AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2484

A Convex Approach to High-fidelity Landing Trajectory Optimization for Advanced Air Mobility

Yufei Wu*, Sabrullah Deniz†, and Zhenbo Wang‡
The University of Tennessee, Knoxville, TN 37996, USA

Daning Huang§
The Pennsylvania State University, University Park, PA 16802, USA

Urban Air Mobility (UAM) presents an innovative solution for intra-urban and inter-urban transportation, promising enhanced flexibility, efficiency, and sustainability. However, the integration of UAM into densely populated city environments brings significant challenges, particularly in the precision landing of multi-rotor vehicles amid complex and dynamic urban landscapes. To address this challenge, our paper introduces a novel convex optimization approach to solve the high-fidelity landing problem of electric vertical take-off and landing (eVTOL) vehicles. In our method, we first conceptualize the eVTOL vehicle landing trajectory optimization as a high-dimensional, highly nonconvex optimal control problem. We then implement a series of convenient convexification techniques to transform this problem into a convex form. The core of our approach lies in the application of sequential convex programming (SCP), an advanced method known for its efficacy and real-time performance in handling complex optimization challenges. We conduct a comparative analysis of our SCP-based solution with results obtained from the GPOPS-II solver, a widely recognized general-purpose tool in optimal control. This comparison not only benchmarks the performance of our method but also highlights its potential advantages in solving complicated, dynamic trajectory optimization problems in the context of UAM.

Nomenclature

A	=	blade area, m^2
c	=	cord length of blade, m
C_D	=	drag coefficient
C_Q	=	torque coefficient
C_T	=	thrust coefficient
C_{roll}	=	roll moment coefficient
C_{pitch}	=	pitch moment coefficient
C_p	=	power coefficient
D	=	drag force, N
F	=	aerodynamic force produced by rotor, N
I_{xx}	=	moment of inertia about the x-axis, kg·m^2
I_{yy}	=	moment of inertia about the y-axis, kg·m^2
I_{zz}	=	moment of inertia about the z-axis, kg·m^2
g	=	gravitational acceleration, m/s^2
l	=	arm length, m
m	=	vehicle's mass, kg
M	=	aerodynamic moment produced by rotor, kg·m^2
N	=	number of blades

*PhD Candidate, Department of Mechanical, Aerospace, and Biomedical Engineering, Student Member AIAA, ywu86@vols.utk.edu
†PhD Student, Department of Mechanical, Aerospace, and Biomedical Engineering, Student Member AIAA, sdeniz@vols.utk.edu
‡Assistant Professor, Department of Mechanical, Aerospace, and Biomedical Engineering, Senior Member AIAA, zwang124@utk.edu
§Assistant Professor, Department of Aerospace Engineering, Member AIAA, daning@psu.edu

R	=	radius of blade, m
$\mathbf{u}$	=	control vector
u	=	tangential component of freestream velocity in rotor plane, m/s
v_i	=	nondimensional uniform steady induced velocity
w	=	vertical component of freestream velocity in rotor plane, m/s
t_f	=	time of flight, s
$\mathbf{x}$	=	state vector
θ	=	pitch angle, deg
ϕ	=	roll angle, deg
ψ	=	yaw angle, deg
ρ	=	atmospheric density, kg/m^3
λ_i	=	nondimensional inflow distribution
Ω_i	=	angular velocity of rotor i, rad/s
τ_x, τ_y, τ_z	=	roll, pitch, and yaw torques with respect to the vehicle body frame
δ	=	trust-region radius
ε	=	tolerance
ν	=	virtual control vector

I. Introduction

Equipped with the latest technologies such as electric vertical take-off and landing (eVTOL) vehicles, urban air mobility (UAM) is emerging as a revolutionary mode of transportation in congested urban areas. These eVTOL vehicles, characterized by their ability to ascend and descend vertically, promise a novel solution for alleviating ground traffic congestion, thereby reshaping the landscape of urban transportation [1–5]. This innovative approach to urban transit is not only seen as a potential game-changer for daily commutes but also as a pivotal element in emergency response and urban logistics. However, despite these promising developments, deployment and operation of eVTOL vehicles in UAM missions face significant challenges.

Critical factors, such as energy capacity and the need to navigate complex urban environments safely, are constraining successful operation of eVTOL vehicles for UAM missions. Unlike traditional aircraft, eVTOLs rely on battery technology, which, despite rapid advancements, still poses limitations in terms of energy density and weight. Consequently, the optimization of flight trajectories becomes a crucial concern in the application and deployment of eVTOLs for UAM. Trajectory optimization involves a comprehensive consideration of various mission objectives, which may range from passenger transport to cargo delivery, each with its own set of requirements and constraints. Depending on the flight missions, each eVTOL flight may include different flying phases, such as take-off, ascending, cruising, descent, and landing, each demanding specific flight paths and energy usage strategies. Moreover, the dynamic and often unpredictable nature of urban environments, combined with varying weather conditions, require a flexible and robust trajectory optimization approach that can calculate safe and efficient flying paths in real-time. Therefore, developing rigorous yet easily implementable trajectory optimization methods is crucial for ensuring the efficient and safe performance of eVTOL aircraft in diverse UAM mission operations.

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 [6–9] or various optimization methods [10–14] 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 [15, 16].

The existing research largely relies on simplified dynamic models ignoring the deviations in aircraft behavior that can occur under real-world urban conditions, leading 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 [17, 18]. 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 paper, 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 obtain the solution to the landing trajectory optimization problem and compare the results with those from other existing solvers.

The rest of this paper is structured as follows. We introduce the problem formulation in Section II, which includes thrust coefficient calculation, rotor inflow model, ODE-based rotor dynamics, and 6-DoF flight dynamics. In Section III, we provide our convexification procedure that reformulates the original problem into a convex problem and develop a SCP algorithm to solve for an approximate optimal solution. Numerical simulation results are provided in Section IV to demonstrate the performance of our proposed method. Lastly, we summarize the work in Section V.

II. Problem Formulation

In this section, we integrate the ODE-based aerodynamic rotor models with the 6-DoF eVTOL vehicle flight dynamics model to formulate an optimal landing control problem. The first subsection introduces the calculation of the thrust coefficient for each rotor. In the second subsection, we present the rotor inflow model used in this paper. In the third subsection, we formulate the ODE-based model for rotor dynamics. In the fourth and fifth subsections, we present the 6-DoF eVTOL flight dynamic model and the essential flight constraints, respectively. Finally, we introduce the performance index and formulate the trajectory optimization problem for landing.

A. Thrust Coefficient of a Single Rotor

The thrust coefficient of a rotor blade is a fundamental parameter for aerodynamic performance and control of eVTOL vehicles. It determines how efficiently a rotor converts rotational power into thrust, which directly impacts the vehicle's lifting capability and overall performance. In this section, we employ the blade element momentum theory (BEMT) to calculate the lift and thrust of a single rotor blade. BEMT is a widely-recognized and established method in rotor aerodynamics, which allows for a detailed and accurate calculation of the lift and thrust generated by rotor blades. 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 each element [19].

As illustrated in Figs. 1 and 2, the geometric positioning of each blade element is defined by its radial (r) and azimuthal (φ) coordinates. Key to our analysis are the free-stream velocity components, i.e., the tangential component, u, and the perpendicular component, w. Then, the advance 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 is created when generating thrust, resulting in an inflow distribution $\lambda_i(r, \varphi)$.

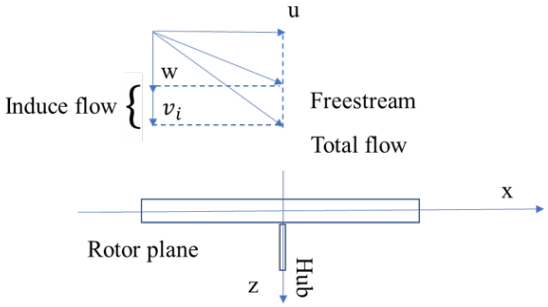

Fig. 1 BEMT for single rotor (side view).

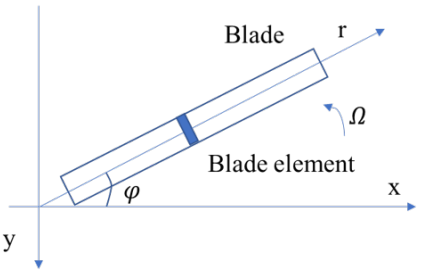

Fig. 2 BEMT for single rotor (top view).

For a blade element at location (r, φ), the local total free-stream speed is defined as:

$$\frac{u_\infty}{\Omega R} = \sqrt{(r/R + \mu_x \sin\varphi)^2 + \lambda_i(r,\varphi)^2} \equiv \bar{u}_\infty(r,\varphi) \tag{1}$$

and the local lift and drag are:

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

where the effective angle of attack is:

$$\vartheta = \alpha - \tan^{-1}\frac{\lambda_i}{\mu_x} \equiv \alpha - k \tag{3}$$

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

$$dF_z = dL\cos k - dD\sin k \tag{4}$$

$$dF_x = -dL\sin k + dD\cos k \tag{5}$$

In this paper, we assume that the drag term $dD\sin k$ is small compared to the lift term. We also assume a 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:

$$C_T(\mu_x, \mu_z, \lambda_i) = \sigma \int_0^{2\pi}\int_0^1 [\bar{u}_\infty^2(k-\alpha)\cos k]\, d\bar{r} d\varphi \tag{6}$$

B. Rotor Inflow Model

The rotor inflow model is a crucial component in the aerodynamic analysis and optimization of eVTOL vehicles, especially when considering the dynamic and complex urban environments in which the vehicles operate. In the context of eVTOL trajectory optimization, including inflow modeling in the overall system enables more precise trajectory calculation, especially during critical maneuvers such as landing and takeoff. In this paper, the rotor inflow model is adapted from [20] and was developed by Pitt and Peters in [21] using a system of linear ODEs to describe the dynamic inflow of a rotor disk. The model relates the dimensionless aerodynamic loading to the dimensionless induced flow distribution, which is described by the equation below:

$$\lambda(\bar{r}, \psi, t) = \lambda_0(t) + \lambda_s(t)\bar{r}\sin\psi_r + \lambda_c(t)\bar{r}\cos\psi_r \tag{7}$$

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

$$\lambda = u_x/\Omega R \tag{8}$$

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

$$\begin{bmatrix}\dot{\lambda_0}\\ \dot{\lambda_s}\\ \dot{\lambda_c}\end{bmatrix} = -M^{-1}L^{-1}\begin{bmatrix}\lambda_0\\ \lambda_s\\ \lambda_c\end{bmatrix} + M^{-1}\begin{bmatrix}C_T\\ C_{roll}\\ C_{pitch}\end{bmatrix} \tag{9}$$

with M and L coefficient matrices defined as:

$$L = \frac{1}{\bar{V}_\infty} \begin{bmatrix} \frac{1}{2} & 0 & \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} \\ 0 & -\frac{4}{1+\sin\alpha} & 0 \\ \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} & 0 & -\frac{4\sin\alpha}{1+\sin\alpha} \end{bmatrix} \tag{10}$$

$$M = \begin{bmatrix} \frac{128}{75\pi} & 0 & 0 \\ 0 & \frac{-16}{45\pi} & 0 \\ 0 & 0 & \frac{-16}{45\pi} \end{bmatrix} \tag{11}$$

The coefficients C_T, C_{roll}, C_{pitch} respectively represent the thrust coefficient, and the torque coefficients in the roll and pitch directions.

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:

$$\bar{V}_\infty = \frac{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}{2\pi n R}, \quad \alpha = \pi/2 - \cos^{-1}\left(\frac{[\dot{x}, \dot{y}, \dot{z}] \cdot \mathbf{n}_{rotordisk}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}\right) \tag{12}$$

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 9 defines the inflow dynamics for a single rotor. In the next subsection, we will introduce the state-space model for the four-motor inflow aerodynamic system considering a quadrotor vehicle configuration.

C. ODE Model of Rotor Dynamics

Following the inflow dynamic model for a single rotor disk, we can construct the state-space model for all rotors to calculate the thrust, torques, and dynamic inflow states for each motor. The output of this system will feed into the vehicle dynamic model, which will be discussed in the next subsection. In this paper, we consider the states of the system to include the inflow states of each rotor, along with the total vertical thrust F_z, the sum of the moments with respect to the x-, y-, and z-axes, denoted as τ_x, τ_y, τ_z, respectively, together forming the state vector $\mathbf{x}_0 \in \mathbb{R}^{16x1}$:

$$\mathbf{x}_0 = [\lambda_{0,r1}, \lambda_{s,r1}, \lambda_{c,r1}, \ldots, \lambda_{0,r4}, \lambda_{s,r4}, \lambda_{c,r4}, F_z, \tau_x, \tau_y, \tau_z]^T \tag{13}$$

where $\lambda_{0,ri}, \lambda_{s,ri}, \lambda_{c,ri}$ are the inflow states of i-the rotor. The terms F_z, τ_x, τ_y, and τ_z are defined as follows:

- F_z is the total vertical thrust generated by the rotors, a key output influenced by the speed of each rotor.
- τ_x, τ_y, and τ_z are the sum of the moments (torques) with respect to the x-, y-, and z-axes respectively, resulting from the combined action of all rotors.

. The control vector $\mathbf{u}_0 \in \mathbb{R}^{4x1}$ is considered to be the square of each motor's rotational speed:

$$\mathbf{u}_0 = [\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T \tag{14}$$

where Ω_i, $i = 1, 2, 3, 4$ is the rotational speed of each rotor i in revolution per second. Consequently, we can formulate the state-space form of the aerodynamic system as follows:

$$\dot{\mathbf{x}}_0 = A_0\mathbf{x}_0 + B_0\mathbf{u}_0 \tag{15}$$

where A_0 is as follows:

$$A_0 = \begin{bmatrix} -M^{-1}L^{-1} & & & & \\ & -M^{-1}L^{-1} & & & \\ & & -M^{-1}L^{-1} & & \\ & & & -M^{-1}L^{-1} & \\ & & & & -I^{4x4} \end{bmatrix} \tag{16}$$

The control matrix B_0 is defined below:

$$B_0 = \begin{bmatrix} 0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1} \\ b_1 & b_2 & b_3 & b_4 \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} & -(b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) & -(b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) \\ k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \end{bmatrix} \tag{17}$$

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

In order for the model to be solved by optimal control algorithms more efficiently, two assumptions are made in this work. First, the drag coefficient C_T and thrust coefficients C_{roll}, C_{pitch}, and C_{yaw} are assumed to be constant in the control matrix at low-speed landing scenarios. Second, the angle of attack, α, in the equations above is assumed to be constant as well. The resulting aerodynamic model will be combined with the flight dynamics to formulate the landing trajectory optimization problem in the following subsections.

D. Flight Dynamics

The flight dynamics of eVTOL vehicles form a foundational aspect of trajectory optimization. In this paper, our focus is on a comprehensive 6-DoF flight dynamic model. The 6-DoF model captures the complete range of motion possible for the eVTOL vehicle, including three translational movements (up/down, left/right, forward/backward) and three rotational movements (pitch, roll, yaw), and the 12 states are defined as:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^T \in \mathbb{R}^{12} \tag{18}$$

where x, y, z represent the displacement of the mass center of the vehicle in the Earth reference frame (North, East and Down); 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:

$$\mathbf{u} = [F_z \quad \tau_x \quad \tau_y \quad \tau_z]^T \tag{19}$$

for simplicity, the Coriolis term is neglected, and the model in the inertial frame can be represented as:

$$\begin{aligned} \ddot{x} &= -\frac{F_z}{m}[\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)] \\ \ddot{y} &= -\frac{F_z}{m}[\cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi)] \\ \ddot{z} &= g - \frac{F_z}{m}[\cos(\phi)\cos(\theta)] \\ \ddot{\phi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_x}{I_x} \\ \ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_y}{I_y} \\ \ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{\tau_z}{I_z} \end{aligned} \tag{20}$$

We rewrite the system in a state-space form as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \mathbf{f}(\mathbf{x}) + B(\mathbf{x})\mathbf{u} \tag{21}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ 0 \\ 0 \\ g \\ C_{Ix}qr \\ C_{Iy}pr \\ C_{Iz}pq \end{bmatrix} \tag{22}$$

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

The control matrix is defined as:

$$B(\mathbf{x}) = \begin{bmatrix} 0^{6\times1} & 0^{6\times1} & 0^{6\times1} & 0^{6\times1} \\ g_1 & 0 & 0 & 0 \\ g_2 & 0 & 0 & 0 \\ g_3 & 0 & 0 & 0 \\ 0 & l/I_x & 0 & 0 \\ 0 & 0 & l/I_y & 0 \\ 0 & 0 & 0 & 1/I_z \end{bmatrix} \tag{23}$$

where

$$\begin{aligned} g_1 &= -\frac{1}{m}[\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)] \\ g_2 &= \frac{1}{m}[\cos(\psi)\sin(\phi) - \cos(\phi)\sin(\psi)\sin(\theta)] \\ g_3 &= -\frac{1}{m}[\cos(\phi)\cos(\theta)] \end{aligned} \tag{24}$$

E. Flight Constraints

In this paper, 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:

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{25}$$

$$\mathbf{x}(t_f) = \mathbf{x}_f \tag{26}$$

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

$$0 \le x \le x_{max} \tag{27}$$

$$0 \le y \le y_{max} \tag{28}$$

$$0 \le z \le z_{max} \tag{29}$$

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

$$\Omega_{min} \le \Omega_i(t) \le \Omega_{max} \tag{30}$$

where the subscript i denotes the motor number. In the model considered in this paper, we have four motors.

F. Performance Index and Optimal Control Problem

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

$$J = \int_{t_0}^{t_f} \sum_{i=1}^{4} \Omega_i^2(t) dt \tag{31}$$

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:

$$\begin{gathered} \text{Minimize:} \quad (31) \\ \text{Subject to:} \quad (15), (21), (25), (26), (27), (28), (29), (30) \end{gathered}$$

Overall, we want to minimize the objective function (31) while satisfying the rotor aerodynamics (15), the flight dynamics (21), the boundary conditions (25) and (26), and the inequality state and control constraints (27)-(30).

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

III. Convex Approach

In recent years, convex optimization, which studies the problem of minimizing convex functions over convex sets, has been widely used to solve optimal control problems in aerospace engineering due to the advantages of fast convergence and guaranteed global minima for a single convex optimization problem, such as linear programming (LP), quadratic programming (QP), second-order cone programming (SOCP), or semidefinite programming (SDP). In these cases, the problem can be solved in polynomial time because of its low complexity. As such, convex optimization approaches are an ideal method for onboard real-time applications in complex engineering fields, such as eVTOL trajectory optimization.

However, two significant challenges arise when applying convex approaches: identifying a convex problem and transforming a nonconvex problem into a convex problem. Usually, nonconvex constraints in the optimal control problem are convexified by introducing slack variables and constraint relaxations, and the highly nonlinear dynamic equations are often replaced by approximated formula, typically derived from the first-order Taylor series approximation [22].

A. Modified Model for Convex Approach

In this paper, we introduce a critical assumption to simplify the complex aerodynamics involved in eVTOL flight. We assume that the aerodynamic forces and moments have a linear relationship with the rotor speeds, allowing us to reduce the overall system to a total of 16 states. This reduction leads to four aerodynamic states that linearly relate rotor speed to the thrust and torques in the x, y, and z directions, along with the 12 standard flight dynamic states. After the simplification the overall states are defined as:

$$\mathbf{X} = \begin{bmatrix} F_z & \tau_x & \tau_y & \tau_z & x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^T \in \mathbb{R}^{16} \tag{32}$$

The new system in a state-space form as follows:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = \mathbf{F}(\mathbf{X}) + \tilde{B}(\mathbf{X})\mathbf{U} \tag{33}$$

where:

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} -F_z \\ -\tau_x \\ -\tau_y \\ -\tau_z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ 0 \\ 0 \\ g \\ C_{Ix}qr \\ C_{Iy}pr \\ C_{Iz}pq \end{bmatrix} \tag{34}$$

And the updated control matrix is defined as:

$$\tilde{B}(\mathbf{X}) = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} & -(b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) & -(b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) \\ k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \\ 0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1} \end{bmatrix} \tag{35}$$

This assumption significantly simplifies the mathematical model while retaining sufficient accuracy for practical application, making it more tractable for convex optimization. By adopting this approach, we ensure that the resulting trajectory optimization problem remains within a computationally feasible domain. This balance is vital for the development of real-time, onboard control systems that can reliably and efficiently operate in dynamic and often unpredictable UAM environments.

B. Convexification

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

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = A(\mathbf{X})\mathbf{X} + \tilde{B}\mathbf{U} + \mathbf{g}^* \tag{36}$$

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

$$A(\mathbf{X}) = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ g_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & l/I_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Ix}r & C_{Ix}q & 0 & 0 \\ 0 & 0 & l/I_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iy}r & 0 & C_{Iy}p & 0 \\ 0 & 0 & 0 & l/I_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iz}q & C_{Iz}p & 0 & 0 \end{bmatrix} \tag{37}$$

$$\mathbf{g}^* = \begin{bmatrix} 0^{1x12} & g & 0 & 0 & 0 \end{bmatrix}^T \tag{38}$$

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

$$\tilde{B} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} & -(b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll}) & b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{roll} \\ b_1 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & b_2 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch} & -(b_3 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) & -(b_4 \cdot \frac{\sqrt{2}}{2} l_{arm} + k_{pitch}) \\ k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \\ 0^{12x1} & 0^{12x1} & 0^{12x1} & 0^{12x1} \end{bmatrix} \tag{39}$$

To convexify the system, we further write the above equation as follows by separating the linear and nonlinear components:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) = A_L \mathbf{X} + \tilde{B}\mathbf{U} + \mathbf{f}_{NL}(\mathbf{X}, t) + \mathbf{g}^* \tag{40}$$

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

$$A_L = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & l/I_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & l/I_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{41}$$

The nonlinear part $\mathbf{f}_{NL}$ is defined as:

$$\mathbf{f}_{NL}(\mathbf{X}, t) = \begin{bmatrix} 0^{22\times 1} \\ F_z g_1 \\ F_z g_2 \\ F_z g_3 \\ C_{Ix} qr \\ C_{Iy} pr \\ C_{Iz} pq \end{bmatrix} \tag{42}$$

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

$$\mathbf{f}_{NL}(\mathbf{X}, t) \approx \mathbf{f}_{NL}\left(\mathbf{X}^*, t\right) + \frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{X}}\left(\mathbf{X}^*, t\right)\left(\mathbf{X} - \mathbf{X}^*\right) \tag{43}$$

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

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) \approx A_L \mathbf{X} + \tilde{B}\mathbf{U} + \mathbf{g}^* + \mathbf{f}_{NL}\left(\mathbf{X}^*, t\right) + \frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{X}}\left(\mathbf{X}^*, t\right)\left(\mathbf{X} - \mathbf{X}^*\right) \tag{44}$$

where the partial derivative matrices are shown below:

$$\frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{X}} = \begin{bmatrix} 0^{10\times 1} & & \cdots & \cdots & \cdots & & & \cdots & \cdots & \cdots & & \cdots & \cdots & \cdots & & 0^{10\times 1} \\ g_1^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_1}{\partial \phi} & \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\ g_2^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_2}{\partial \phi} & \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\ g_3^* & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial g_3}{\partial \phi} & \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial \psi} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Ix} r & C_{Ix} q \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iy} r & 0 & C_{Iy} p \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{Iz} q & C_{Iz} p & 0 \end{bmatrix} \tag{45}$$

with

$$\begin{bmatrix} \frac{\partial g_1}{\partial \phi} & \frac{\partial g_1}{\partial \theta} & \frac{\partial g_1}{\partial \psi} \\ \frac{\partial g_2}{\partial \phi} & \frac{\partial g_2}{\partial \theta} & \frac{\partial g_2}{\partial \psi} \\ \frac{\partial g_3}{\partial \phi} & \frac{\partial g_3}{\partial \theta} & \frac{\partial g_3}{\partial \psi} \end{bmatrix} = \frac{F_z}{m} \begin{bmatrix} -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & -\cos\psi\cos\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\sin\psi\cos\theta\cos\phi & -\sin\psi\sin\phi - \cos\psi\sin\theta\cos\phi \\ \cos\theta\sin\phi & \sin\theta\cos\phi & 0 \end{bmatrix} \tag{46}$$

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

$$\mathbf{X}_i = \mathbf{X}_{i-1} + \frac{\Delta t}{2}\left(\dot{\mathbf{X}}_{i-1} + \dot{\mathbf{X}}_i\right) \tag{47}$$

where Δt is the step size. After rearranging all the terms, we obtain the following discretized state equation:

$$\begin{aligned} &\left[\left(I - \frac{\Delta t}{2}(A_L^{k-1} + A_{NL_i}^{k-1})\right)\mathbf{X}_i\right] - \left[\left(I + \frac{\Delta t}{2}(A_L^{k-1} + A_{NL_{i-1}}^{k-1})\right)\mathbf{X}_{i-1}\right] - \frac{\Delta t}{2}\tilde{B}\mathbf{U}_{i-1} - \frac{\Delta t}{2}\tilde{B}\mathbf{U}_i \\ &= \frac{\Delta t}{2}\left[\left(\mathbf{f}_{NLi-1}^{k-1} - A_{NL_{i-1}}^{k-1}\mathbf{X}_{i-1}^{k-1} + \mathbf{g}^*\right) + \left(\mathbf{f}_{NL_i}^{k-1} - A_{NL_i}^{k-1}\mathbf{X}_i^{k-1} + \mathbf{g}^*\right)\right] \end{aligned} \tag{48}$$

Finally, we formulate the convex form of the landing problem as:
Problem 2:

$$\begin{aligned} &\text{Minimize:} \quad (31) \\ &\text{Subject to:} \quad (25) - (30), (48) \end{aligned}$$

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

C. Virtual Control

During the process of convexification, one big 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 $\boldsymbol{\nu}$ 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 $\boldsymbol{\nu}$ usually has the same length as the state vector. With the virtual control the state equation will become:

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) \approx A_L\mathbf{X} + \tilde{B}\mathbf{U} + \mathbf{g}^* + \mathbf{f}_{NL}\left(\mathbf{X}^*, t\right) + \frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{X}}\left(\mathbf{X}^*, t\right)\left(\mathbf{X} - \mathbf{X}^*\right) + E\boldsymbol{\nu} \tag{49}$$

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:

$$J = \int_{t_0}^{t_f} \sum_{i=1}^{4} \Omega_i^2(t)dt + w_\nu \cdot \text{ess sup}\|E(t)\boldsymbol{\nu}(t)\|_1 \tag{50}$$

D. Sequential Convex Programming

Sequential convex programming (SCP) stands out for its ability to handle complex, nonlinear problems through a series of iterative steps that gradually converge to a solution. In this paper, for the first time, an SCP method is used to address the high-fidelity, nonconvex landing problem of eVTOL vehicles. SCP operates by transforming the nonconvex problem into a series of local convex optimization problems. Each iteration solves the convex subproblem based on the current solution estimate, leading to a progressively more accurate solution. The methodology of SCP implemented in our research is summarized in the following key 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
$$\sup \, |\mathbf{X}_i^{(k)} - \mathbf{X}_i^{(k-1)}| \leq \varepsilon \tag{51}$$
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 $\{\mathbf{X}^{(k)}, \mathbf{U}^{(k)}\}$.

By employing SCP, we aim not only to navigate the complexities of the eVTOL landing problem efficiently but also to explore its potential for real-time applications. This capability is vital in UAM, where flight conditions can change rapidly, and the ability to quickly adjust landing trajectories in response to these changes is crucial for safety and efficiency.

IV. Numerical Simulations

In this paper, a small quadrotor serves as an example model for the simulations to emulate the UAM landing scenario. The vehicle parameters are listed in Table 1. These parameters define the basic aspects of the vehicle such as mass, rotor configuration, and aerodynamic coefficients. Using this model, several cases have been solved to demonstrate the performance of the proposed method as follows.

For the first landing scenario, the initial and terminal conditions are outlined in Table 2. These conditions shows the starting and the desired end states of the quadrotor during a landing maneuver. To initialize the SCP algorithm, we generated an initial trajectory using linear interpolation between these initial and final states. This method provides a starting point that is sufficiently close to the desired trajectory to begin the iterative process of refinement.

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 (Problem 1) is solved. Both the 12-state and 16-state models have been solved using the GPOPS-II solver, a well-established tool in optimal control [23]. The results obtained from our SCP-based method and the GPOPS-II 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.

Table 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 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

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 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 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 5 presents the convergence process for the objective function, where we can see that the SCP method converges to a stable objective value. Figures 6 to 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.

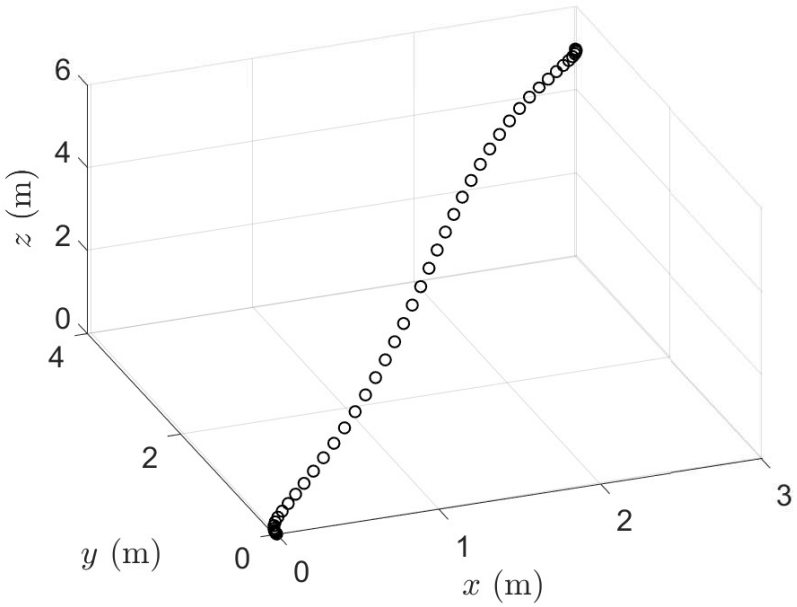

Fig. 3 Three-dimensional 12-state optimal landing trajectory by SCP.

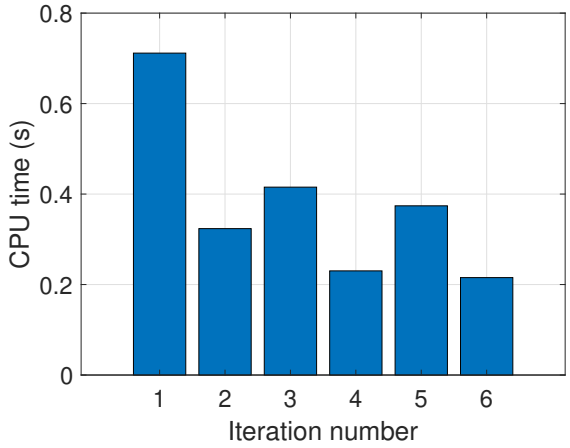

Fig. 4 Computational time of each subproblem for SCP algorithm.

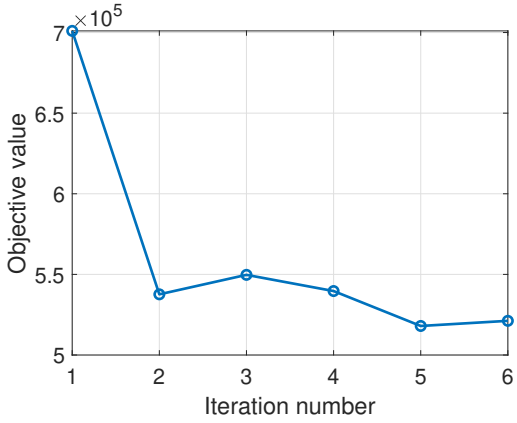

Fig. 5 SCP convergence profile for objective function.

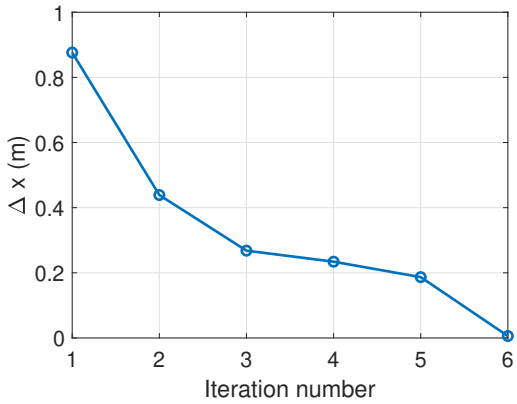

Fig. 6 SCP convergence profile for x direction.

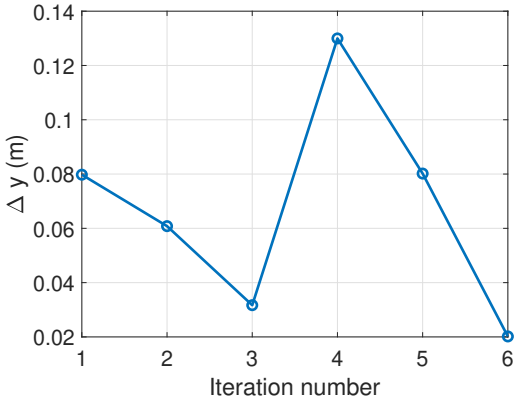

Fig. 7 SCP convergence profile for y direction.

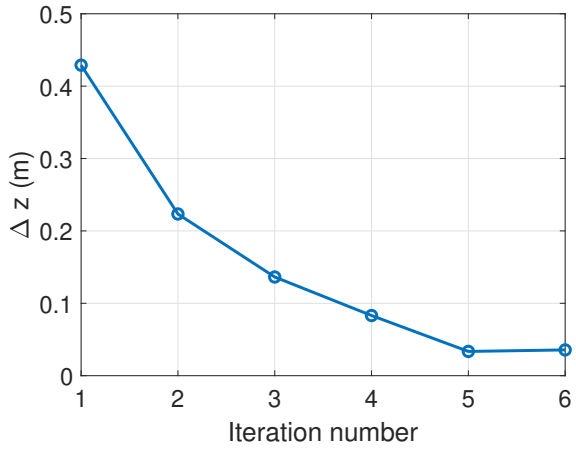

Fig. 8 SCP convergence profile for z direction.

Figure 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 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 Fig. 11, we present a side-by-side comparison of the thrust curves derived 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 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 13 and 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.

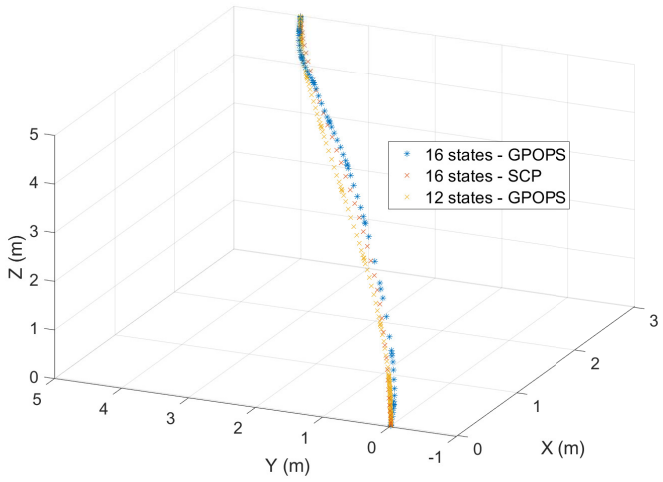

Fig. 9 Trajectory comparison between GPOPS and SCP.

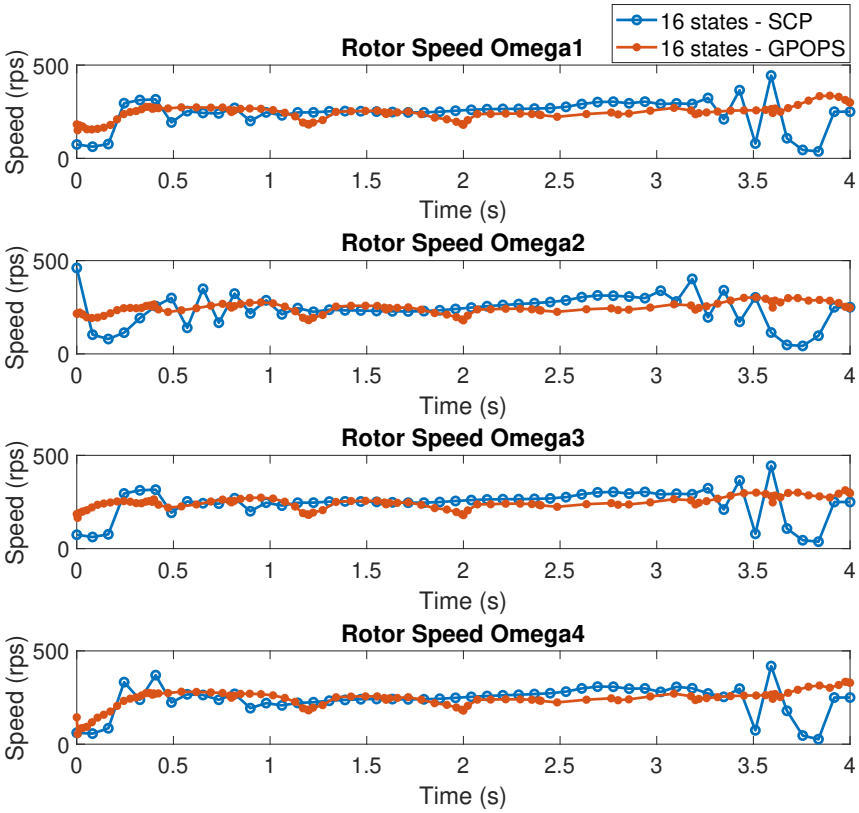

Fig. 10 Rotor speed comparison between GPOPS and SCP.

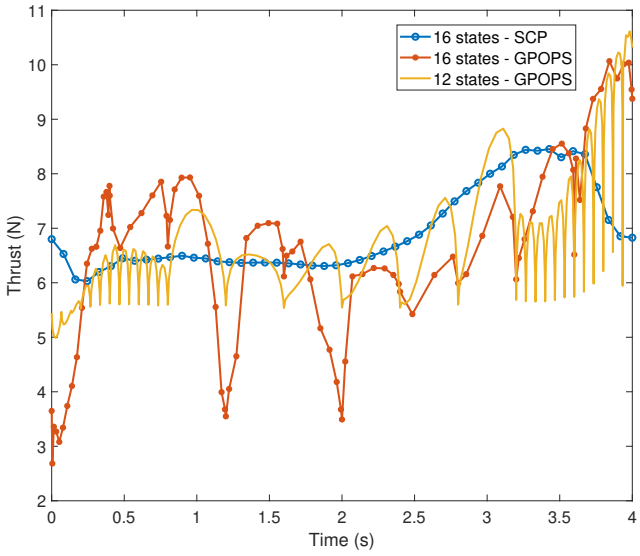

Fig. 11 Thrust comparison between GPOPS and SCP.

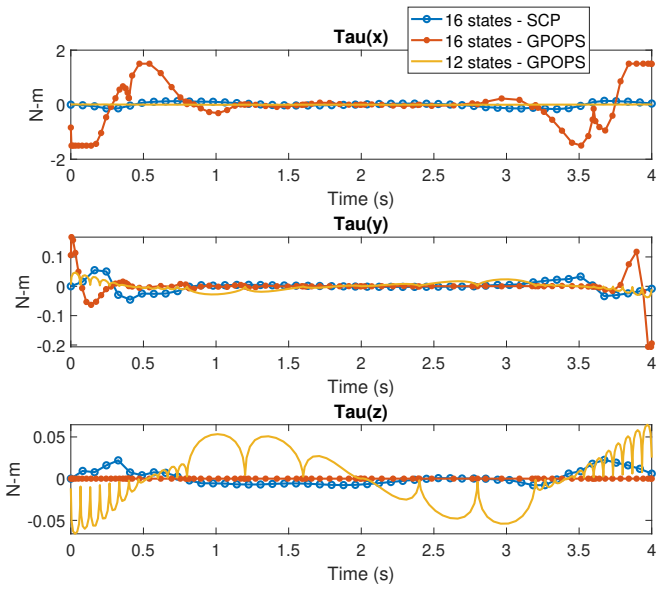

Fig. 12 Torques comparison between GPOPS and SCP.

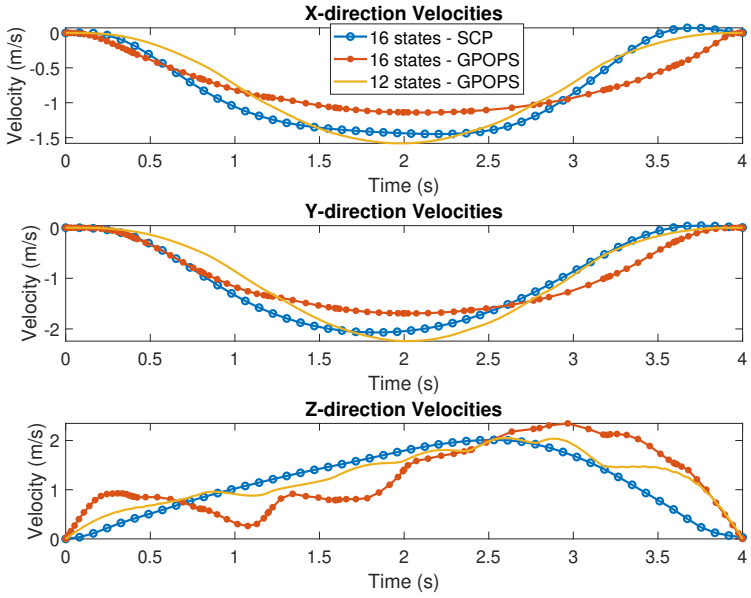

Fig. 13 Velocity profile comparison between GPOPS and SCP.

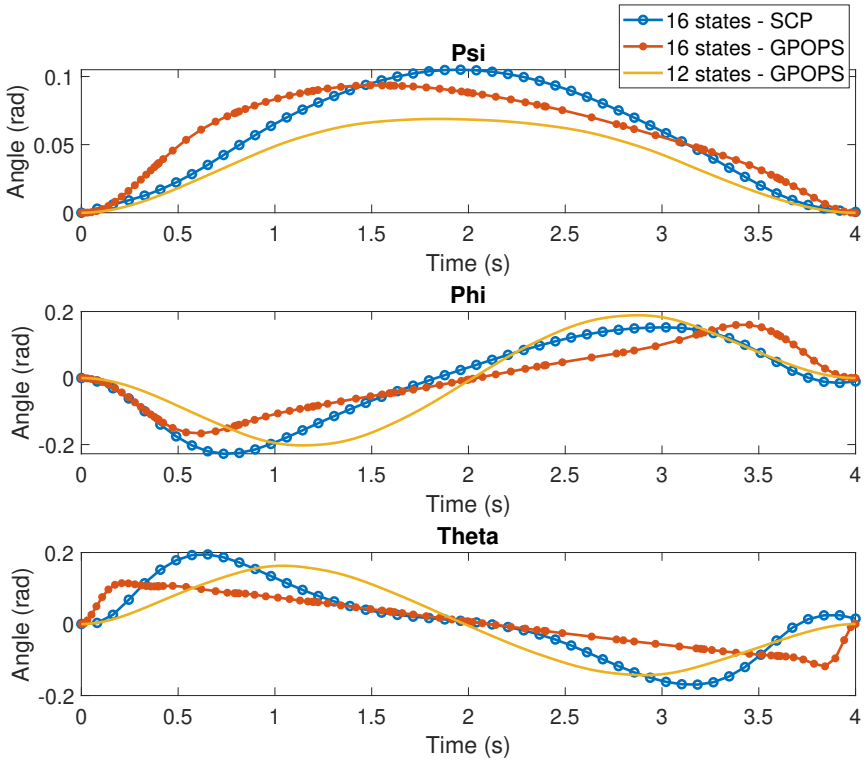

Fig. 14 Angular position comparison between GPOPS and SCP.

Table 3 Comparison of computational time of different solvers

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

Lastly, Table 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.

V. Conclusions

In this paper, we investigate the possibility of integrating high-fidelity, ODE-based aerodynamic models with the conventional flight dynamics to achieve a more accurate and realistic representation of the eVTOL vehicle's behavior during landing maneuvers. 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.

References

[1] 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*, 2023.

[2] Polaczyk, N., Trombino, E., Wei, P., and Mitici, M., "A review of current technology and research in urban on-demand air mobility applications," *8th Biennial Autonomous VTOL Technical Meeting and 6th Annual Electric VTOL Symposium*, 2019, pp. 333–343.

[3] Kim, H. D., Perry, A. T., and Ansell, P. J., "A review of distributed electric propulsion concepts for air vehicle technology," *2018 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2018, pp. 1–21.

[4] Garrow, L. A., German, B. J., and Leonard, C. E., "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*, Vol. 132, 2021, p. 103377.

[5] Elevate, U., "Fast-forwarding to the future of on-demand, urban air transportation [R/OL]," 2016.

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

[7] Deniz, S., and Wang, Z., "A Multi-Agent Reinforcement Learning Approach to Traffic Control at Future Urban Air Mobility Intersections," *AIAA SCITECH 2022 Forum*, 2022, p. 1509.

[8] Deniz, S., Wu, Y., Shi, Y., and Wang, Z., "A Multi-Agent Reinforcement Learning Approach to Traffic Control at Merging Point of Urban Air Mobility," *AIAA AVIATION 2022 Forum*, 2022, p. 3912.

[9] Deniz, S., Wu, Y., Shi, Y., and Wang, Z., "Autonomous Landing of eVTOL Vehicles via Deep Q-Networks," *AIAA AVIATION 2023 Forum*, 2023.

[10] 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.

[11] 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*, 2021, p. 0577.

[12] McDonald, L., Wu, Y., Deniz, S., and Wang, Z., "Real-Time Generation of Comfort-Optimal Flight Trajectories for Urban Air Mobility Missions," *AIAA SCITECH 2022 Forum*, 2022, p. 2157.

[13] 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*, 2022, p. 2159.

[14] 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*, 2022, p. 3319.

[15] Marconi, L., Isidori, A., and Serrani, A., "Autonomous vertical landing on an oscillating platform: an internal-model based approach," *Automatica*, Vol. 38, No. 1, 2002, pp. 21–32.

[16] Ngo, T. D., and Sultan, C., "Model predictive control for helicopter shipboard operations in the ship airwakes," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 574–589.

[17] Wang, Z., Wu, Y., and Huang, D., "Optimal Landing Control of eVTOL Vehicles Using ODE-Based Aerodynamic Model," *AIAA SCITECH 2023 Forum*, 2023, p. 1742.

[18] Wu, Y., Deniz, S., Shi, Y., Wang, Z., and Huang, D., "Precision Landing Trajectory Optimization for eVTOL Vehicles with High-Fidelity Aerodynamic Models," *AIAA AVIATION 2023 Forum*, 2023, p. 3409.

[19] Leishman, G. J., *Principles of helicopter aerodynamics with CD extra*, Cambridge University Press, 2006.

[20] Cao, P., Hwang, J. T., Bewley, T., and Kuester, F., "Mission-Oriented Trajectory Optimization for Search-and-Rescue Multirotor UAVs in Cluttered and GPS-Denied Environments," *AIAA AVIATION 2022 Forum*, 2022, p. 3999.

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

[22] Wang, Z., "A Survey on Convex Optimization for Guidance and Control of Vehicular Systems," *arXiv preprint arXiv:2311.05115*, 2023.

[23] Patterson, M. A., and Rao, A. V., "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)*, Vol. 41, No. 1, 2014, pp. 1–37.