vols.utk.edu. Student Member AIAA.

[‡]Ph.D. Candidate, Department of Mechanical, Aerospace, and Biomedical Engineering; yshi38@vols.utk.edu. Student Member AIAA.

[§]Assistant Professor, Department of Mechanical, Aerospace, and Biomedical Engineering; zwang124@utk.edu. Senior Member AIAA.

tactical planning, trajectory optimization, feedback control, and actuator commands [8]. First, navigation and perception data are obtained and processed from different sensors. Second, users define the mission goals, flight constraints, and possible waypoints for the flight mission in the mission planning section. Third, the system needs to update the detailed mission goals with the real-world mission environment in the tactical planning section. Finally, 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 [9], eVTOL merging control [10], and wind-optimal trajectory planning [11]. Based on the outcomes of trajectory optimization, feedback control is used to calculate the real-time control signal, considering real-world disturbances such as wind gusts, sensor errors, and other environmental factors.

In this work, we focus on trajectory optimization, one of the key modules to enable safe and efficient eVTOL operations. Trajectory optimization problems are usually solved by two main methods: indirect and direct methods [12]. The indirect methods leverage the calculus of variations or Pontryagin's minimum principle to determine the first-order optimality conditions for the problem and then transfer the original problem to a two-point (or multipoint) boundary value problem. 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 difficult to obtain for 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. 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 [13], 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 sensitive to the initial guess, the parameter settings, and the nonlinearity and complexity of the problem. Second, the algorithm is not guaranteed 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 UAM applications, where flight safety and performance are of vital importance.

It is worth noting that the trajectory optimization program addressed in this paper is different from other path planning programs such as the traveling salesman problem (TSP) [14]. One of the key differences lies in the incorporation of nonlinear flight dynamic systems (i.e., translational equations of motion) in our considered trajectory optimization problem. In addition, the dynamic path constraints, control constraints, and speed limit have been traditionally ignored in path planning problems but can be included in our problem formulation. As such, a continuous-time optimal control problem, instead of a discrete static mathematical programming problem, is formulated and solved in this work, although the continuous-time problem is eventually converted into and solved as a discrete-time numerical optimization problem by convex optimization.

Over the last few years, convex optimization, which studies the problem of minimizing a convex function over a convex set, 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 [15]. If the 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 because of its low

complexity and convex structure [16]. As such, convex optimization approaches have great potential for onboard, real-time applications.

Nevertheless, two big challenges for 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 to a sequence of convex subproblems [17]. For the SCP method, the convex terms in the problem remain unchanged, and only the nonconvex components are convexified by convex approximations or relaxations. Generally, the nonconvex constraints in an optimal control problem can be convexified by introducing slack variables and constraint relaxations, and the highly nonlinear dynamic equations are commonly replaced by an approximate formula obtained from first-order Taylor series expansions. Then, the solution can be obtained numerically by using interior-point methods. The SCP method has been studied in the aerospace field, and example applications include powered descent guidance for Mars landing [18,19], orbital transfers [20,21], entry guidance and trajectory optimization [22–24], launch trajectory optimization [25,26], and low-speed air and ground vehicles [27–29]. In our previous work [30], we formulated a single-phase, cruise-to-landing eVTOL trajectory optimization problem and solved it using a convex optimization approach that was shown 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 UAM missions involve multiple flight phases, including takeoff, climb, cruise, hover, descent, approach, and landing. Building on our preliminary work in [31], we develop a novel convex optimization approach in this paper to obtain numerical solutions to the minimum-energy-cost, multiphase eVTOL trajectory optimization problem. First, the problem is formulated as a general nonlinear trajectory optimization with a fixed RTA constraint. Then, the nonlinearity in the flight dynamics is reduced via a convenient change of variables, 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. To address the possible breakdown of the underlying assumptions for exact convexification, an alternative approach is devised without the need for control constraint relaxation. Two different eVTOL cruise-to-landing scenarios are solved with the proposed methods. 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 [32].

The rest of this paper is organized as follows: A two-phase eVTOL trajectory optimization problem is formulated in Sec. II based on the EHang 184 eVTOL vehicle. In Sec. III, we provide two convexification procedures to reformulate the original problem into a convex problem and develop an SCP algorithm to solve the problem. In Sec. IV, the numerical results of two examples are presented to demonstrate the performance of the proposed methods. Lastly, we summarize the work in Sec. V.

II. Problem Formulation

In this section, we present the details of the problem formulation. A general optimal control problem formulation consists of a few components, including a dynamic system that governs the motion of the vehicle, the physical constraints to be satisfied, and a performance criterion to be optimized. The dynamic system utilized in this paper is developed based on a quad-rotor eVTOL aircraft and is described in Sec. II.A. Following that, the flight constraints are discussed in Sec. II.B. Finally, the objective function and the overall trajectory optimization problem are presented in Sec. II.C.

A. System Dynamics

In this paper, the dynamic model of the eVTOL vehicle is adapted from [31]. The vehicle is assumed to take a geodesic path, and only the longitudinal motion of the vehicle is modeled to formulate a

two-dimensional (2D) trajectory optimization problem. The flight dynamic equations are as follows:

$$\dot{x} = V_x \quad (1)$$

$$\dot{z} = V_z \quad (2)$$

$$\dot{V}_x = \frac{T \sin \theta}{m} - \frac{D_x}{m} \quad (3)$$

$$\dot{V}_z = \frac{T \cos \theta}{m} - \frac{D_z}{m} - g \quad (4)$$

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 $\mathbf{y}$ and the control vector $\mathbf{u}$ for the system are defined as

$$\mathbf{y} = [x, z, V_x, V_z]^T \quad \text{and} \quad \mathbf{u} = [T, \theta]^T \quad (5)$$

We assume that the maximum ground speed of the eVTOL vehicle is equal to 100 km/h. Thus, the drag force on the fuselage of the aircraft can be calculated from 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 [7,30] as follows:

$$D_x = \frac{\rho V_x^2 C_D S_x}{2} \quad \text{and} \quad D_z = \frac{\rho V_z^2 C_D S_z}{2} \quad (6)$$

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 paper. Therefore, D_x and D_z are only dependent on the velocity components.

B. Flight Constraints

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

$$\mathbf{y}(t_0) = \mathbf{y}_0 \quad (7)$$

$$\mathbf{y}(t_f) = \mathbf{y}_f \quad (8)$$

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

$$0 \leq x \leq x_{\max} \quad (9)$$

$$0 \leq z \leq z_{\max} \quad (10)$$

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

$$\sqrt{V_x^2 + V_z^2} \leq V_{\max} \quad (11)$$

$$0 \leq T \leq T_{\max} \quad (12)$$

Moreover, we restrict the vehicle's pitch angle θ based on the passenger's comfort level as follows:

$$|\theta| \leq \theta_{\max} \quad (13)$$

Because the problem considered in this paper 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 as follows:

$$\mathbf{y}^p(t_f^p) - \mathbf{y}^{p+1}(t_0^{p+1}) = 0 \quad (14)$$

$$t_f^p - t_0^{p+1} = 0 \quad (15)$$

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

C. Performance Index and Optimal Control Problem

For the multiphase 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 as follows:

$$J = \int_{t_0}^{t_f} \frac{1}{2} T^2 dt \quad (16)$$

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

$$J = \sum_{p=1}^2 \int_{t_0^p}^{t_f^p} \frac{1}{2} T^2 dt \quad (17)$$

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

Problem 1:

Minimize : (17)

Subject to : (1) – (4), (7) – (15)

Overall, we aim to minimize the objective function (17) while satisfying the system dynamics (1–4), the boundary conditions (7) and (8), the inequality state and control constraints (9–13), and the linkage constraints (14) and (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 make 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 compute the solution.

III. Problem Reformulation and Convexification

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 allow us to obtain optimal solutions in real time. Multiphase convex optimization method has been successfully used to solve problems such as reentry trajectory optimization [33] and rocket launch trajectory optimization [25,26]. Motivated by these previous works, we develop and customize a convex approach to the multiphase eVTOL trajectory optimization problem for UAM scenarios in this section.

It is worth mentioning that our considered multiphase UAM problem is fundamentally different from the multiphase problems solved in [25,26,33] in terms of the path constraints, relaxed control constraints, and objective functions involved. For example, bounds on the state variables are enforced in this paper such that the vehicle flies within a more confined airspace. More relaxed control constraints are considered in this paper due to the constrained tilt angle for safe and comfortable 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 rederived.

The convexification process for the considered multiphase trajectory optimization problem is adapted partially from our previous work [30]; 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 Sec. III.A, an alternative strategy for addressing possible nonequivalent relaxations in Sec. III.B, convexification of the free time-of-arrival at the intermediate waypoint in Sec. III.C, and problem discretization in Sec. III.D, followed by a multiphase SCP algorithm in Sec. III.E.

A. 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:

$$u_1 = T \sin \theta, \quad u_2 = T \cos \theta, \quad \text{and} \quad u_3 = T^2 \quad (18)$$

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

$$u_1^2 + u_2^2 = u_3 \quad (19)$$

Then, the original system dynamics in Eqs. (1–4) become

$$\dot{x} = V_x \quad (20)$$

$$\dot{z} = V_z \quad (21)$$

$$\dot{V}_x = \frac{u_1}{m} - \frac{D_x}{m} \quad (22)$$

$$\dot{V}_z = \frac{u_2}{m} - \frac{D_z}{m} - g \quad (23)$$

and the objective function takes the following new form:

$$J = \sum_{p=1}^2 \int_{t_0^p}^{t_f^p} \frac{1}{2} u_3 \, dt \quad (24)$$

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 Eqs. (12) and (13) are met:

$$0 \leq u_3 \leq T_{\max}^2 \quad (25)$$

$$u_2 \tan(-\theta_{\max}) \leq u_1 \leq u_2 \tan(\theta_{\max}) \quad (26)$$

Notice that the feasible set defined by the new equality control constraint in Eq. (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:

$$u_1^2 + u_2^2 \leq u_3 \quad (27)$$

Remark 2: Note that constraint (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 $-(\pi/2)$ and $\pi/2$, which makes constraint (26) equivalent to the original constraint (13).

Now, u_3 is bounded by Eqs. (26) and (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. 1.

The equivalence of the constraint relaxation in Eq. (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 Eqs. (25) and (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 Eqs. (25) and (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 Eq. (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 Eq. (10) is inactive almost everywhere on $[t_0, t_f]$.

Proposition 1: If $\{y^*(t), u^*(t)\}$ is an optimal solution to the problem after relaxation, then the optimal control $u^*(t)$ lies on the boundary of the cone defined by Eq. (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 for optimal control [34]. First, the Hamiltonian for the relaxed problem is formulated as follows:

$$H(y, u, \lambda_0, \lambda) = \lambda_0 \frac{1}{2} u_3 + \lambda_1 V_x + \lambda_2 V_z + \lambda_3 \left(\frac{u_1}{m} - \frac{D_x}{m} \right) + \lambda_4 \left(\frac{u_2}{m} - \frac{D_z}{m} - g \right) \quad (28)$$

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 Eq. (10) and the maximum speed constraint in Eq. (11) are not required to be inactive. To derive the KKT conditions for minimizing H over the constraints, we introduce the Lagrangian L as follows based on Assumptions 1 and 2:

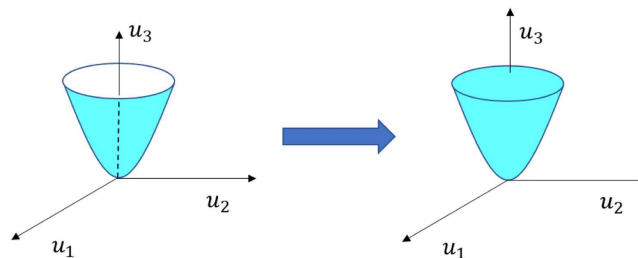


Fig. 1 Relaxation of control constraint.

$$L(\mathbf{y}, \mathbf{u}, \lambda_0, \lambda, \nu) = H(\mathbf{y}, \mathbf{u}, \lambda_0, \lambda) + \nu_1(u_3 - u_1^2 - u_2^2) + \nu_2(z_{\max} - z) + \nu_3(V_{\max}^2 - V_x^2 - V_z^2) \quad (29)$$

where $\nu = [\nu_1; \nu_2; \nu_3]$ is the Lagrangian multiplier vector. Note that the airspeed constraint in Eq. (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 as follows:

$$\dot{\lambda}_1 = -\frac{\partial L}{\partial x} = 0 \quad (30)$$

$$\dot{\lambda}_2 = -\frac{\partial L}{\partial z} = \nu_2 \quad (31)$$

$$\dot{\lambda}_3 = -\frac{\partial L}{\partial V_x} = -\lambda_1 + \lambda_3 \frac{\rho C_D S_x V_z}{m} + 2\nu_3 V_x \quad (32)$$

$$\dot{\lambda}_4 = -\frac{\partial L}{\partial V_z} = -\lambda_2 + \lambda_4 \frac{\rho C_D S_z V_x}{m} + 2\nu_3 V_z \quad (33)$$

The stationary conditions are as follows:

$$\frac{\partial L}{\partial u_1} = \frac{\lambda_3}{m} - 2\nu_1 u_1 = 0 \quad (34)$$

$$\frac{\partial L}{\partial u_2} = \frac{\lambda_4}{m} - 2\nu_1 u_2 = 0 \quad (35)$$

$$\frac{\partial L}{\partial u_3} = \frac{\lambda_0}{2} + \nu_1 = 0 \quad (36)$$

The complementary slack conditions are as follows:

$$\nu_1 \geq 0, \quad \nu_1(u_3^* - u_1^{*2} - u_2^{*2}) = 0 \quad (37)$$

$$\nu_2 \geq 0, \quad \nu_2(z_{\max} - z^*) = 0 \quad (38)$$

$$\nu_3 \geq 0, \quad \nu_3(V_{\max}^2 - V_x^{*2} - V_z^{*2}) = 0 \quad (39)$$

The Informal Theorem 4.1 in [34] 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. (30–39) 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 Eq. (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 Eq. (37) we have $\nu_1 = 0$. Using $\nu_1 = 0$ in Eqs. (34) and (35), we can get $\lambda_3 = \lambda_4 = 0$. With this result, we solve Eqs. (32) and (33) for λ_1 and λ_2 , respectively, as follows:

$$\lambda_1 = 2\nu_3 V_x \quad (40)$$

$$\lambda_2 = 2\nu_3 V_z \quad (41)$$

From Eq. (30), we know that λ_1 is constant, while V_x is varying during the flight. Therefore, to make Eq. (40) hold, we have $\nu_3 = 0$ and then $\lambda_1 = 0$. Further, we get $\lambda_2 = 0$ from Eq. (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 Eq. (31), we have $\nu_2 = 0$. At last, solving Eq. (36) for λ_0 we obtain $\lambda_0 = -2\nu_1 = 0$. 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 Eq. (27) is equivalent. $\square$

Finally, the only nonconvex term left in the problem is the nonlinear dynamic system defined by Eqs. (20–23). Based on the state and control vectors defined in Eq. (5), the equations of motion can be rewritten as a state representation as follows:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}) + \mathbf{B}\mathbf{u} \quad (42)$$

where

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} V_x \\ V_z \\ -\frac{\rho V_x^2 C_D S_x}{2m} \\ -\frac{\rho V_z^2 C_D S_z}{2m} - g \end{bmatrix} \quad \text{and} \quad \mathbf{B}\mathbf{u} = \begin{bmatrix} 0 \\ 0 \\ \frac{u_1}{m} \\ \frac{u_2}{m} \end{bmatrix} \quad (43)$$

To convexify Eq. (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 Sec. III.E). With the linear approximation method, the state equation becomes

$$\dot{\mathbf{y}} \approx \mathbf{f}(\mathbf{y}^*) + \mathbf{A}(\mathbf{y}^*)(\mathbf{y} - \mathbf{y}^*) + \mathbf{B}\mathbf{u} \quad (44)$$

where $\mathbf{A}$ is the state matrix defined by partial derivatives and evaluated at the reference solution as follows:

$$\mathbf{A}(\mathbf{y}^*) = \left[\frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}} \right]_{\mathbf{y}=\mathbf{y}^*} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{\rho V_x C_D S_x}{m} & 0 \\ 0 & 0 & 0 & -\frac{\rho V_z C_D S_z}{m} \end{bmatrix}_{\mathbf{y}=\mathbf{y}^*} \quad (45)$$

and $\mathbf{B}$ is a constant control matrix:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \end{bmatrix} \quad (46)$$

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

$$\|\mathbf{y} - \mathbf{y}^*\| \leq \delta \quad (47)$$

where δ 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 in the algorithm. The trust-region size can be potentially adjusted based on the iteration process of the algorithm for better convergence [24]. In this paper, 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 Sec. IV.C.

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.

B. 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 paper, it is possible that the relaxed control constraint (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:

$$u_1 = T \sin \theta, \quad u_2 = T \cos \theta \quad (48)$$

Therefore, the constraints (25) and (27) can be combined as

$$u_1^2 + u_2^2 \leq T_{\max}^2 \quad (49)$$

with the same inequality constraint for u_1 and u_2 as in Eq. (26). Then, the objective function can be rewritten as follows:

$$J = \sum_{p=1}^2 \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} (u_1^2 + u_2^2) d\tau \quad (50)$$

Remark 5: Note that the new control constraint (49) is equivalent to the original thrust constraint (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 paper 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 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 Sec. IV, the second approach seems superior to the first one in terms of convergence and stability.

C. 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 [26]. The time dilation σ between t and τ is defined as

$$\sigma = \frac{dt}{d\tau} = t_f - t_0 \quad (51)$$

Then, the objective functions (24) and (50) are therefore, respectively, changed to

$$J = \sum_{p=1}^2 \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} u_3 \sigma d\tau \quad (52a)$$

$$J = \sum_{p=1}^2 \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} (u_1^2 + u_2^2) \sigma d\tau \quad (52b)$$

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. (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:

$$J = \sum_{p=1}^2 \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} [\bar{u}_3 \bar{\sigma} + \bar{\sigma} (u_3 - \bar{u}_3) + \bar{u}_3 (\sigma - \bar{\sigma})] d\tau \quad (53a)$$

$$J = \sum_{p=1}^2 \int_{\tau_0^p}^{\tau_f^p} \frac{1}{2} [(\bar{u}_1^2 + \bar{u}_2^2) \bar{\sigma} + \bar{\sigma} (u_1^2 + u_2^2 - \bar{u}_1^2 - \bar{u}_2^2) + (\bar{u}_1^2 + \bar{u}_2^2) (\sigma - \bar{\sigma})] d\tau \quad (53b)$$

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

$$\mathbf{y}' = \frac{d\mathbf{y}}{d\tau} = \sigma \mathbf{f}(\mathbf{y}(\tau), \mathbf{u}(\tau)) \approx \mathbf{A}\mathbf{y} + \mathbf{B}\mathbf{u} + \mathbf{F}\sigma + \mathbf{c} \quad (54)$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$, and $\mathbf{c}$ are obtained from the following:

$$\mathbf{A}(\tau) = \bar{\sigma} \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \quad (55)$$

$$\mathbf{B}(\tau) = \bar{\sigma} \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) \quad (56)$$

$$\mathbf{F}(\tau) = \mathbf{f}(\bar{\mathbf{y}}, \bar{\mathbf{u}}) = \mathbf{f}(\bar{\mathbf{y}}) + \mathbf{B}\bar{\mathbf{u}} \quad (57)$$

$$\mathbf{c}(\tau) = -(\mathbf{A}\bar{\mathbf{y}} + \mathbf{B}\bar{\mathbf{u}}) \quad (58)$$

Note that all the variables with a bar on top denote the reference solution and value that will be updated and obtained from the previous step of the SCP procedure that will be detailed in Sec. III.E.

With the above development and through the application of the two convex approaches, the overall optimal control problems are now defined, respectively, as follows:

Problem 2 (First Convex Approach):

$$\begin{aligned} &\text{Minimize : } (53a) \\ &\text{Subject to : } (54), (7) - (11), (14), (15), (25) - (27), (47) \end{aligned}$$

Problem 3 (Second Convex Approach):

$$\begin{aligned} &\text{Minimize : } (53b) \\ &\text{Subject to : } (54), (7) - (11), (14), (15), (49), (47) \end{aligned}$$

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 the 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.

D. Discretization

To obtain a numerical solution, we will discretize the continuous-time optimal control problem (Problems 2 and 3) into a parameter optimization problem with a finite set of variables and constraints that can be solved by numerical optimization solvers. In this paper, we apply a trapezoidal rule for discretization [22]. 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 $\{\tau_0, \tau_1, \dots, \tau_N\}$ with $\tau_i = \tau_{i-1} + \Delta\tau$, $i = 1, 2, \dots, N$. The corresponding state and control profiles are then discretized at the $N + 1$ points.

By applying the trapezoidal rule, the current state Y_i can be obtained from the previous step Y_{i-1} as follows:

$$Y_i = Y_{i-1} + \frac{\Delta\tau}{2} [(\sigma^{k-1} A_{i-1}^{k-1} Y_{i-1} + \sigma^{k-1} B u_{i-1} + \sigma(f_{i-1}^{k-1} + B u_{i-1}^{k-1}) - (\sigma^{k-1} A_{i-1}^{k-1} + \sigma^{k-1} B u_{i-1}^{k-1})) + (\sigma^{k-1} A_i^{k-1} Y_i + \sigma^{k-1} B u_i + \sigma(f_i^{k-1} + B u_i^{k-1}) - (\sigma^{k-1} A_i^{k-1} + \sigma^{k-1} B u_i^{k-1}))] \quad (59)$$

After rearrangement, we have

$$\begin{aligned} \left(I - \frac{\Delta\tau}{2} \sigma^{k-1} A_i^{k-1}\right) Y_i - \left(I + \frac{\Delta\tau}{2} \sigma^{k-1} A_{i-1}^{k-1}\right) Y_{i-1} - \frac{\Delta\tau}{2} \sigma^{k-1} B u_i \\ - \frac{\Delta\tau}{2} \sigma^{k-1} B u_{i-1} = \frac{\Delta\tau}{2} [(\sigma(f_{i-1}^{k-1} + B u_{i-1}^{k-1}) - (\sigma^{k-1} A_{i-1}^{k-1} + \sigma^{k-1} B u_{i-1}^{k-1})) \\ + (\sigma(f_i^{k-1} + B u_i^{k-1}) - (\sigma^{k-1} A_i^{k-1} + \sigma^{k-1} B u_i^{k-1}))] \end{aligned} \quad (60)$$

The linearized objective functions in Eqs. (53a) and (53b) can be discretized and represented, respectively, as follows:

$$J = \frac{\Delta\tau}{2} \sum_{p=1}^2 \sum_{i=1}^{N-1} [\bar{u}_{3i}^{k-1} \bar{\sigma}^{k-1} + \bar{\sigma}^{k-1} (u_{3i} - \bar{u}_{3i}^{k-1}) + \bar{u}_{3i}^{k-1} (\sigma - \bar{\sigma}^{k-1})] \quad (61a)$$

$$\begin{aligned} J = \frac{\Delta\tau}{2} \sum_{p=1}^2 \sum_{i=1}^{N-1} [(\bar{u}_{1i}^{2(k-1)} + \bar{u}_{2i}^{2(k-1)}) \bar{\sigma}^{k-1} \\ + \bar{\sigma}^{k-1} (u_{1i}^2 + u_{2i}^2 - \bar{u}_{1i}^{2(k-1)} - \bar{u}_{2i}^{2(k-1)}) \\ + (\bar{u}_{1i}^{2(k-1)} + \bar{u}_{2i}^{2(k-1)}) (\sigma - \bar{\sigma}^{k-1})] \end{aligned} \quad (61b)$$

Lastly, because an RTA constraint is considered in this paper, 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:

$$\sum_{p=1}^2 \sigma^p = t_f \quad (62)$$

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

Problem 4 (First Convex Approach):

$$\begin{aligned} \text{Minimize : } & (61a) \\ \text{Subject to : } & (60), (7) - (11), (14), (15), (25) - (27), (47), (62) \end{aligned}$$

Problem 5 (Second Convex Approach):

$$\begin{aligned} \text{Minimize : } & (61b) \\ \text{Subject to : } & (60), (7) - (11), (14), (15), (49), (47), (62) \end{aligned}$$

Remark 8: At this point, Problems 2 and 3 have been discretized into Problems 4 and 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 Problems 4 and 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

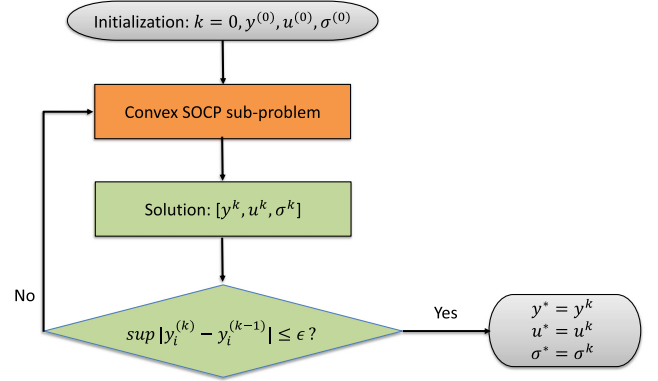


Fig. 2 SCP flowchart.

(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 problem. A general convex optimization problem has the following structure [16]:

$$\begin{aligned} \text{Minimize } & f_0(y) \\ \text{Subject to } & f_i(y) \leq b_i \quad i = 1, \dots, m \end{aligned} \quad (63)$$

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

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \quad (64)$$

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

E. Multiphase Sequential Convex Programming

In this work, for the first time, an SCP method is developed to obtain the solution to the multiphase nonconvex UAM trajectory optimization problem. Figure 2 shows a flowchart that illustrates the SCP algorithm developed in this paper. The SCP method for solving the two-phase problem is described in the following steps:

1) Initialize the iteration index $k = 0$ and phase number $i = 0$. Specify the initial state $y_i(t_0)$. Propagate the equations of motion with the initial condition to obtain an initial trajectory $y^{(0)}$. In this paper, 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 $\{y_i^{(k)}, u_i^{(k)}, \sigma_i^{(k)}\} (i = 0, 1)$ at the current step.

4) Check the convergence condition

$$\sup |y_i^{(k)} - y_i^{(k-1)}| \leq \epsilon, \quad i = 0, 1 \quad (65)$$

where ϵ 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 $\{y_i^{(k)}, u_i^{(k)}, \sigma_i^{(k)}\} (i = 0, 1)$.

IV. Numerical Simulation

In this section, the proposed convex approach is validated through numerical simulations based on the EHang 184 model with vehicle parameters listed in Table 1, the initial and terminal conditions shown in Table 2, and the number of discretized segments defined in Table 3. The number of discretized nodes used in the simulations is based on the tradeoff between solution accuracy and computational time. The eVTOL vehicle starts at a cruise speed of 13.83 m/s and is 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

Table 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, $T_{\max}$	4800 N
Maximum rotor tip-path-plane pitch angle, $\theta_{\max}$	6 deg
Time of flight, t_f	25 min

Table 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 Number of discretized nodes for each phase of each scenario

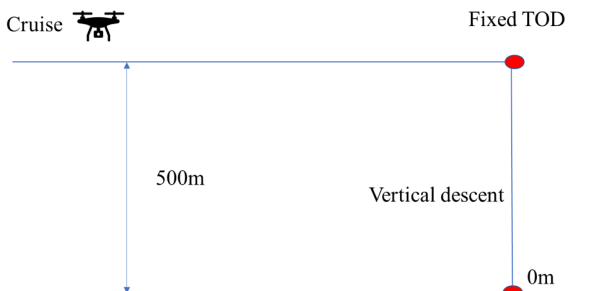
Scenario	N of phase one	N of phase two
1	50	50
2	50	70

Ryzen 7 1800X eight-core processor. The convex optimization problems are implemented in the YALMIP [35] optimization toolbox, with ECOS [36] as the convex optimization solver.

For both simulation cases considered in this paper, the baseline trust-region size in Eq. (47) is selected as $\delta = [200 \text{ m}, 5 \text{ m}, 3 \text{ m/s}, 3 \text{ m/s}, 0.1 * \sigma_{1\text{init}}, 0.1 * \sigma_{2\text{init}}]^T$, where $\sigma_{1\text{init}}$ and $\sigma_{2\text{init}}$ are the guessed flight times for the first and second phases, respectively, based on the flight distance and the steady-state airspeed. The stopping criteria ϵ in Eq. (65) is selected as $\epsilon = [0.5 \text{ m}, 0.5 \text{ m}, 0.3 \text{ m/s}, 0.3 \text{ m/s}, 0.8 \text{ s}, 0.8 \text{ s}]^T$.

A. Case One: Vertical Descent and Landing

Figure 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 1500 s.

**Fig. 3** Schematic for the first scenario.

Both convexification methods are used to solve the problem, and the results are compared in Figs. 4–7. The subplots on the left illustrate the results of the first convex approach, while the subplots on the right display the outcomes of the second convexification approach. Figure 4 presents the convergence of the objective value by the two convex 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 5 and 6 illustrate the convergence of differences between consecutive intermediate steps in the along-track and vertical directions, respectively. Both convex approaches have met 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 $i = 1, 2, \dots, N$. In each step, ECOS takes about 1.5 s 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 7 shows a clear convergence of the along-track airspeed versus 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 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 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 8–12 show the detailed comparisons between the results from the GPOPS solver and our proposed SCP methods. From Fig. 8, we can see that both solvers and both convex 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. 9 and 10. From Figs. 11 and 12, we can see that the net thrust curves obtained by the SCP methods are smoother than those of the GPOPS solver in the last descent and landing phases. However, the pitch angle curves from the SCP algorithms have more jitters during the second phase. Overall, the control curves from both solvers follow a 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 s to converge. In contrast, the SCP method exhibits quicker convergence: with the first approach, it converges in 10 iterations, taking a total of 10 s; with the second approach, it converges in just five iterations, requiring less than 6 s. 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 convex 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 (27) throughout the flight mission.

B. Case Two: Inclined Descent and Landing

In the second scenario, as shown in Fig. 13, 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 endpoints 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

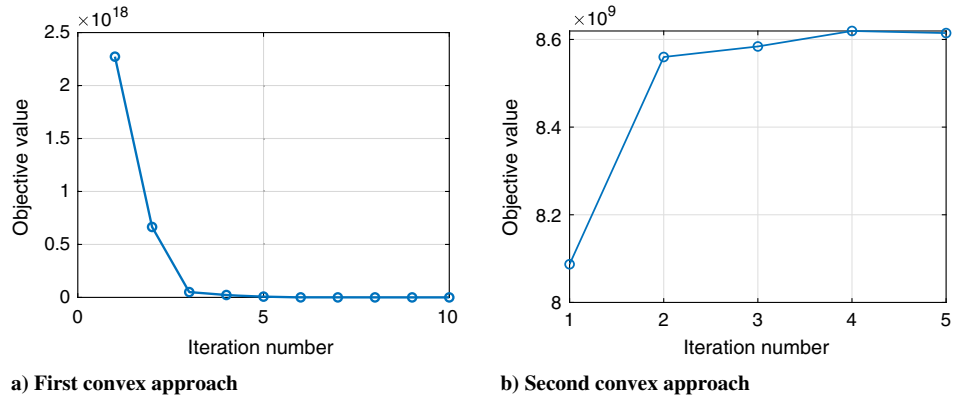


Fig. 4 Convergence of objective value for the first scenario.

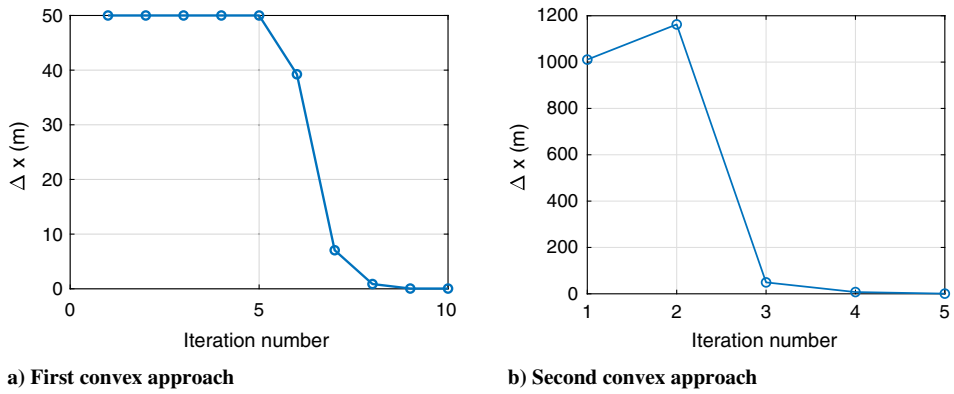


Fig. 5 Convergence of along-track distance for the first scenario.

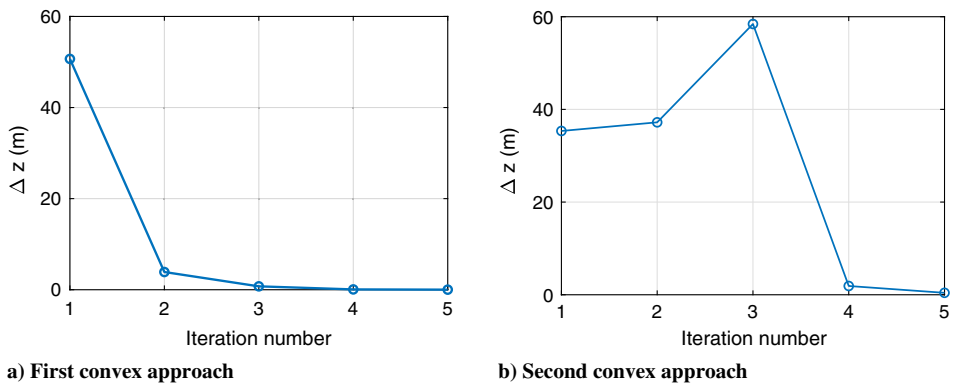


Fig. 6 Convergence of altitude for the first scenario.

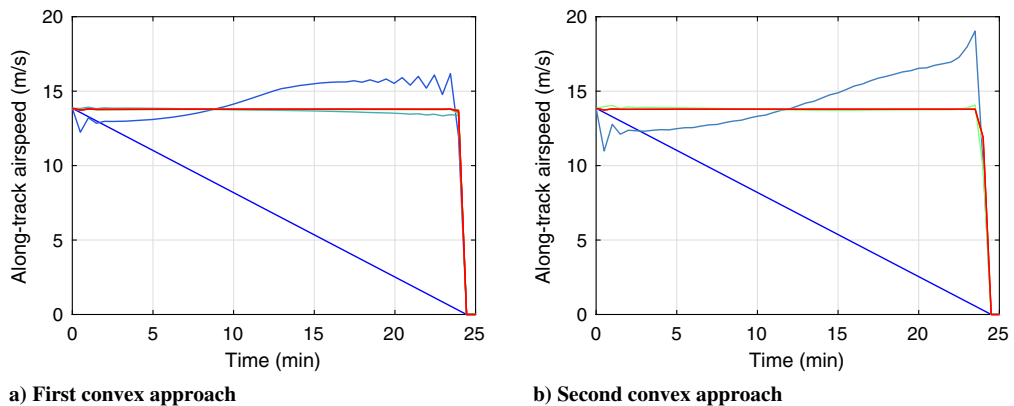


Fig. 7 Convergence of along-track airspeed for the first scenario.

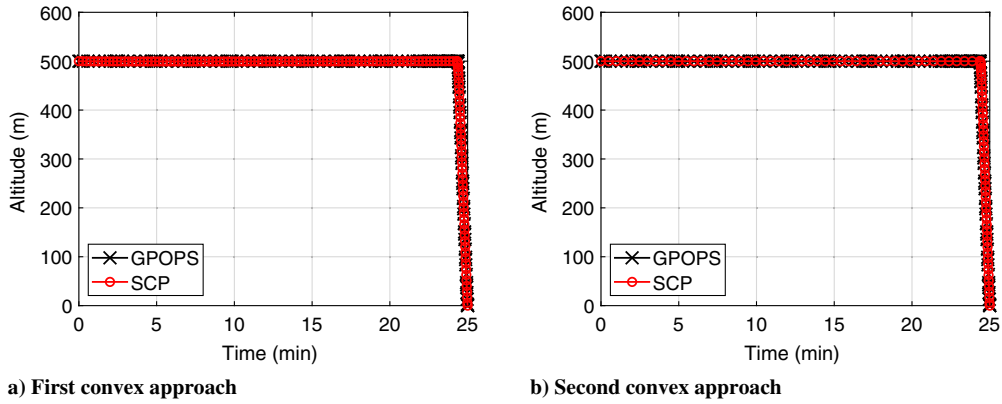


Fig. 8 Altitude versus time profiles for the first scenario.

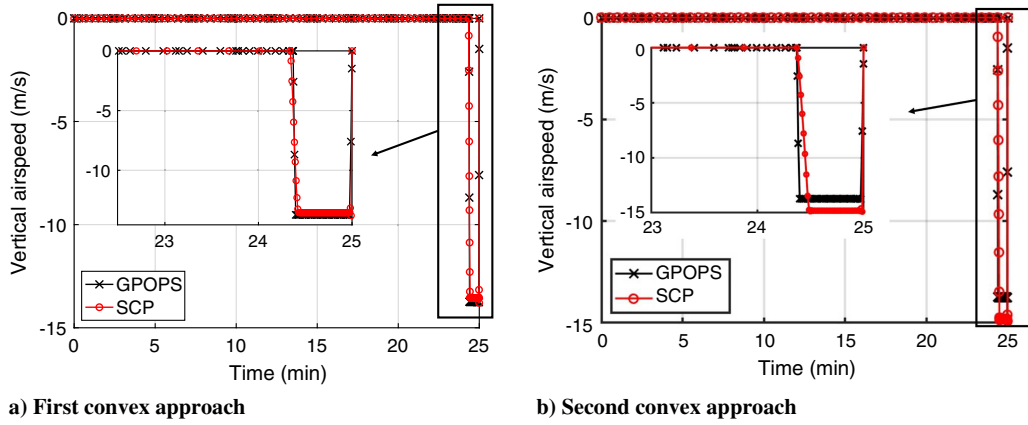


Fig. 9 Vertical speed versus time profiles for the first scenario.

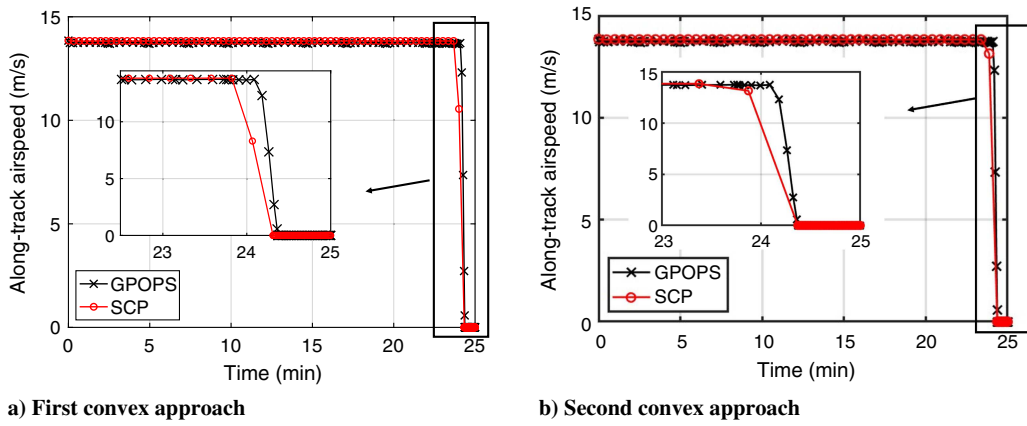


Fig. 10 Along-track speed versus time profiles for the first scenario.

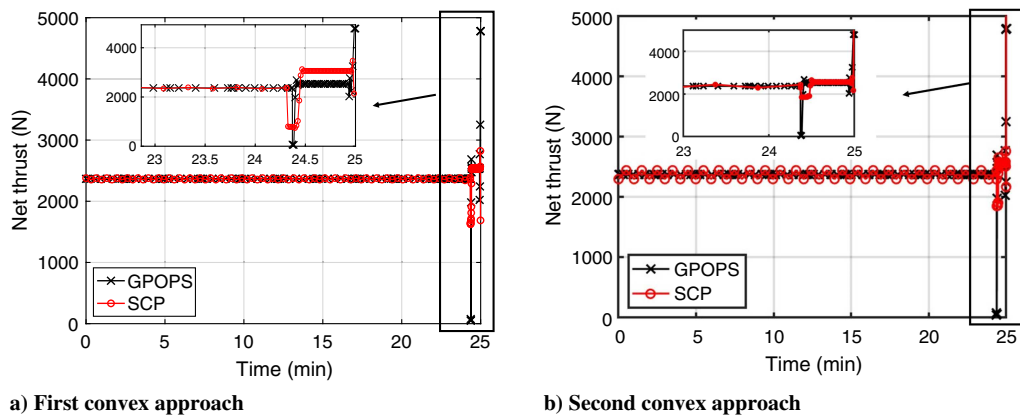


Fig. 11 Net thrust versus time profiles for the first scenario.

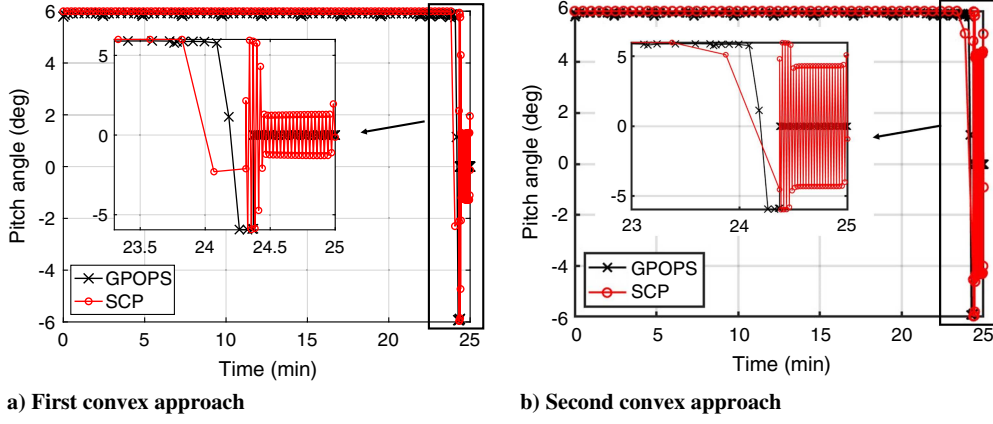


Fig. 12 Pitch angle versus time profiles for the first scenario.

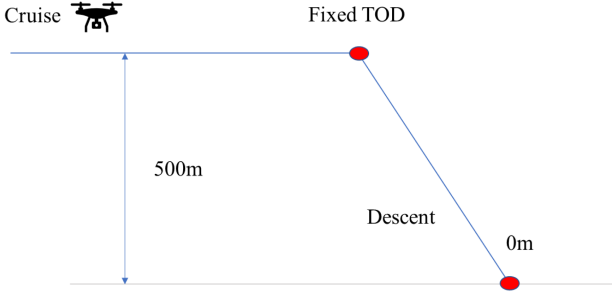


Fig. 13 Schematic for the second scenario.

optimize. This scenario usually happens in a 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 convex approaches as well as GPOPS are then compared with each other.

Figure 14 depicts the variation in the objective value at each iteration when applying the SCP method. The curves clearly demonstrate that both convex approaches achieved convergence in five iterations. Figures 15 and 16 display the convergence of the changes in the distances along the x and z directions, respectively. Figure 17 shows the trajectories of all iterations using 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 overlap after two to three steps. The convergence plots from Figs. 14–17 for both convex approaches exhibit very similar trends. These plots suggest a high degree of similarity in the performance and convergence characteristics of the two methods.

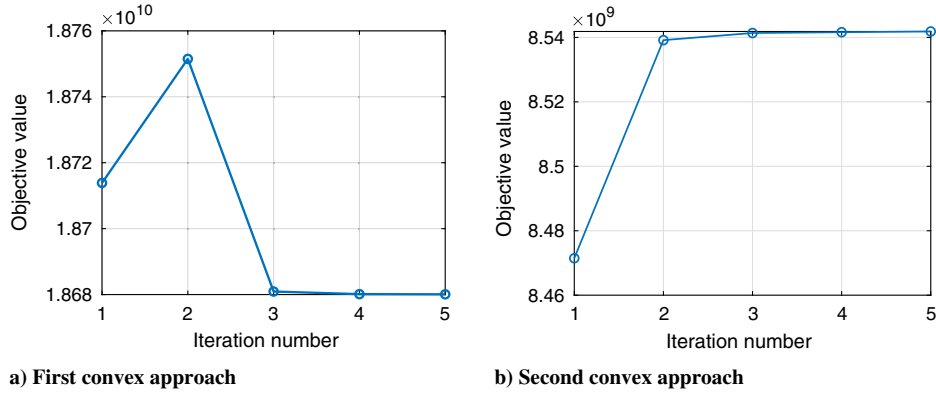


Fig. 14 Convergence of objective value for the second scenario.

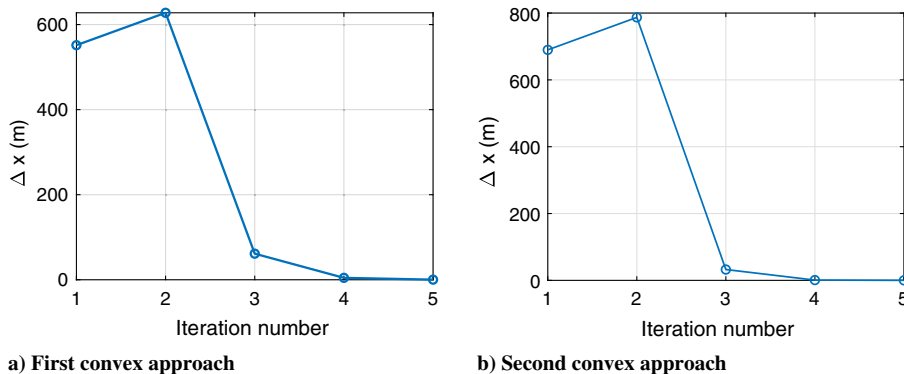


Fig. 15 Convergence of along-track distance for the second scenario.

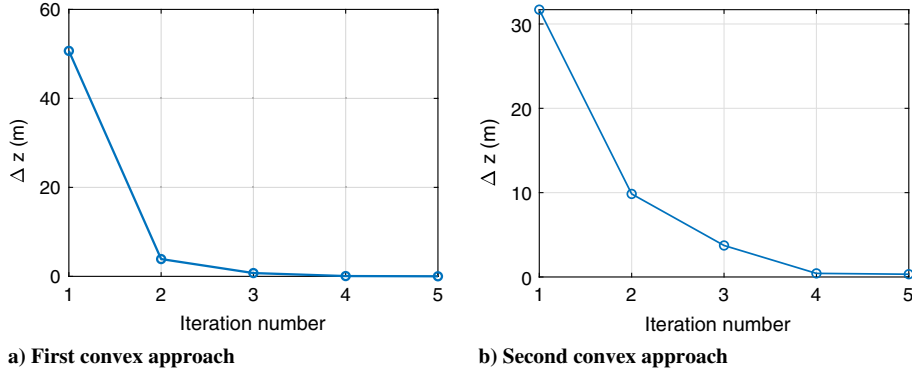


Fig. 16 Convergence of altitude for the second scenario.

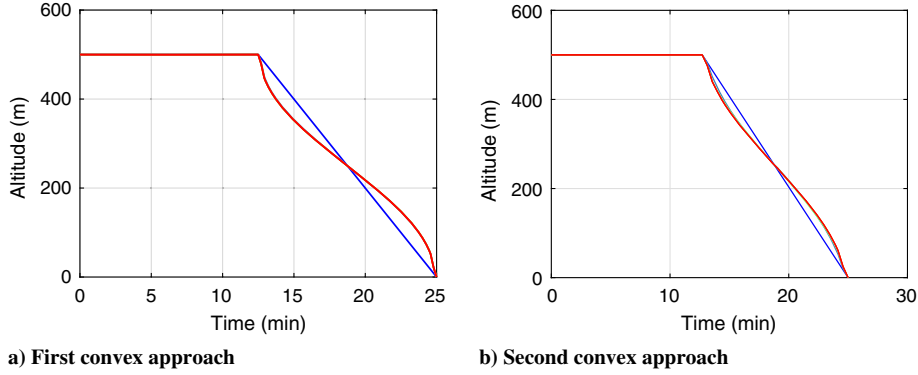


Fig. 17 Convergence of trajectories for the second scenario.

Detailed comparisons between the results from GPOPS and those from the SCP method are shown in Figs. 18–22. Figure 18 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. 19–22 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 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 paper has proven to be helpful in reducing 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 convex 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 s 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 s to solve each SOCP subproblem. Compared to the first simulation case, the longer computational time in each iteration may be caused by the more discretized nodes used in the second case and the 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.

C. 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 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 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

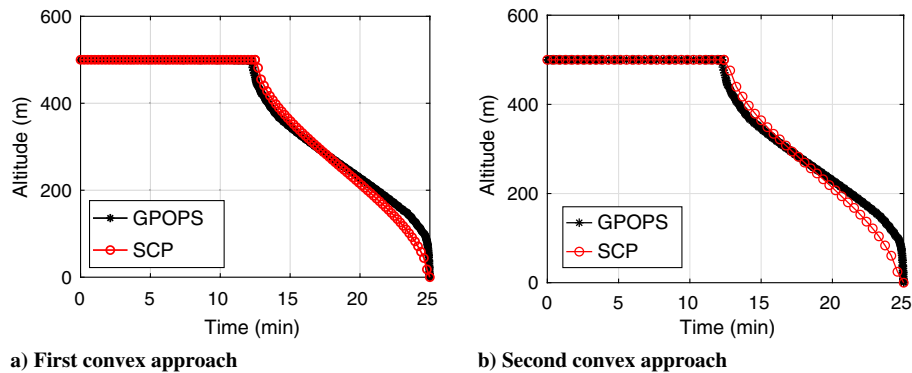


Fig. 18 Altitude versus time profiles for the second scenario.

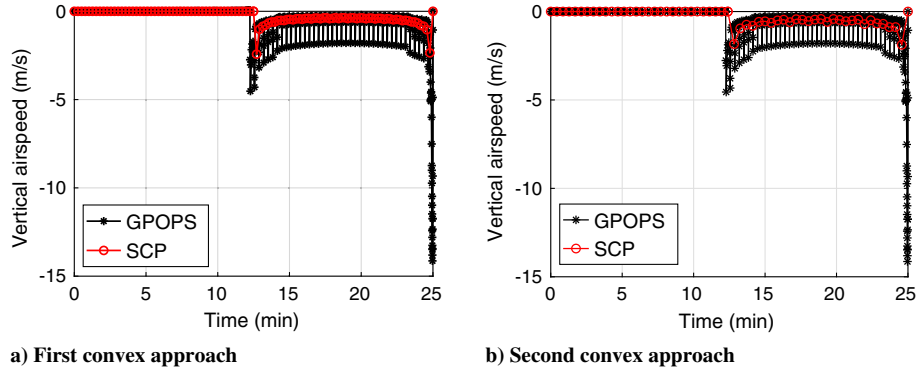


Fig. 19 Vertical speed versus time profiles for the second scenario.

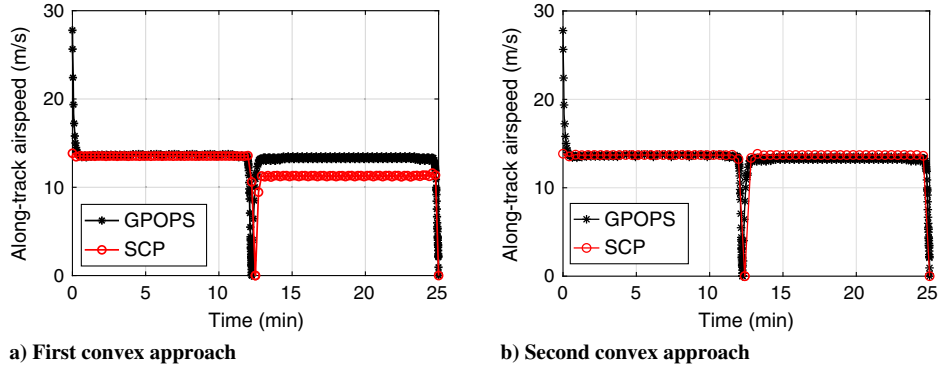


Fig. 20 Along-track speed versus time profiles for the second scenario.

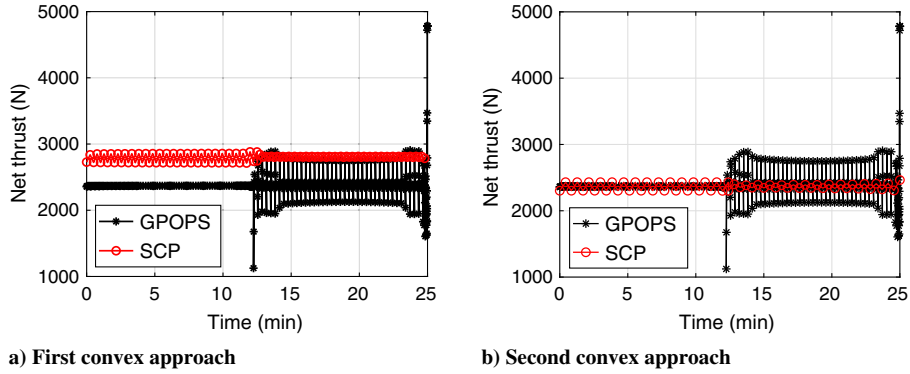


Fig. 21 Net thrust versus time profiles for the second scenario.

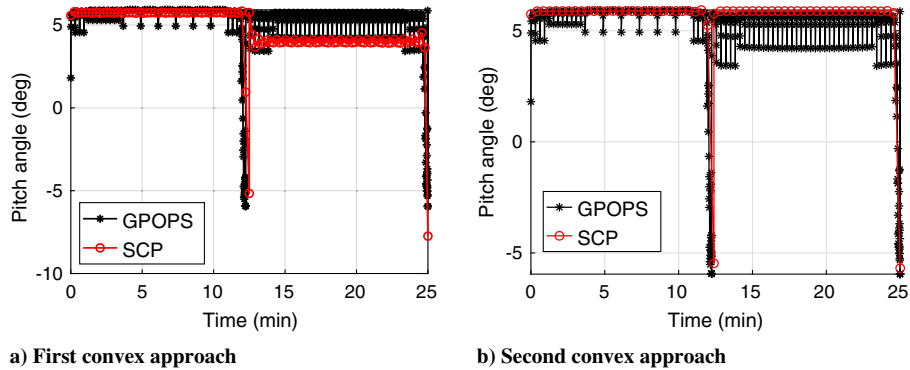


Fig. 22 Pitch angle versus time profiles for the second scenario.

subproblems. This aligns with 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 big

(>80% of the baseline value), the SCP method struggles to converge, especially 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.

Table 4 Impact of trust region size on SCP convergence

Trust region size (% of baseline value)	No. 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 nonconvergence)

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 be 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.

D. 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 Tables 5–8 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 $1e-2$ (for less accurate solutions) and $1e-5$ (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 $1e-0$ to $1e-3$ 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.

Table 5 Computational time of GPOPS for the first CONOP

Mesh tolerance	Average time, s	Median time, s	Standard deviation, s
$1e-2$	3.94	3.76	0.62
$1e-3$	16.52	14.50	4.23
$1e-4$	38.73	35.07	9.72
$1e-5$	98.34	94.72	14.78

Table 6 Computational time of SCP for the first CONOP

Convergence tolerance	Average time, s	Median time, s	Standard deviation, s
$1e-0$	8.01	8.63	0.51
$1e-1$	10.29	9.93	0.64
$1e-2$	19.69	19.83	1.17
$1e-3$	48.82	47.74	2.26

Table 7 Computational time of GPOPS for the second CONOP

Mesh tolerance	Average time, s	Median time, s	Standard deviation, s
$1e-2$	6.08	5.91	0.27
$1e-3$	17.54	18.30	3.29
$1e-4$	46.50	43.23	9.55
$1e-5$	180.50	172.67	20.05

Table 8 Computational time of SCP for the second CONOP

Convergence tolerance	Average time, s	Median time, s	Standard deviation, s
$1e-0$	7.09	7.54	1.12
$1e-1$	14.52	15.15	3.3
$1e-2$	19.91	19.99	2.27
$1e-3$	36.87	35.43	2.32

To provide a statistical analysis of computational efficiency, we ran 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 Tables 5–8. 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 NLP 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 resources, thereby causing an exponential increase in computational time and indicating substantial variability in the solver's performance at high precision levels. 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, 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 tradeoffs 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 integrated the control effort over the time domain for both SCP and GPOPS and compared the results. Table 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 $8.4083e+09$, while SCP with the second convexification approach yielded a slightly higher value of $8.4778e+09$. This suggests that while SCP provides a gain in 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 ($8.4486e+09$) compared to $8.4819e+09$ for GPOPS. This outcome indicates that SCP can be more efficient

Table 9 Comparison of objective values for both scenarios

CONOP	GPOPS objective value	SCP objective value
First	$8.4083e+09$	$8.4778e+09$
Second	$8.4819e+09$	$8.4486e+09$

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 of the convexification process, which can be seen as a tradeoff against its computational benefits.

E. 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 a normal distribution and random variation from their baseline values, as shown in Table 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 paper.

For the cases with $\pm 5\%$ variations, Fig. 23 showcases the converged trajectory profiles of the SCP method across 300 simulations, and Fig. 24 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 of $\pm 5\%$, a looser convergence tolerance of $1e-3$ 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. 25. Each boxplot shows the results of 300

Table 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	$\pm 5\%$	$\pm 10\%$

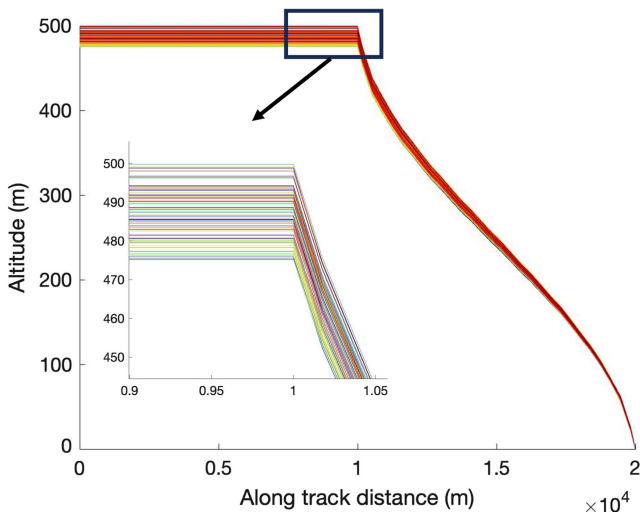


Fig. 23 SCP trajectory profiles from different initial positions with $\pm 5\%$ variations.

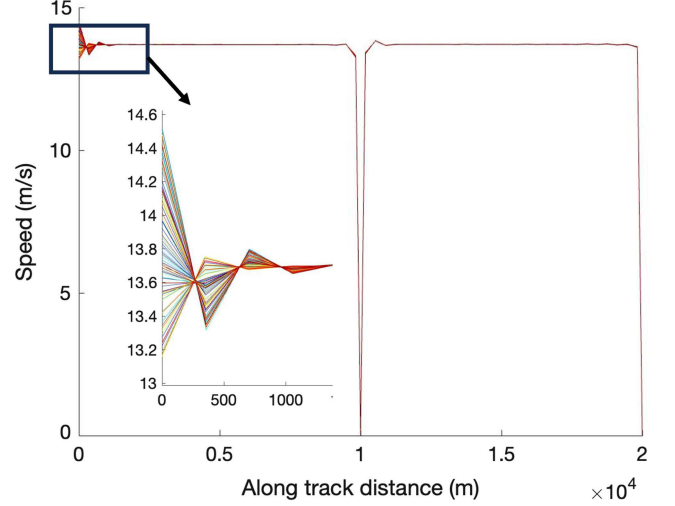


Fig. 24 SCP velocity profiles from different initial velocities with $\pm 5\%$ variations.

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. 26. With a looser tolerance and random variations of $\pm 5\%$, GPOPS achieved an average convergence time of 31.5 s, while the SCP method consistently converged in less than 10 s at $\pm 5\%$ variations and within 12 s at the $\pm 10\%$ variation level. Notably, the SCP method displays wider variations in solution times, with less than five outliers that require more than 20 s 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 are not trying to prove that SCP is superior to GPOPS in all aspects. In fact, GPOPS is a relatively mature and widely recognized optimal control solver, whereas our SCP method developed in this paper is still in its early stages 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.

V. Conclusions

Real-world UAM missions involve multiple flight phases, including takeoff, climb, cruise, descent, and landing, and require the onboard computer to be capable of planning feasible or even optimal trajectories in real time in a complex urban environment. In this paper, we present a successive convexification approach to solving the multiphase trajectory optimization problem for eVTOL vehicles to cruise, 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 a 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 violations of the underlying assumptions for exact relaxation, an alternative convexification method is developed

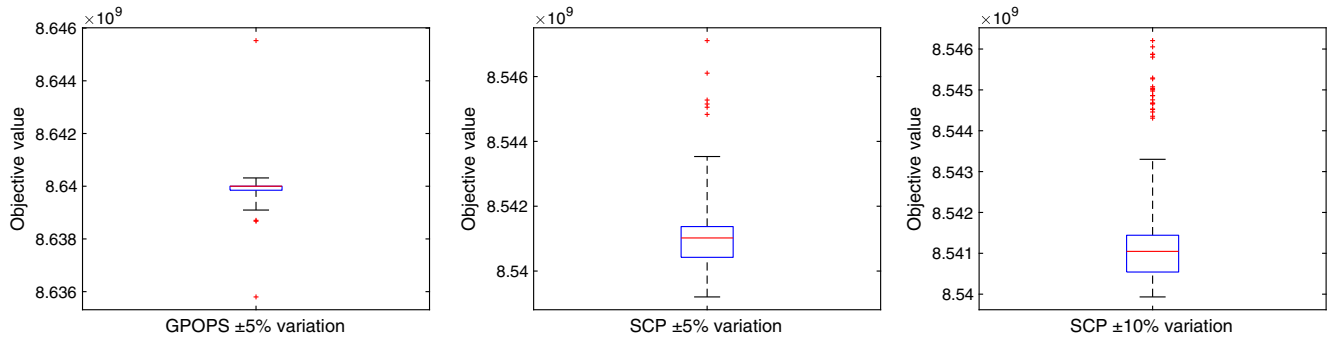


Fig. 25 Boxplot comparison of objective values for SCP and GPOPS.

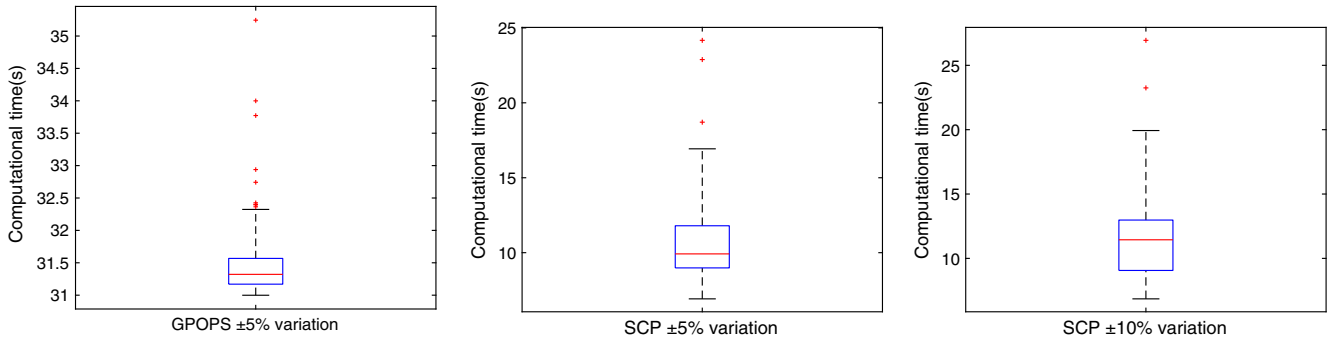


Fig. 26 Boxplot comparison of computational times for SCP and GPOPS.

by eliminating the need for 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-II 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 became 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.

Acknowledgments

The authors acknowledge the support for this work from the National Science Foundation grant CMMI-CAREER-2237215. The authors would also like to acknowledge the anonymous reviewers and the Associate Editor for providing valuable comments and suggestions on improving the quality of this work.

References

- [1] Holden, J., and Goel, N., "Fast-Forwarding to a Future of On-Demand Urban Air Transportation," San Francisco, 2016.
- [2] Hill, B. P., DeCarme, D., Metcalfe, M., Griffin, C., Wiggins, S., Metts, C., Bastedo, B., Patterson, M. D., and Mendonca, N. L., "UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4," Tech. Rept., NASA, Dec. 2020.
- [3] Prevot, T., Rios, J., Kopardekar, P., Robinson, J. E., III, Johnson, M., and Jung, J., "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2016-3292, 2016.
<https://doi.org/10.2514/6.2016-3292>
- [4] Thippavong, D. P., Apaza, R., Barmore, B., Battiste, V., Burian, B., Dao, Q., Feary, M., Go, S., Goodrich, K. H., Homola, J., et al., "Urban Air Mobility Airspace Integration Concepts and Considerations," *2018 Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2018-3676, 2018.
<https://doi.org/10.2514/6.2018-3676>
- [5] Vascik, P. D., Hansman, R. J., and Dunn, N. S., "Analysis of Urban Air Mobility Operational Constraints," *Journal of Air Transportation*, Vol. 26, No. 4, 2018, pp. 133–146.
<https://doi.org/10.2514/1.D0120>
- [6] Al Haddad, C., Chaniotakis, E., Straubinger, A., Plötner, K., and Antoniou, C., "Factors Affecting the Adoption and use of Urban Air Mobility," *Transportation Research Part A: Policy and Practice*, Vol. 132, Feb. 2020, pp. 696–712.
<https://doi.org/10.1016/j.tra.2019.12.020>
- [7] Pradeep, P., and Wei, P., "Energy-Efficient Arrival with RTA Constraint for Multirotor eVTOL in Urban Air Mobility," *Journal of Aerospace Information Systems*, Vol. 16, No. 7, 2019, pp. 263–277.
<https://doi.org/10.2514/1.1010710>
- [8] Marshall, J. A., Sun, W., and L'Afflito, A., "A Survey of Guidance, Navigation, and Control Systems for Autonomous Multi-Rotor Small Unmanned Aerial Systems," *Annual Reviews in Control*, Vol. 52, Jan. 2021, pp. 390–427.
<https://doi.org/10.1016/j.arcontrol.2021.10.013>
- [9] Chauhan, S. S., and Martins, J. R., "Tilt-Wing eVTOL Takeoff Trajectory Optimization," *Journal of Aircraft*, Vol. 57, No. 1, 2020, pp. 93–112.
<https://doi.org/10.2514/1.C035476>
- [10] Wu, Y., Deniz, S., Shi, Y., and Wang, Z., "A Convex Optimization Approach to Real-Time Merging Control of eVTOL Vehicles for Future Urban Air Mobility," *AIAA AVIATION 2022 Forum*, AIAA Paper 2022-3319, 2022.
<https://doi.org/10.2514/6.2022-3319>
- [11] Pradeep, P., Lauderdale, T. A., Chatterji, G. B., Sheth, K., Lai, C. F., Sridhar, B., Edholm, K.-M., and Erzberger, H., "Wind-Optimal Trajectories for Multirotor eVTOL Aircraft on UAM Missions," *AIAA AVIATION 2020 FORUM*, AIAA Paper 2020-3271, 2020.
<https://doi.org/10.2514/6.2020-3271>
- [12] Von Stryk, O., and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, No. 1, 1992, pp. 357–373.
<https://doi.org/10.1007/BF02071065>
- [13] Wächter, A., and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57.
<https://doi.org/10.1007/s10107-004-0559-y>

- [14] Lin, S., "Computer Solutions of the Traveling Salesman Problem," *Bell System Technical Journal*, Vol. 44, No. 10, 1965, pp. 2245–2269. <https://doi.org/10.1002/j.1538-7305.1965.tb04146.x>
- [15] Wang, Z., "A Survey on Convex Optimization for Guidance and Control of Vehicular Systems," arXiv preprint arXiv:2311.05115, 2023. <https://doi.org/10.48550/arXiv.2311.05115>
- [16] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004. <https://doi.org/10.1017/CBO9780511804441>
- [17] Duchi, J., "Sequential Convex Programming," 2018, https://stanford.edu/class/ee364b/lectures/seq_notes.pdf.
- [18] Acikmese, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. <https://doi.org/10.2514/1.27553>
- [19] Blackmore, L., Aikmee, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. <https://doi.org/10.2514/1.47202>
- [20] Wang, Z., and Grant, M. J., "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 5, 2018, pp. 2274–2290. <https://doi.org/10.1109/TAES.2018.2812558>
- [21] Wang, Z., and Grant, M. J., "Optimization of Minimum-Time Low-Thrust Transfers Using Convex Programming," *Journal of Spacecraft and Rockets*, Vol. 55, No. 3, 2018, pp. 586–598. <https://doi.org/10.2514/1.A33995>
- [22] Wang, Z., and Grant, M. J., "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2603–2615. <https://doi.org/10.2514/1.G002150>
- [23] Wang, Z., and Grant, M. J., "Autonomous Entry Guidance for Hypersonic Vehicles by Convex Optimization," *Journal of Spacecraft and Rockets*, Vol. 55, No. 4, 2018, pp. 993–1006. <https://doi.org/10.2514/1.A34102>
- [24] Wang, Z., and Lu, Y., "Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization," *Journal of Spacecraft and Rockets*, Vol. 57, No. 6, 2020, pp. 1373–1386. <https://doi.org/10.2514/1.A34640>
- [25] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E., "Convex Approach to Three-Dimensional Launch Vehicle Ascent Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 6, 2021, pp. 1116–1131. <https://doi.org/10.2514/1.G005376>
- [26] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E., "Convex Optimization of Launch Vehicle Ascent Trajectory with Heat-Flux and Splash-Down Constraints," *Journal of Spacecraft and Rockets*, Vol. 59, No. 3, 2022, pp. 900–915. <https://doi.org/10.2514/1.A35194>
- [27] Wang, Z., and McDonald, S. T., "Convex Relaxation for Optimal Rendezvous of Unmanned Aerial and Ground Vehicles," *Aerospace Science and Technology*, Vol. 99, April 2020, Paper 105756. <https://doi.org/10.1016/j.ast.2020.105756>
- [28] Shi, Y., Wang, Z., LaClair, T. J., Wang, C., and Shao, Y., "Real-Time Control of Connected Vehicles in Signalized Corridors Using Pseudospectral Convex Optimization," *Optimal Control Applications and Methods*, Vol. 44, No. 4, 2023, pp. 2257–2277. <https://doi.org/10.1002/oca.2978>
- [29] Rice, C. R., McDonald, S. T., Shi, Y., Gan, H., Lee, W. S., Chen, Y., and Wang, Z., "Perception, Path Planning, and Flight Control for a Drone-Enabled Autonomous Pollination System," *Robotics*, Vol. 11, No. 6, 2022, p. 144. <https://doi.org/10.3390/robotics11060144>
- [30] Wang, Z., Wei, P., and Sun, L., "Optimal Cruise, Descent, and Landing of eVTOL Vehicles for Urban Air Mobility Using Convex Optimization," *AIAA SciTech 2021 Forum*, AIAA Paper 2021-0577, 2021. <https://doi.org/10.2514/6.2021-0577>
- [31] Wu, Y., Wang, Z., Benedikter, B., and Zavoli, A., "A Convex Approach to Multi-Phase Trajectory Optimization of eVTOL Vehicles for Urban Air Mobility," *AIAA SCITECH 2022 Forum*, AIAA Paper 2022-2159, 2022. <https://doi.org/10.2514/6.2022-2159>
- [32] Darby, C. L., Hager, W. W., and Rao, A. V., "An Hp-Adaptive Pseudospectral Method for Solving Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 32, No. 4, 2011, pp. 476–502. <https://doi.org/10.1002/oca.957>
- [33] Zhao, D., and Song, Z., "Reentry Trajectory Optimization with Waypoint and No-Fly Zone Constraints Using Multiphase Convex Programming," *Acta Astronautica*, Vol. 137, Aug. 2017, pp. 60–69. <https://doi.org/10.1016/j.actaastro.2017.04.013>
- [34] Nesterov, Y., *Introductory Lectures on Convex Optimization: A Basic Course*, Vol. 87, Springer-Verlag, Berlin, 2013. <https://doi.org/10.1007/978-1-4419-8853-9>
- [35] Lofberg, J., "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," *2004 IEEE International Conference on Robotics and Automation*, Inst. of Electrical and Electronics Engineers, New York, 2004, pp. 284–289. <https://doi.org/10.1109/CACSD.2004.1393890>
- [36] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP Solver for Embedded Systems," *2013 European Control Conference (ECC)*, Inst. of Electrical and Electronics Engineers, New York, 2013, pp. 3071–3076. <https://doi.org/10.23919/ECC.2013.6669541>