Convex Optimization-Based Trajectory Planning for Quadrotors Landing on Aerial Vehicle Carriers

Zhipeng Shen, Guanzhong Zhou, Hailong Huang, Chao Huang, Yutong Wang, Fei-Yue Wang

***Abstract*—This paper presents a novel trajectory planning algorithm for quadrotors landing on aerial vehicle carriers (AVCs). The algorithm involves a quadrotor trajectory planning method based on the lossless convexification (LC) theory and a sequential convex programming (SCP) method enabling quadrotors to autonomously land on both static and moving AVCs in a three-dimensional space. By incorporating landing cone constraints, the safety of the quadrotor during landing is ensured. The LC method transforms the original nonconvex optimal control problem (OCP) into a convex optimization problem, enabling the efficient computation of a 3-degree-of-freedom (3-DoF) safe landing trajectory. The designed SCP algorithm utilizes the 3-DoF trajectory as an initial guess and iteratively solves convex subproblems to obtain a safe, agile, and accurate landing trajectory for the complete 6-DoF quadrotor dynamics. Real-world experiments validate the effectiveness and real-time performance of the proposed method.**

***Index Terms*—Motion and trajectory planning, convex optimization, quadrotor, autonomous landing, aerial vehicle carrier.**

I. Introduction

Traditionally, intelligent vehicles (IVs) have primarily referred to ground vehicles [1]–[5]. However, given the remarkable advancements in aerial vehicles [6]–[8], it has become imperative to consider the future of aerial IVs. As a result, the possibility of aerial vehicle carriers (AVCs) has surfaced, wherein aerial IVs play a significant role [9]. These AVCs can transport multiple small unmanned aerial vehicles (UAVs) to specific target areas for various tasks such as surveillance [10], [11], package delivery [12], [13], and search and rescue operations [14], [15]. Upon completion of their tasks, the small UAVs can autonomously return to the AVC (see Fig. 1), which compensates for their limited operation range. To facilitate this process, one of the critical technologies is trajectory planning for UAVs to safely land on AVCs. This article proposes a trajectory planning algorithm specifically designed for quadrotors, incorporating both 3-degree-of-freedom (3-DoF) and 6-degree-of-freedom (6-DoF) approaches, which ensures the safe, accurate, and agile landing of quadrotors on AVCs.

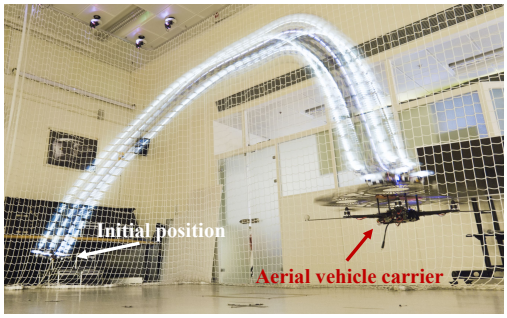


Fig. 1. A small quadrotor landing on an AVC.

A. Related Work

1) Drone Landing on Platforms: Extensive research has been conducted on the topic of drone landing, but most studies have primarily focused on platforms that operate on the ground or water surfaces [16]–[19]. Notably, [16] developed a nonlinear controller to achieve successful hover and landing on a mobile platform for vertical take-off and landing (VTOL) UAVs. Additionally, [17] achieved the landing on a ground-inclined mobile platform. Similarly, [18] designed perception, control, and trajectory planning algorithms to achieve landing on ground-moving vehicles. Another study [19] successfully implemented a safe landing that takes potential hazards into account during ground landing. In contrast, this article focuses on addressing the planning and control challenges specifically related to quadrotors landing on aerial platforms. As opposed to previous studies that only landed on surface platforms, this work aims to achieve secure, agile, and accurate landings on moving aerial platforms, i.e., the AVCs, which are highly dynamic and unstable.

2) Drone Trajectory Planning: The trajectory planning of drones has always been a popular topic, with various methods being explored, such as the polynomial-based methods [20]–[22], the searching and sampling-based methods [23]–[26], the learning-based methods [27]–[29], and the optimization-based methods [30]–[32]. The use of continuous polynomials in trajectory planning based on the differential-flatness properties of quadcopters allows for high computational efficiency [20]–[22]. Another approach uses searching and sampling methods [23]–[26], which easily becomes computationally intractable

This work was partially supported by the Research Institute for Sports Science and Technology [P0043566] and Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles (ICRI-IACV). *Corresponding author: Hailong Huang.*

Z. Shen, G. Zhou, and H. Huang are with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, China (e-mail: {zhipeng.shen, guanzhong.zhou}@connect.polyu.hk and hailong.huang@polyu.edu.hk).

Chao Huang is with the Department of Industrial and Systems Engineering, the Hong Kong Polytechnic University, Hong Kong. (E-mail: hchao.huang@polyu.edu.hk).

Yutong Wang and Fei-Yue Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. Fei-Yue Wang is also with the Institute of Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: yutong.wang@ia.ac.cn, feiyue@ieee.org).

due to the curse of dimensionality, especially when the scale of the state space is large [32]. Optimization-based methods have gained significant attention due to their ability to generate high-quality trajectories. In [31], a hybrid maximum principle based on Pontryagin's Maximum Principle (PMP) was used to derive optimality conditions for the unconstrained minimum control effort of multicopter systems, resulting in a new trajectory representation that supports general state-input constraints. On the other hand, a numerical optimal control-based planner was proposed in [32] to find a time-optimal trajectory by utilizing the complete dynamics of the quadrotor and applying exact actuator constraints. This method demonstrated exceptional performance in drone racing against human professional pilots. However, this method involves solving a nonconvex optimization problem, making it computationally challenging. The runtime of trajectory planning using this method could be up to nearly 40 minutes, which is unacceptable [32]. Despite its computational complexities, this numerical method's discretization and optimization framework can fully utilize the potential of quadrotor actuators by adequately representing dynamics and the capability of the motors [32]. This approach allows for the utilization of the full dynamics of quadrotors and enforces the exact actuator constraints, thereby ensuring that low-level controllers can easily track the open-loop trajectories. We will also design an aerial carrier landing algorithm for quadrotors based on the numerical optimization method. However, unlike the previous approach, we will transform the nonconvex optimal control problems (OCPs) into convex ones to achieve high computational efficiency. Another merit of our method is that it enables imposing constraints on both state and control profiles, ensuring the safety and reliability of the landing process.

3) Convex Optimization-Based Trajectory Planning: The optimal control method, based on convex optimization, is widely utilized in trajectory planning in the areas of aerospace and robotics [33]–[36]. This method successfully solves the original nonconvex OCP by transforming it into a convex optimization problem based on lossless convexification (LC) or sequential convex programming (SCP). LC, based on the PMP, proves that solving the new convex OCP can yield the global optimal solution for the original problem [37]–[41]. Initially applied to rocket landing problems [37]–[40], LC is typically employed for specific scenarios due to its rigorous mathematical proof, such as free-time OCPs with linear systems. Notably, [41] has established the conditions under which a fixed-time OCP can be solved by addressing a single convex optimization problem. Based on the LC theory, this article presents a trajectory planning method for quadrotor aerial landing, specifically designed for a fixed-final-time 3-DoF scenario. It only requires solving a single convex optimization problem, enabling real-time trajectory planning. The method is tested on a commonly used quadrotor onboard computer, with a solving time of approximately 10 ms.

In cases where LC is challenging to apply, such as with nonlinear control systems, SCP serves as a reliable convex optimization-based optimal control method [42]–[44]. In [42], SCP successfully addresses the path planning problem for a 3-DoF UAV under state-triggered constraints. In [43], the descent guidance problem for a 6-DoF rocket is transformed into a convex subproblem, allowing for the computation of the rocket's landing trajectory through iterative subproblem solving. This article further utilizes the actuator potential of quadrotors by developing trajectory planning algorithms that address the complete dynamics of a 6-DoF quadrotor, ensuring safe, agile, and accurate landing on AVCs. By iteratively solving convex subproblems using the proposed SCP algorithm, the solution to the original OCP can be obtained. Leveraging the good initial guesses provided by our LC-based 3-DoF trajectory, convex subproblems can be iteratively solved with high computational efficiency.

B. Contribution

We propose a safe, agile, and accurate trajectory planning algorithm for a quadrotor landing on an AVC. The key contributions can be summarized as follows:

1) We have designed a 3-DoF quadrotor landing trajectory planning method based on the LC theory. This method addresses the issue of quadrotor landing on AVCs. Additionally, we extend the sufficient conditions for the LC of unconstrained fixed-time OCPs with linear time-invariant (LTI) systems [41], to include convex state constraints. By incorporating the landing cone constraint, the safety of the landing process is ensured.
2) We have developed the SCP method to obtain the landing trajectory for the 6-DoF quadrotor dynamics, which enables us to fully exploit the potential of quadrotors. Using the high-quality 3-DoF trajectories as initial guesses, the algorithm achieves high computational efficiency as the satisfactory initial guess facilitates SCP convergence [34].
3) In contrast to previous research [16]–[19], this article expands the scope of the landing target beyond surface platforms. Our approach encompasses aerial platforms situated within a three-dimensional space, providing a reliable solution for landing in diverse scenarios.

The remaining sections are organized as follows. In Section II, we describe the quadrotor landing on AVCs as OCPs. Section III provides detailed explanations of how we convert the original problems into convex optimization problems based on the LC and SCP methods. Section IV presents the results obtained from Gazebo simulations and physical experiments. Finally, in Section V, we conclude our findings.

NOTATION

We use $\cdot$ to denote the vector dot product, $\times$ to represent the vector cross product, $\|\cdot\|$ to denote the Euclidean norm, $\mathcal{N}_p^q$ to denote the set $\{p, p+1, \ldots, q-1, q\}$, and $\mathbb{S}_+^n$ to denote the set of $n \times n$ positive definite matrices. We denote time as $t \in \mathbb{R}$. Subscripts $\mathcal{I}$ and $\mathcal{B}$ represent parameters expressed in the inertial frame $\mathcal{F}_\mathcal{I}$ and the body-fixed frame $\mathcal{F}_\mathcal{B}$, respectively. $\boldsymbol{p}_\mathcal{I}(t) \in \mathbb{R}^3$ and $\boldsymbol{v}_\mathcal{I}(t) \in \mathbb{R}^3$ denote the position and velocity of the quadrotor, respectively. The mass of the quadrotor is $m \in \mathbb{R}_+$, and the gravity acceleration is g. $\boldsymbol{s}(t) \in \mathbb{R}^6$ is the state vector for 3-DoF dynamics, $\boldsymbol{u}_T(t) \in \mathbb{R}^3$ denotes the thrust force vector in the inertial frame, i.e., the control vector for 3-DoF dynamics. $\boldsymbol{T}_\mathcal{B}$ is the collective thrust, and $R_{\mathcal{B}\mathcal{I}}(t)$

is the rotation matrix corresponding to quaternion $\boldsymbol{q}_{\mathcal{BI}}(t)$. The attitude dynamics are established based on quaternions, and $\boldsymbol{q}_{\mathcal{BI}}(t)$ denotes the unit quaternion that represents the transformation from $\mathcal{F}_{\mathcal{B}}$ to $\mathcal{F}_{\mathcal{I}}$. Similarly, the quaternion that represents the transformation from $\mathcal{F}_{\mathcal{I}}$ to $\mathcal{F}_{\mathcal{B}}$ can be denoted as $\boldsymbol{q}_{\mathcal{IB}}(t)$, and $\boldsymbol{q}_{\mathcal{IB}}(t)$ is the conjugate of $\boldsymbol{q}_{\mathcal{BI}}(t)$. $J_{\mathcal{B}} \in \mathbb{S}^3_+$ is the moment of inertia of the quadrotor, $\boldsymbol{\tau}_{\mathcal{B}}(t) \in \mathbb{R}^3$ is the torque acting on the quadrotor, and $\boldsymbol{\omega}_{\mathcal{B}}(t) \in \mathbb{R}^3$ is the angular velocity. T_i is the thrust at rotor $i \in \mathcal{N}_1^4$, l_{arm} is the quadrotor's arm length, and c_τ is the rotor's torque constant. $\boldsymbol{x}(t) \in \mathbb{R}^{13}$ and $\boldsymbol{u}(t) \in \mathbb{R}^4$ denote the state and control vectors for 6-DoF dynamics, respectively, and $f : \mathbb{R}^{13} \times \mathbb{R}^4 \to \mathbb{R}^{13}$ denotes the continuous-time nonlinear dynamics.

II. Problem Statement

This section formulates the free-final-time 3-DoF and 6-DoF quadrotor aerial landing problems as two nonconvex OCPs. Section II-A presents the 3-DoF and 6-DoF dynamics of quadrotors. Section II-B formulates the nonconvex OCP for aerial landing with the state and control constraints.

A. Dynamics

The 3-DoF dynamics of the quadrotor can be described by

$$\dot{\boldsymbol{p}}_{\mathcal{I}}(t) = \boldsymbol{v}_{\mathcal{I}}(t), \tag{1}$$

$$\dot{\boldsymbol{v}}_{\mathcal{I}}(t) = \frac{1}{m}\boldsymbol{u}_T(t) - \boldsymbol{g}_{\mathcal{I}}, \tag{2}$$

where $\boldsymbol{g}_{\mathcal{I}} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^{\mathrm{T}}$. The 3-DoF dynamics can be described as an LTI system by

$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega}, \tag{3}$$

where

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, B = \begin{bmatrix} 0_{3\times3} \\ \frac{1}{m}I_{3\times3} \end{bmatrix}, \tag{4}$$

$$\boldsymbol{s}(t) \triangleq \begin{bmatrix} \boldsymbol{p}_{\mathcal{I}}^{\mathrm{T}} & \boldsymbol{v}_{\mathcal{I}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{5}$$

$$\boldsymbol{\omega} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -g \end{bmatrix}^{\mathrm{T}}. \tag{6}$$

To fully exploit the quadrotor's actuator potential, the 6-DoF quadrotor dynamics are also utilized. The translational dynamics of the quadrotor are given as

$$\dot{\boldsymbol{p}}_{\mathcal{I}}(t) = \boldsymbol{v}_{\mathcal{I}}(t), \tag{7}$$

$$\dot{\boldsymbol{v}}_{\mathcal{I}}(t) = \frac{1}{m}R_{\mathcal{BI}}(t)\boldsymbol{T}_{\mathcal{B}}(t) + \boldsymbol{g}_{\mathcal{I}}. \tag{8}$$

The attitude dynamics are given by

$$\dot{\boldsymbol{q}}_{\mathcal{IB}}(t) = \frac{1}{2}\Omega\left(\boldsymbol{\omega}_{\mathcal{B}}(t)\right)\boldsymbol{q}_{\mathcal{IB}}(t), \tag{9}$$

$$J_{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{B}}(t) = \boldsymbol{\tau}_{\mathcal{B}}(t) - \boldsymbol{\omega}_{\mathcal{B}}(t) \times J_{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}}(t), \tag{10}$$

where $\Omega(\cdot)$ is a skew-symmetric matrix defined for the quaternion kinematics [45].

The collective thrust input $\boldsymbol{T}_{\mathcal{B}}(t)$ can be decomposed into the single rotor thrusts. The total thrust and the torque can be presented as

$$\boldsymbol{T}_{\mathcal{B}}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\boldsymbol{u}(t), \tag{11}$$

$$\boldsymbol{\tau}_{\mathcal{B}}(t) = \begin{bmatrix} l_\tau & l_\tau & -l_\tau & -l_\tau \\ -l_\tau & l_\tau & l_\tau & -l_\tau \\ c_\tau & -c_\tau & c_\tau & -c_\tau \end{bmatrix}\boldsymbol{u}(t), \tag{12}$$

where $l_\tau = l_{arm}/\sqrt{2}$. The 6-DoF dynamics (7)-(12) can also be represented as

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall t \in [t_0, t_f], \tag{13}$$

where

$$\boldsymbol{x}(t) \triangleq \begin{bmatrix} \boldsymbol{p}_{\mathcal{I}}^{\mathrm{T}}(t) & \boldsymbol{v}_{\mathcal{I}}^{\mathrm{T}}(t) & \boldsymbol{q}_{\mathcal{IB}}^{\mathrm{T}}(t) & \boldsymbol{\omega}_{\mathcal{B}}^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}, \tag{14}$$

$$\boldsymbol{u}(t) \triangleq \begin{bmatrix} T_1(t) & T_2(t) & T_3(t) & T_4(t) \end{bmatrix}^{\mathrm{T}}. \tag{15}$$

B. Aerial Landing Problem

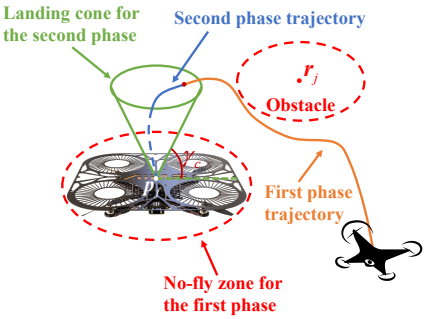


Fig. 2. Aerial landing scenario.

To safely land on the AVC, the quadrotor should perform a collision-avoidance flight to reach the airspace above the AVC, and then accurately land on the AVC; see Fig. 2. To prevent collisions with both the AVC and obstacles, a set of no-fly zones (NFZs) is implemented. If there are a total of k_{obs} NFZs within the flight space, each NFZ is represented as an ellipsoid centered at $\boldsymbol{r}_j \in \mathbb{R}^3$, denoting the center of the j-th NFZ. The shape of the j-th ellipsoidal NFZ is specified by the matrix $H_j \in \mathbb{S}^3_+$. Through the incorporation of the NFZs, the collision-avoidance constraint can be defined as follows:

$$\|H_j(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{r}_j)\| \geq 1, \quad \forall j \in \mathcal{N}_1^{k_{obs}}. \tag{16}$$

To further ensure safety in the landing process, the quadrotor's trajectory is constrained in a landing cone as shown in Fig. 2. The landing cone constraint can be defined by

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f)\| \leq \cot(\gamma_c)\boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f), \tag{17}$$

$$\boldsymbol{e} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, H_c \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{18}$$

where $\boldsymbol{p}_f$ is the final landing point, $\gamma_c \in [0°, 90°)$ is the angle between the cone and the horizontal, and $\cot(\cdot)$ is the cotangent function.

For the 6-DoF dynamics, the thrust magnitude of each rotor is limited in an applicable interval, specified by:

$$T_{min} \leq T_i(t) \leq T_{max} \quad \forall i \in \mathcal{N}_1^4. \tag{19}$$

The nonconvex 6-DoF aerial landing trajectory planning problem is summarized as follows.

$$\min_{\boldsymbol{u}(t), t_l, t_f} \int_{t_0}^{t_f} \|\boldsymbol{u}(t)\| \, dt, \tag{20a}$$

subject to:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \quad \forall t \in [t_0, t_f], \tag{20b}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \boldsymbol{x}(t_f) = \boldsymbol{x}_f, \tag{20c}$$

$$T_{min} \leq T_i(t) \leq T_{max} \quad \forall i \in \mathcal{N}_1^4 \quad \forall t \in [t_0, t_f], \tag{20d}$$

$$\|H_j(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{r}_j)\| \geq 1 \quad \forall j \in \mathcal{N}_1^{k_{obs}} \quad \forall t \in [t_0, t_l], \tag{20e}$$

$$\|\boldsymbol{p}_{\mathcal{I}}(t_l) - (\boldsymbol{p}_f + h_s \boldsymbol{e})\| \leq h_s \cos\gamma_c, \tag{20f}$$

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f)\| \leq \cot(\gamma_c) \boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f) \quad \forall t \in [t_l, t_f], \tag{20g}$$

where t_0 is the initial time, $\boldsymbol{x}_0$ is the initial state, t_f is the terminal time, $\boldsymbol{x}_f$ is the terminal state, h_s is a safe height above the AVC as shown in Fig. 3, and t_l is the moment when the quadrotor reaches the airspace above the AVC and begins to land on the AVC, i.e., t_l is the final time of the first phase. The constraint (20f) ensures $\boldsymbol{p}_{\mathcal{I}}(t_l)$ is within the landing cone and above the AVC.

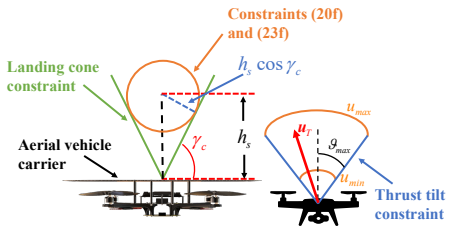


Fig. 3. Safe height, thrust tilt, and thrust magnitude constraints.

Problem (20) is nonconvex because the constraints (20b) and (20e) are nonconvex. The key focus of this article is to tackle the OCP (20). This nonconvex OCP will be addressed using the SCP algorithm given in Section III.

III. Methodology

This section introduces the LC method and the SCP algorithm to solve 3-DoF and 6-DoF aerial landing problems.

A. Lossless Convexification

While our primary objective is to address the nonconvex OCP (20), we also aim to expedite the convergence of the SCP algorithm by obtaining a reliable initial guess. To accomplish this, we tackle the 3-DoF quadrotor landing problem using LC theory. The high-quality initial guess based on the 3-DoF trajectory ensures the high computational efficiency of SCP.

For the 3-DoF dynamics, the thrust generated by the motors has both maximum and minimum limits. As shown in Fig. 3, the total thrust magnitude is limited as

$$u_{min} \leq \|\boldsymbol{u}_T(t)\| \leq u_{max}, \tag{21}$$

where u_{min} and u_{max} are the minimum and maximum total thrust, respectively. Moreover, the thrust tilt angle is constrained to a maximum angle $\vartheta_{max} \in (0°, 90°)$ off-vertical as shown in Fig. 3. The thrust tilt constraint is defined by

$$\|H_c \boldsymbol{u}_T(t)\| \leq \tan(\vartheta_{max}) \boldsymbol{e} \boldsymbol{u}_T(t). \tag{22}$$

The objective function and boundary conditions of the OCP can be designed according to the mission requirements. In this paper, we focus on minimizing the control effort to save energy. The formulation of the 3-DoF aerial landing trajectory planning problem can be summarized as follows.

$$\min_{\boldsymbol{u}_T(t)} \int_{t_0}^{t_f} \|\boldsymbol{u}_T(t)\| \, dt, \tag{23a}$$

subject to:

$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega} \quad \forall t \in [t_0, t_f], \tag{23b}$$

$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{23c}$$

$$u_{min} \leq \|\boldsymbol{u}_T(t)\| \leq u_{max} \quad \forall t \in [t_0, t_f], \tag{23d}$$

$$\|H_c \boldsymbol{u}_T(t)\| \leq \tan(\vartheta_{max}) \boldsymbol{e} \boldsymbol{u}_T(t) \quad \forall t \in [t_0, t_f], \tag{23e}$$

$$\|\boldsymbol{p}_{\mathcal{I}}(t_l) - (\boldsymbol{p}_f + h_s \boldsymbol{e})\| \leq h_s \cos\gamma_c, \tag{23f}$$

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f)\| \leq \cot(\gamma_c) \boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{p}_f) \quad \forall t \in [t_l, t_f], \tag{23g}$$

where $\boldsymbol{s}_0$ is the initial state, and $\boldsymbol{s}_f$ is the terminal state.

The OCP (23) is nonconvex because the lower bound in the input constraint (23d) is nonconvex. This nonconvexity can be inferred by noting that the upper bound in (23d) is convex, while the lower bound is nonconvex, as it opposes the upper bound. Meanwhile, the constraints (23e-23g) can be confirmed to be convex by noting that the left side of the inequalities are Euclidean norms containing an affine mapping of optimization variables, while on the right side are constants or affine to the optimization variables. The nonconvex OCP will be transformed into a convex problem using LC in this section. The collision-avoidance constraint (16) is not involved in Problem (23) because the nonconvexity in (16) cannot be seamlessly handled by the LC method. The SCP algorithm proposed in Section III-B will address collision avoidance to ensure safety. The solution to (23) can be utilized as the initial guess for the SCP algorithm in Section III-B to efficiently solve the 6-DoF aerial landing problem.

To begin, we reformulate the OCP statement (23) as a problem without the convex state constraints (23e)-(23g):

$$\min_{\boldsymbol{u}_T(t)} \int_{t_0}^{t_f} \|\boldsymbol{u}_T(t)\| \, dt, \tag{24a}$$

subject to:

$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega} \quad \forall t \in [t_0, t_f], \tag{24b}$$

$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{24c}$$

$$u_{min} \leq \|\boldsymbol{u}_T(t)\| \leq u_{max}, \tag{24d}$$

The nonconvexity of (24) lies in the lower bound on the input defined by (24d). Based on the LC theory [41], we relax the

nonconvex OCP (24) to a convex formulation as

$$\min_{\boldsymbol{u}_T(t),\sigma} \int_{t_0}^{t_f} \sigma(t)dt, \tag{25a}$$
subject to:
$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega} \quad \forall t \in [t_0, t_f], \tag{25b}$$
$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{25c}$$
$$u_{min} \leq \sigma(t) \leq u_{max}, \tag{25d}$$
$$\|\boldsymbol{u}_T(t)\| \leq \sigma(t), \tag{25e}$$

where $\sigma(t) \in \mathbb{R}$ is the slack input. The control set defined by (25d) and (25e) is a convex relaxation of the original control set (24d). Thus, solutions to (25) may be infeasible for (24). If $\|\boldsymbol{u}_T(t)\| = \sigma(t)$ holds almost everywhere, a solution to (25) would also be feasible w.r.t. (24). To achieve this, the sufficient conditions for LC are stated below.

Lemma 1 (Lossless Convexification): The optimal solution to (25) is optimal for (24) if 1) (A, B) is controllable and 2) t_f is between the minimal feasible terminal time and the terminal time that minimizes (24a). If t_f is larger than the time that minimizes (24a), there exists a feasible solution to (25) that is globally optimal for (24).

Proof: The proof follows from [41] (see Theorem 6, Corollary 7, Theorem 8, Theorem 9, and Theorem 13 of [41], compare also with Theorem 7 of [35]). The only difference here is that the system (25b) has a constant term $\boldsymbol{\omega}$. Since $\boldsymbol{\omega}$ is defined by (6), the system can be equivalently written as

$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B(\boldsymbol{u}_T(t) + \boldsymbol{\omega}_g), \tag{26}$$

where $\boldsymbol{\omega}_g = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^{\mathrm{T}}$. $(\boldsymbol{u}_T(t) + \boldsymbol{\omega}_g)$ can be treated as the new control input. And this does not change the controllability, since A and B remain the same. Then, the proof can be directly adapted from [41]. ■

Using Lemma 1, the nonconvexity in (24) can be eliminated. However, this result does not take the convex state constraints into account. The original problem (23) with state constraints can be rewritten as

$$\min_{\boldsymbol{u}_T(t)} \int_{t_0}^{t_f} \|\boldsymbol{u}_T(t)\| \, dt, \tag{27a}$$
subject to:
$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega} \quad \forall t \in [t_0, t_f], \tag{27b}$$
$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{27c}$$
$$u_{min} \leq \|\boldsymbol{u}_T(t)\| \leq u_{max}, \tag{27d}$$
$$\boldsymbol{s}(t) \in \mathcal{S}, \tag{27e}$$

where $\mathcal{S}$ is the convex set defined by the convex state constraints. Similarly, we can relax this problem to a convex formulation as

$$\min_{\boldsymbol{u}_T(t),\sigma(t)} \int_{t_0}^{t_f} \sigma(t)dt, \tag{28a}$$
subject to:
$$\dot{\boldsymbol{s}}(t) = A\boldsymbol{s}(t) + B\boldsymbol{u}_T(t) + \boldsymbol{\omega} \quad \forall t \in [t_0, t_f], \tag{28b}$$
$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{28c}$$
$$u_{min} \leq \sigma(t) \leq u_{max}, \tag{28d}$$
$$\|\boldsymbol{u}_T(t)\| \leq \sigma(t), \tag{28e}$$
$$\boldsymbol{s}(t) \in \mathcal{S}. \tag{28f}$$

We use $\mathrm{int}\mathcal{S}$ to represent the interior of the convex set $\mathcal{S}$. The theorem about LC of (27) is given as follows.

Theorem 1: Lemma 1 can be applied to OCPs (27) and (28) if the convex state constraints (27e) and (28f) are activated at isolated time instances, i.e., $\boldsymbol{s}(t) \in \mathrm{int}\mathcal{S}$ almost everywhere.

Proof: Whenever $\boldsymbol{s}(t) \in \mathrm{int}\mathcal{S}$ for any time interval $t \in (t_1, t_2)$ with consecutive junction times $t_1 \leq t_2$, the state of the OCP is unconstrained along (t_1, t_2) (see Corollary 3 of [37] and its proof). Moreover, since the system dynamics (27b) and the boundary constraints (27c) are both time-invariant, i.e., the OCP is autonomous, every portion of the optimal trajectory is itself optimal (see Lemma 2 of [39] and its proof). Consequently, whenever $\boldsymbol{s}(t) \in \mathrm{int}\mathcal{S}$, the solution to (28) is equivalent to that to (25), then Lemma 1 applies (compare also with Theorem 5 of [35]). The proof is completed. ■

Remark 1: In previous studies [37]–[40], it was identified that the LC method cannot be employed for problems where both the final state and final time are fixed [35]. However, a subsequent study [41] extended the LC approach to encompass problems without state constraints where the final state and final time are both fixed. Building upon this, Theorem 1 further broadened the findings of [41] to encompass fixed-final-state and fixed-final-time problems with convex state constraints. This advancement allows for the application of the developed methodology to address trajectory planning issues, specifically in the context of quadrotor landing on AVCs.

Remark 2: Theorem 1 requires the convex state constraints to be activated only at isolated time instances. This condition cannot be checked a priori, and it can be difficult to satisfy when the state constraints are tight. Thus, we only use LC to provide a good initial guess for the SCP algorithm presented in Section III-B. The SCP algorithm can benefit from a good initial guess in terms of convergence rate and optimality [34]. Besides, the SCP algorithm uses the complete dynamics of the quadrotor and thus guarantees the exact constraints specified on the quadrotor are satisfied even if Theorem 1 is invalid.

B. SCP Algorithm

This subsection presents the SCP algorithm to solve (20). The solution generated by this algorithm ensures the dynamical feasibility of the original problem through successive linearization. The proposed algorithm approximates the original problem by iteratively solving a sequence of convex subproblems. Although several iterations are required to obtain a converged solution, this algorithm is still computationally

efficient due to the low computational burden of convex programming (see e.g., Section IV).

The original problem (20) is a free-time nonconvex OCP. We can define a scaled time $\tau \in [0,1]$ to reformulate the original problem into an equivalent problem with a fixed final scaled time. According to the chain rule, the quadrotor's dynamics can be represented as

$$\boldsymbol{x}'(\tau) \triangleq \frac{d}{d\tau}\boldsymbol{x}(\tau) = \frac{dt}{d\tau} f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)). \tag{29}$$

The initial time t_0 is assumed to be 0, such that $t = t_f\tau$. Thus, the transformed dynamics is

$$\boldsymbol{x}'(\tau) = t_f f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \triangleq F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), t_f). \tag{30}$$

The free-final-time problem is now converted into a fixed-final-scaled-time problem on the normalized temporal interval $\tau \in [0,1]$. The original final time t_f becomes a parameter in the system (30).

To utilize the high computational efficiency of convex programming, the original problem (20) should be converted into a convex programming problem. Thus, the dynamics constraint, which is an equality constraint, must be affine [46]. The dynamics (30) is linearized as

$$\boldsymbol{x}'(\tau) \approx A(\tau)\boldsymbol{x}(\tau) + B(\tau)\boldsymbol{u}(\tau) + \boldsymbol{s}(\tau)t_f + \boldsymbol{c}(\tau), \tag{31}$$

$$A(\tau) \triangleq \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{\tilde{z}(\tau)}, B(\tau) \triangleq \left.\frac{\partial F}{\partial \boldsymbol{u}}\right|_{\tilde{z}(\tau)}, \tag{32}$$

$$\boldsymbol{s}(\tau) \triangleq \left.\frac{\partial F}{\partial t_f}\right|_{\tilde{z}(\tau)}, \boldsymbol{c}(\tau) \triangleq -A(\tau)\tilde{\boldsymbol{x}}(\tau) - B(\tau)\tilde{\boldsymbol{u}}(\tau), \tag{33}$$

where $\tilde{\boldsymbol{z}}(\tau) \triangleq \begin{bmatrix} \tilde{t}_f & \tilde{\boldsymbol{x}}^{\mathrm{T}}(\tau) & \tilde{\boldsymbol{u}}^{\mathrm{T}}(\tau) \end{bmatrix}^{\mathrm{T}}$ is the reference trajectory. $\tilde{t}_f$, $\tilde{\boldsymbol{x}}$, and $\tilde{\boldsymbol{u}}$ are the reference values of t_f, $\boldsymbol{x}$, and $\boldsymbol{u}$ obtained from the last iteration, respectively.

The original continuous-time problem should be discretized. By introducing N evenly spaced nodes, the problem is discretized into $N-1$ subintervals. In each subinterval, the control input is approximated by the first-order-hold interpolation as

$$\boldsymbol{u}(\tau) = \hat{\eta}_k(\tau)\boldsymbol{u}_k + \eta_k(\tau)\boldsymbol{u}_{k+1} \quad \forall \tau \in [\tau_k, \tau_{k+1}], \tag{34}$$

$$\hat{\eta}_k(\tau) = \frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}, \quad \eta_k(\tau) = \frac{\tau - \tau_k}{\tau_{k+1} - \tau_k}, \tag{35}$$

where $k \in \mathcal{N}_1^{N-1}$, $\tau_k = (k-1)/(N-1)$, and $\boldsymbol{u}_k \triangleq \boldsymbol{u}(\tau_k)$.

The linearized dynamics (31) is a linear-time-varying (LTV) system. By applying the properties of the state transition matrix [47], we can obtain the discrete-time LTV dynamics for each $k \in \mathcal{N}_1^{N-1}$:

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + \hat{B}_k\boldsymbol{u}_k + B_k\boldsymbol{u}_{k+1} + \boldsymbol{s}_k t_f + \boldsymbol{c}_k, \tag{36}$$

$$\Phi(\tau, \tau_k) \triangleq I + \int_{\tau_k}^{\tau} A(\zeta)\Phi(\zeta, \tau_k)\,\mathrm{d}\zeta, \tag{37a}$$

$$A_k \triangleq \Phi(\tau_{k+1}, \tau_k), \tag{37b}$$

$$\hat{B}_k \triangleq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_i)\,\hat{\eta}(\tau)B(\tau)\mathrm{d}\tau, \tag{37c}$$

$$B_k \triangleq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\,\eta(\tau)B(\tau)\mathrm{d}\tau, \tag{37d}$$

$$\boldsymbol{s}_k \triangleq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_k)\,\boldsymbol{s}(\tau)\mathrm{d}\tau, \tag{37e}$$

$$\boldsymbol{c}_k \triangleq A_k \int_{\tau_k}^{\tau_{k+1}} \Phi^{-1}(\tau, \tau_i)\,\boldsymbol{c}(\tau)\mathrm{d}\tau, \tag{37f}$$

where I is an identity matrix with the appropriate dimension, and $\boldsymbol{x}_k \triangleq \boldsymbol{x}(\tau_k)$. Note that the integration in (37) only uses the reference trajectories $\tilde{\boldsymbol{z}}$. Thus, in implementation, the computation of (37) can be done simultaneously, which reduces the computational burden.

The SCP algorithm needs to iteratively solve the convex subproblems, due to the linearization. Hence, we introduce the trust-region constraint to ensure the validity of the linearization within the trajectory's feasible region. To enable the convex programming to choose the trust region, we augment the objective function with

$$J_{\mathrm{tr}} \triangleq w_{tr} \sum_{k \in \mathcal{N}_1^{N+1}} \zeta_k, \tag{38}$$

$$\|\delta\boldsymbol{x}_k\| \le \zeta_k \quad \forall k \in \mathcal{N}_1^{N}, \tag{39}$$

$$\|\delta t_f\| \le \zeta_{N+1}, \tag{40}$$

where $\delta\boldsymbol{x}_k \triangleq \boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k$, $\delta t_f \triangleq t_f - \tilde{t}_f$, and $w_{tr} > 0$ is the weighting term.

The linearization can also cause artificial infeasibility [43]. The linearized problem may be infeasible, even when the original problem is feasible. For instance, this can occur if the linearized constraints conflict with others because the linearization relies on an unrealistic reference trajectory. We can introduce a virtual control term $\boldsymbol{\nu}_k \in \mathbb{R}^{13}$ into the linearized dynamics (36) to solve this issue. Then, the dynamics can be given as

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + \hat{B}_k\boldsymbol{u}_k + B_k\boldsymbol{u}_{k+1} + \boldsymbol{s}_k t_f + \boldsymbol{c}_k + \boldsymbol{\nu}_k. \tag{41}$$

The virtual control term should be used only as necessary. Hence, the objective function is augmented with

$$J_{\mathrm{vc}} \triangleq w_{vc} \sum_{k \in \mathcal{N}_1^{N-1}} \xi_k, \tag{42}$$

$$\|\boldsymbol{\nu}_k\| \le \xi_k \quad k \in \mathcal{N}_1^{N-1}, \tag{43}$$

where $w_{vc} \in \mathbb{R}_{++}$ is a large weighting term to enable the virtual control only when necessary.

The original problem (20) has the constraint (20e) for collision avoidance, which is also nonconvex. The constraint (20e) can be rewritten as follows.

$$1 - \|H_j(\boldsymbol{p}_{\mathcal{I}}(t) - \boldsymbol{r}_j)\| \le 0 \quad \forall j \in \mathcal{N}_1^{k_{obs}}. \tag{44}$$

The above constraint can be rewritten in a more general way.

$$h(\boldsymbol{y}) \leq 0, \tag{45}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the optimization variable, and $h : \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable concave function. To begin, we give the linearized form of (45) as

$$h(\tilde{\boldsymbol{y}}) + \nabla h(\tilde{\boldsymbol{y}})(\boldsymbol{y} - \tilde{\boldsymbol{y}}) \leq 0, \tag{46}$$

which is linearized with respect to the reference value $\tilde{\boldsymbol{y}}$. The following theorem can be presented to show that once the linearized constraint (46) is satisfied, the original constraint (45) can also be satisfied.

Theorem 2: If a feasible solution satisfies the linearized constraint (46), then the solution also satisfies the original constraint (45).

Proof: To begin, we use the second-order Taylor-series expansion to represent the original concave function $h(\boldsymbol{y})$ as

$$h(\boldsymbol{y}) \approx h(\tilde{\boldsymbol{y}}) + \nabla h(\tilde{\boldsymbol{y}})(\boldsymbol{y} - \tilde{\boldsymbol{y}}) + \frac{1}{2}(\boldsymbol{y} - \tilde{\boldsymbol{y}})^{\mathrm{T}} \nabla^2 h(\tilde{\boldsymbol{y}})(\boldsymbol{y} - \tilde{\boldsymbol{y}}). \tag{47}$$

Since $h(\boldsymbol{y})$ is concave, the Hessian $\nabla^2 h(\tilde{\boldsymbol{y}})$ is negative semidefinite:

$$\nabla^2 h(\tilde{\boldsymbol{y}}) \preceq 0. \tag{48}$$

Then, we have

$$\frac{1}{2}(\boldsymbol{y} - \tilde{\boldsymbol{y}})^{\mathrm{T}} \nabla^2 h(\tilde{\boldsymbol{y}})(\boldsymbol{y} - \tilde{\boldsymbol{y}}) \leq 0. \tag{49}$$

By substituting (46) and (49) into (47), we have

$$h(\boldsymbol{y}) \leq 0 \tag{50}$$

which is (45). The proof is completed. ■

Remark 3: Theorem 2 shows that for a concave constraint like (45), where the function $h(\cdot)$ is concave and twice differentiable, the linearized form (46) imposes a more stringent constraint. Theorem 2 confirms that the constraint (20e) can be linearized without sacrificing safety. Consequently, the original OCP (20) can be converted into a convex formulation via linearization. We will summarize the convex formulation together with the proposed algorithm in the next subsection.

C. Application to Aerial Landing of Quadrotors

In this subsection, the proposed algorithm is summarized based on Section III-A and Section III-B. According to the analysis in Section III-A, we can solve Problem (28) to obtain the 3-DoF trajectory. By introducing N evenly spaced nodes, Problem (28) can be discretized as

Problem 1:

$$\min_{\boldsymbol{u}_T(t_i), \sigma(t_i)} \sum_{k \in \mathcal{N}_1^N} \sigma(t_i), \tag{51a}$$

subject to:

$$\begin{aligned} \boldsymbol{p}_{\mathcal{I}}(t_{i+1}) &= \boldsymbol{p}_{\mathcal{I}}(t_i) + \boldsymbol{v}_{\mathcal{I}}(t_i)\Delta t \\ &\quad + \frac{\Delta t^2}{6m}(2\boldsymbol{u}_T(t_i) + \boldsymbol{u}_T(t_{i+1}) - 3m\boldsymbol{g}_{\mathcal{I}}) \\ \boldsymbol{v}_{\mathcal{I}}(t_{i+1}) &= \boldsymbol{v}_{\mathcal{I}}(t_i) \\ &\quad + \frac{\Delta t}{2m}(\boldsymbol{u}_T(t_i) + \boldsymbol{u}_T(t_{i+1}) - 2m\boldsymbol{g}_{\mathcal{I}}) \quad \forall i \in \mathcal{N}_1^{N-1}, \end{aligned} \tag{51b}$$

$$\boldsymbol{s}(t_0) = \boldsymbol{s}_0, \boldsymbol{s}(t_f) = \boldsymbol{s}_f, \tag{51c}$$

$$u_{min} \leq \sigma(t_i) \leq u_{max} \quad \forall i \in \mathcal{N}_1^N, \tag{51d}$$

$$\|\boldsymbol{u}_T(t_i)\| \leq \sigma(t_i) \quad \forall i \in \mathcal{N}_1^N, \tag{51e}$$

$$\|H_c \boldsymbol{u}_T(t_i)\| \leq \tan(\vartheta_{max}) \boldsymbol{e} \boldsymbol{u}_T(t_i) \quad \forall i \in \mathcal{N}_1^N, \tag{51f}$$

$$\|\boldsymbol{p}_{\mathcal{I}}(t_l) - (\boldsymbol{p}_f + h_s \boldsymbol{e})\| \leq h_s \cos\gamma_c, \tag{51g}$$

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(t_i) - \boldsymbol{p}_f)\| \leq \cot(\gamma_c) \boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(t_i) - \boldsymbol{p}_f) \quad \forall i \in \mathcal{N}_l^N, \tag{51h}$$

where $t_i = (t_f - t_0)(i-1)/(N-1)$ is the i-th time discrete node, $\Delta t = (t_f - t_0)/(N-1)$ denotes the time interval between two discrete nodes, and the l-th time node t_l is when the quadrotor should reach the airspace above the AVC and begin the landing process. The dynamics (51b) are obtained by performing a first-order-hold on the control input. This is the same as the first-order-hold control (34) in the SCP method. This approach can help to enhance the terminal accuracy (see [48] and [49]).

In our algorithm, Problem 1 is solved to provide a good initial guess for the SCP method. The collision avoidance is only considered in the SCP algorithm since the linearization (46) requires a reference trajectory. By solving Problem 1, we can directly obtain the position and velocity along the trajectory of the quadrotor. The SCP algorithm requires a reference trajectory with full dynamics (30). Although a good initial guess aids the convergence of the SCP algorithm, it isn't necessarily required to be dynamically feasible (see also [34]). Therefore, we simply assume the angular velocity and the heading angle are all zero. The attitude of the initial guess can be computed by utilizing the fact that the thrust direction of a quadrotor should align with the z axis of the body frame [20]. The initial guess of the thrust can be computed by

$$T_i(t_k) = \frac{\|\boldsymbol{u}_T(t_k)\|}{4} \quad \forall i \in \mathcal{N}_1^4 \quad \forall k \in \mathcal{N}_1^N. \tag{52}$$

With the initial guess, the convex subproblem of the SCP algorithm can be constructed. To simplify, the initial time t_0 is assumed to be 0, then the convex subproblem is given as Problem 2 for all $k \in \mathcal{N}_1^N$, $\bar{k} \in \mathcal{N}_1^{N-1}$, and $\hat{k} \in \mathcal{N}_1^{N+2}$.

Problem 2:

$$\min_{t_f, t_l, \boldsymbol{u}_k, \boldsymbol{x}_k, \boldsymbol{\nu}_{\bar{k}}, \sigma_k, \xi_{\hat{k}}, \zeta_{\hat{k}}} \sum \sigma_k + w_{tr} \sum \zeta_{\bar{k}} + w_{vc} \sum \xi_{\hat{k}}, \tag{53a}$$

subject to:

$$\begin{aligned} \boldsymbol{x}_{i+1} = A_i \boldsymbol{x}_i + \hat{B}_i \boldsymbol{u}_i + B_i \boldsymbol{u}_{i+1} + \boldsymbol{s}_i t_l + \boldsymbol{c}_i + \boldsymbol{\nu}_i, \\ \forall i \in \mathcal{N}_1^{l-1}, \end{aligned} \tag{53b}$$

$$\begin{aligned} \boldsymbol{x}_{i+1} = A_i \boldsymbol{x}_i + \hat{B}_i \boldsymbol{u}_i + B_i \boldsymbol{u}_{i+1} + \boldsymbol{s}_i (t_f - t_l) + \boldsymbol{c}_i + \boldsymbol{\nu}_i, \\ \forall i \in \mathcal{N}_l^{N-1}, \end{aligned} \tag{53c}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_i, \ \boldsymbol{x}_N = \boldsymbol{x}_f, \tag{53d}$$

$$T_{\min} \leq T_i(\tau_k) \leq T_{\max} \quad \forall i \in \mathcal{N}_1^4, \tag{53e}$$

$$-\boldsymbol{\omega}_{\max} \leq \boldsymbol{\omega}_{\mathcal{B}}(\tau_k) \leq \boldsymbol{\omega}_{\max}, \tag{53f}$$

$$g_j(\tilde{\boldsymbol{x}}_i) + \nabla g_j(\tilde{\boldsymbol{x}}_i)(\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i) \leq 0 \quad \forall j \in \mathcal{N}_1^{k_{obs}} \quad \forall i \in \mathcal{N}_1^l, \tag{53g}$$

$$\|\boldsymbol{p}_{\mathcal{I}}(\tau_l) - (\boldsymbol{p}_f + h_s\boldsymbol{e})\| \leq h_s \cos\gamma_c, \tag{53h}$$

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(\tau_i) - \boldsymbol{p}_f)\| \leq \cot(\gamma_c)\boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(\tau_i) - \boldsymbol{p}_f) \quad \forall i \in \mathcal{N}_l^N, \tag{53i}$$

$$\|\boldsymbol{u}_k\| \leq \sigma_k, \tag{53j}$$

$$\|\boldsymbol{\nu}_{\bar{k}}\| \leq \xi_{\bar{k}}, \tag{53k}$$

$$\|\delta\boldsymbol{x}_k\| \leq \zeta_k, \tag{53l}$$

$$\|\delta t_l\| \leq \zeta_{N+1}, \|\delta t_f\| \leq \zeta_{N+2}, \tag{53m}$$

where $\boldsymbol{x}_i$ is the initial state, $g_j(\boldsymbol{x}_k) = 1 - \|H_j(\boldsymbol{p}_{\mathcal{I}}(\tau_k) - \boldsymbol{r}_j)\|$. The body rate constraint (53f) is introduced to guarantee the trajectory is bounded, such that the optimal trajectory exists (see Lemma 3.1 of [34]). Note that the problem is formulated as a two-phase problem, and each phase is a free-time OCP since t_l and t_f are both optimization variables. The first phase is designed to enable the quadrotor to reach the airspace above the AVC without collision $\forall t \in [t_0, t_l]$. The second phase is designed to enable the quadrotor to safely and precisely land on the AVC $\forall t \in [t_l, t_f]$ as shown in Fig. 2.

Since the SCP algorithm is an iterative approach, the convergence criteria should be given. The algorithm terminates when the following convergence criteria are met:

$$\sum_{k \in \mathcal{N}_1^{N+2}} \zeta_k \leq \epsilon_{\mathrm{tr}}, \tag{54}$$

$$\sum_{k \in \mathcal{N}_1^{N-1}} \|\boldsymbol{\nu}_k\|_1 \leq \epsilon_{\mathrm{vc}}, \tag{55}$$

where $\epsilon_{\mathrm{tr}} \in \mathbb{R}_{++}$ and $\epsilon_{\mathrm{vc}} \in \mathbb{R}_{++}$ are the user-defined convergence tolerances. The convergence criterion (55) ensures that the solution is dynamically feasible, thus, (55) must be satisfied. The convergence criterion (54) ensures that the solution is converged. As long as (55) is met, a solution that does not meet (54) is still dynamically feasible but sub-optimal. Hence, if the algorithm reaches maximum iterations before both criteria are met, the solution is safe to use under the satisfaction of (55).

The proposed algorithm is summarized as Algorithm 1, and i_{max} is the user-defined maximum iteration.

Algorithm 1 Aerial Landing Algorithm.

LOSSLESS CONVEXIFICATION
  solve Problem 1 to compute the initial trajectory
  return $(\tilde{t_f}, \tilde{t_l}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}})$
SEQUENTIAL CONVEX PROGRAMMING
  $i \leftarrow 0$
  while *not converged* and $i \leq i_{max}$ **do**
    use $(\tilde{t_f}, \tilde{t_l}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}})$ to construct Problem 2
    solve Problem 2 to get $(t_f, t_l, \boldsymbol{x}, \boldsymbol{u})$
    if criteria (54) and (55) are satisfied *converged*
    end if
    $(\tilde{t_f}, \tilde{t_l}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}) \leftarrow (t_f, t_l, \boldsymbol{x}, \boldsymbol{u})$
    $i \leftarrow i + 1$
  end while
  return $(t_f, t_l, \boldsymbol{x}, \boldsymbol{u})$

D. Time-varying Landing Cone for Moving AVCs

To ensure a safe landing on a moving AVC, it is essential to design the time-varying landing cone based on constraints (53h) and (53i) as

$$\|\boldsymbol{p}_{\mathcal{I}}(\tau_l) - (\boldsymbol{p}_a(\tau_l) + h_s\boldsymbol{e})\| \leq h_s \cos\gamma_c, \tag{56}$$

$$\|H_c(\boldsymbol{p}_{\mathcal{I}}(\tau_i) - \boldsymbol{p}_a(\tau_i))\| \leq \cot(\gamma_c)\boldsymbol{e}(\boldsymbol{p}_{\mathcal{I}}(\tau_i) - \boldsymbol{p}_a(\tau_i)), \tag{57}$$

where $\boldsymbol{p}_a(\tau_i)$ represents the AVC's position at τ_i, $\forall i \in \mathcal{N}_l^N$.

Apart from guaranteeing safety, the time-varying landing cone constraint holds the advantage of aligning the quadrotor's landing process with the motion trend of the AVC. By maintaining this synchronization, the experiment's success is further assured. For more details regarding these experiments and the corresponding discussion, please refer to Section IV-B.

IV. RESULTS

This section demonstrates the simulation and real-world experiments to show the capabilities of our method. A supplementary video can be found at: https://www.youtube.com/watch?v=yUllqTXWafs or https://www.bilibili.com/video/BV1Mz4y1V7T6/.

A. Simulation Experiments

We present and analyze the simulation results of two static aerial landing scenarios in the Gazebo simulation. The first scenario demonstrates the normal landing process, while the second scenario involves the quadrotor starting below the AVC and requiring a collision-avoidance flight to reach above the AVC before landing within the landing cone. We used the Intel NUC 12 with Core i5-1250P, which can serve as an onboard computer for quadrotors [50], to obtain the simulation results. The CVXPY and MOSEK solvers are used to solve convex optimization problems [51]. The quadrotor position tracking control method used in the simulation follows the nominal NMPC method in [52].

TABLE I
INITIAL CONDITIONS AND PARAMETERS

Parameter	Value	Parameter	Value
$\boldsymbol{p}_{\mathcal{I}}(t_0)$ (m)	$[1,1,2]^{\mathrm{T}}$	m (kg)	0.68
$\boldsymbol{v}_{\mathcal{I}}(t_0)$ (m/s)	$[0,0,0]^{\mathrm{T}}$	$\tilde{t_l}$ (s)	2.45
$\boldsymbol{q}_{\mathcal{IB}}(t_0)$	$[1,0,0,0]^{\mathrm{T}}$	$\tilde{t_f}$ (s)	3.5
$\boldsymbol{\omega}_{\mathcal{B}}(t_0)$ (deg/s)	$[0,0,0]^{\mathrm{T}}$	l_{arm} (m)	0.17
$\boldsymbol{u}_T(t_0)$ (N)	$[0,0,mg]^{\mathrm{T}}$	c_τ(m)	0.05
$\boldsymbol{T}_{rotor}(t_0)$ (N)	$[mg,mg,mg,mg]^{\mathrm{T}}/4$	N	20
$J_{\mathcal{B}}$	diag([0.007, 0.007, 0.012])	h_s (m)	1
$\omega_{\max}$ (rad/s)	1.0	u_{max} (N)	20
ϵ_{tr}	1×10^{-2}	u_{min} (N)	5
ϵ_{vc}	1×10^{-4}	w_{tr}	10
w_{vc}	1×10^{4}	l	10
H_1	diag([1, 1, 1])	r_1	$\boldsymbol{p}_{\mathcal{I}}(t_f)$
$\boldsymbol{p}_{\mathcal{I}}(t_f)$ (m)	$[4,4,2.2]^{\mathrm{T}}$	T_{max} (N)	20
$\boldsymbol{v}_{\mathcal{I}}(t_f)$ (m/s)	$[0,0,0]^{\mathrm{T}}$	T_{min} (N)	5
$\boldsymbol{q}_{\mathcal{IB}}(t_f)$	$[1,0,0,0]^{\mathrm{T}}$	γ_c (deg)	45
$\boldsymbol{\omega}_{\mathcal{B}}(t_f)$ (deg/s)	$[0,0,0]^{\mathrm{T}}$	$\vartheta_{\max}$ (deg)	80

Firstly, we will present the simulation results of the nominal aerial landing. The parameter settings for this simulation are

shown in Table I. The planned trajectory and MPC tracking results are depicted in Figs. 4 and 5, demonstrating the quadrotor's ability to land on the AVC with agility and accuracy. For the 3-DoF problem, it efficiently computes the trajectory in a mere 10.02 ms. Furthermore, by implementing the SCP algorithm iteratively, based on the 3-DoF trajectory, it achieves a remarkable total solution time of 184.60 ms after 4 iterations (also compared with [32], which also utilized full dynamics of quadrotors). The average solving time of MPC is 2.14 ms, and the average tracking error (root mean square error, RMSE) is 6.86 cm. The small average error confirms the dynamical feasibility of the trajectory.

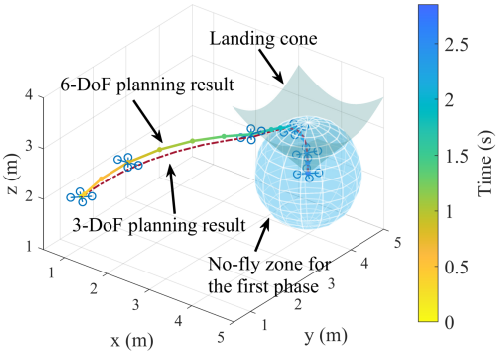


Fig. 4. Planning results for the Gazebo simulation.

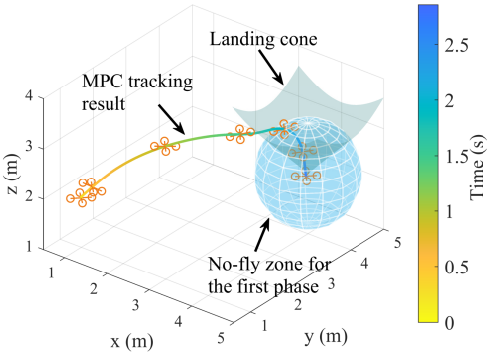


Fig. 5. MPC tracking result in Gazebo.

Additionally, Fig. 6 shows that the planning result satisfies the control constraints. It is worth noting that the rotor thrusts calculated by the MPC during tracking may slightly violate control constraints. This discrepancy between actual control and planned control is necessary to counteract errors caused by model mismatches and disturbances. A similar observation can also be found in [32], which offers valuable insights for trajectory design. While our method allows for precise thrust boundaries, it is still advisable to set a slightly conservative thrust boundary to ensure the quadrotor retains sufficient residual thrust to withstand disturbances or model mismatches.

One notable difference between landing on an AVC and landing on a ground or surface platform is the AVC's free position in three-dimensional space. If the quadrotor is initially

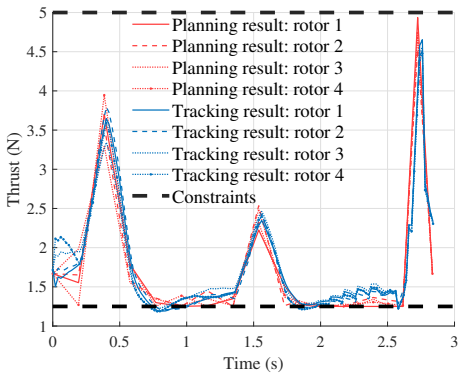


Fig. 6. Control profile of the simulation.

TABLE II
INITIAL CONDITIONS AND PARAMETERS

Parameter	Value	Parameter	Value
$\boldsymbol{p}_{\mathcal{I}}(t_0)$ (m)	$[1, 1, 2]^{\mathrm{T}}$	$\tilde{t}_l$ (s)	2.0
$\boldsymbol{p}_{\mathcal{I}}(t_f)$ (m)	$[1, 1.2, 4.2]^{\mathrm{T}}$	$\tilde{t}_f$ (s)	2.5

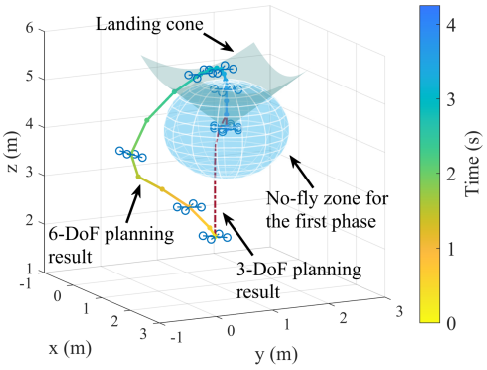


Fig. 7. Planning results for the Gazebo simulation, where the quadrotor is initially below the AVC.

positioned below the AVC, a direct approach to the AVC may result in a collision. The simulation results presented here demonstrate that our proposed method can still secure the quadrotor's safe and accurate landing in such a situation. Table II provides the simulation parameters, with the parameters not shown being the same as those in Table I. Figs. 7 and 8 showcase the planned trajectory and MPC tracking results. It takes 11.09 ms to compute the 3-DoF trajectory, and 223.46 ms to compute the 6-DoF trajectory after 5 iterations. It can be observed that the quadrotor first avoids collision and then lands safely on the AVC. The average solving time of MPC is 2.19 ms, and the average tracking RMSE is 3.76 cm.

B. Real-world Experiments

To further confirm the effectiveness of the proposed method, we conducted real-world experiments on both static and moving AVCs. For this purpose, we adapted a quadrotor as the AVC, while employing a smaller quadrotor to land on the AVC, depicted in Fig. 9. Both quadrotors are equipped with Pixhawk and PX4 firmware, which serve as the flight control

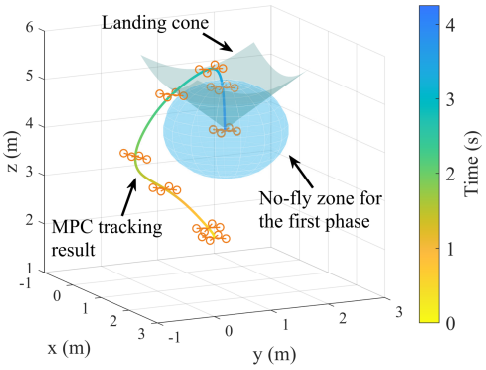


Fig. 8. MPC tracking result in Gazebo, where the quadrotor is initially below the AVC.

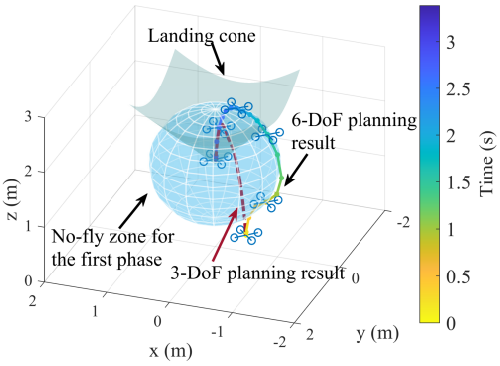


Fig. 10. Planning results for the experiment of landing on a static AVC, where the quadrotor is initially below the carrier.

systems. Our computing unit, Intel NUC 12 with Core i5-1250P, is responsible for calculating control commands using NMPC and transmitting them to the Pixhawk via WiFi. The inner-loop attitude controller employs a PID controller with PX4 firmware. To obtain accurate positioning information, we used the VICON motion capture system.

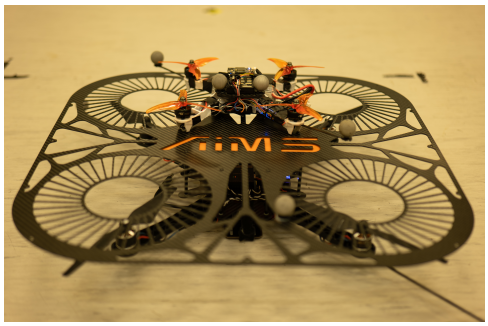


Fig. 9. The in-house AVC and an off-the-shelf small quadrotor.

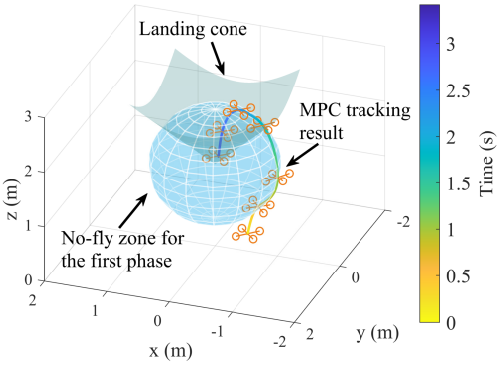


Fig. 11. Real-world flight of the experiment of landing on a static AVC, where the quadrotor is initially below the AVC.

TABLE III
INITIAL CONDITIONS AND PARAMETERS

Parameter	Value	Parameter	Value
$\boldsymbol{p}_{\mathcal{I}}(t_0)$ (m)	$[-0, 5, 0, 0.3]^{\mathrm{T}}$	m (kg)	0.661
$\boldsymbol{v}_{\mathcal{I}}(t_0)$ (m/s)	$[0, 0, 0]^{\mathrm{T}}$	$\tilde{t}_l$ (s)	1.8
$\boldsymbol{q}_{\mathcal{IB}}(t_0)$	$[1, 0, 0, 0]^{\mathrm{T}}$	$\tilde{t}_f$ (s)	3.0
$\boldsymbol{\omega}_{\mathcal{B}}(t_0)$ (deg/s)	$[0, 0, 0]^{\mathrm{T}}$	l_{arm} (m)	0.125
$\boldsymbol{u}_T(t_0)$ (N)	$[0, 0, mg]^{\mathrm{T}}$	c_τ(m)	0.016
$\boldsymbol{T}_{rotor}(t_0)$ (N)	$[mg, mg, mg, mg]^{\mathrm{T}}/4$	N	20
$J_{\mathcal{B}}$	$\mathrm{diag}([2.1, 2.5, 4.1] \times 10^{-3})$	h_s (m)	1
$\omega_{\max}$ (rad/s)	1.0	u_{max} (N)	30
ϵ_{tr}	1×10^{-2}	u_{min} (N)	4
ϵ_{vc}	1×10^{-4}	w_{tr}	10
w_{vc}	1×10^{4}	l	10
H_1	$\mathrm{diag}([1, 1, 1])$	r_1 (m)	$\boldsymbol{p}_{\mathcal{I}}(t_f)$
$\boldsymbol{p}_{\mathcal{I}}(t_f)$ (m)	$[0, 0, 1.5]^{\mathrm{T}}$	T_{max} (N)	30
$\boldsymbol{v}_{\mathcal{I}}(t_f)$ (m/s)	$[0, 0, 0]^{\mathrm{T}}$	T_{min} (N)	4
$\boldsymbol{q}_{\mathcal{IB}}(t_f)$	$[1, 0, 0, 0]^{\mathrm{T}}$	γ_c (deg)	45
$\boldsymbol{\omega}_{\mathcal{B}}(t_f)$ (deg/s)	$[0, 0, 0]^{\mathrm{T}}$	$\vartheta_{\max}$ (deg)	80

We begin by discussing the experiments conducted on a static AVC. The experimental results for the nominal landing closely align with the simulations, hence these specific results have been excluded from this paper. However, these results can be found in the supplementary video. Similar to the simulation, we also conducted experiments with the quadrotor's initial position below the AVC to verify the collision avoidance capability of the algorithm. The relevant parameters are listed in Table III . The planned trajectory and corresponding flight trajectory are presented in Figs. 10 and 11, respectively. These results demonstrate the quadrotor's successful collision avoidance and subsequent accurate and safe landing on the AVC. Notably, the 3-DoF trajectory calculation required only 10.83 ms, while the SCP algorithm achieved the 6-DoF trajectory after five iterations within 253.95 ms. The MPC exhibited an average solving time of 1.92 ms, with an average position RMSE of 12.76 cm.

In experiments involving a moving AVC, the trajectory of the AVC is circular with a maximum speed of 1 m/s. The initial conditions and parameters are shown in Table IV, and the unlisted parameters are the same as those in Table III. It is assumed that the quadrotor knows the AVC's trajectory, and it cooperatively adjusts its trajectory to land on the moving AVC. This assumption is reasonable in various scenarios, such as the AVC carrying the quadrotor to deliver a package to a specific location, with the quadrotor needing to return to the AVC after the delivery. The planned and flight trajectories are depicted in Figs. 12 and 13, respectively. The quadrotor successfully

TABLE IV
INITIAL CONDITIONS AND PARAMETERS

Parameter	Value	Parameter	Value
$\boldsymbol{p}_{\mathcal{I}}(t_0)$ (m)	$[-1, 5, 1.5, 0.5]^{\mathrm{T}}$	$\tilde{t}_l$ (s)	2.28
$\boldsymbol{p}_{\mathcal{I}}(t_f)$ (m)	$[1.706, -0.708, 0.5]^{\mathrm{T}}$	$\tilde{t}_f$ (s)	3.8
$\boldsymbol{v}_{\mathcal{I}}(t_f)$ (m/s)	$[0.71, 0.70, 0]^{\mathrm{T}}$	$\boldsymbol{q}_{\mathcal{IB}}(t_f)$	free
γ_c (deg)	30		

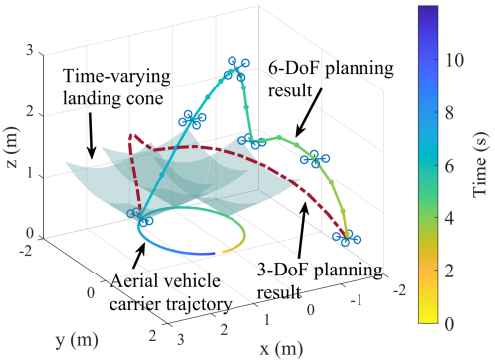


Fig. 12. Planning results for the experiment of landing on a moving AVC.

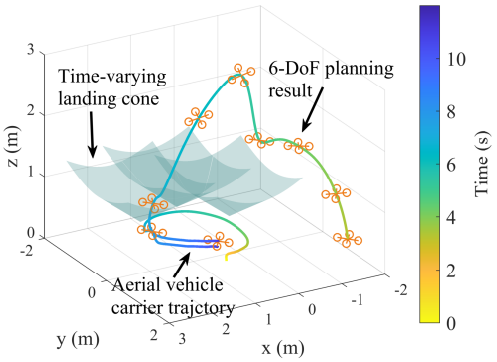


Fig. 13. Real-world flight of the experiment of landing on a moving AVC.

TABLE V
NUMBER OF ITERATIONS FOR
DIFFERENT INITIALIZATION METHODS TO CONVERGE

Initialization \ Case	1	2	3	4
Zero initialization	5	6	6	5
Straight-line initialization	4	Failure*	Failure	5
Ours	4	5	5	5

* It fails to converge after 50 iterations.

meets the time-varying landing cone constraints and ultimately lands smoothly and safely on the AVC. For this scenario, the trajectory calculation time for the 3-DoF model is 10.69 ms, while the SCP algorithm requires 230.28 ms for 5 iterations. The average solving time of MPC is 1.03 ms, and the position RMSE is 12.74 cm.

Please note that the trajectory of the 3-DoF system shown in Fig. 12 deviates significantly from the 6-DoF trajectory due to the absence of time-varying cone constraints. The inclusion of time-varying cone constraints ensures a consistent motion pattern with the AVC during the quadrotor's landing, consequently minimizing the impact of downwash airflow on the AVC caused by the quadrotor. Without these constraints, the downwash airflow acts on the AVC's edge first, generating considerable torque and disturbance that ultimately leads to experimental failure (see also the supplementary video). It should also be mentioned that the AVC employs NMPC controllers in all experiments. However, a different NMPC controller is utilized when the quadrotor is positioned above the AVC to compensate for the quadrotor's mass. As a result, our research focuses more on quadrotor landing rather than adaptive control for AVCs. The seamless transition between two NMPCs on an AVC guarantees a smooth landing and flight, thanks to the effectiveness of the proposed method in achieving precise, agile, and safe landings.

The LC-based 3-DoF trajectory can be used as an initial guess to aid in the convergence of SCP. To demonstrate the significance of this LC initialization, we compare this approach with two alternatives: 1) Zero initialization, where the variables are all set to zero, and 2) Straight-line initialization, which involves linear interpolation based on the initial and final states and was used in Section VII of [34]. These initialization methods are tested on the 4 experiments, and the corresponding results are presented in Table V.

Results in Table V reveal that the LC-based initialization converges in fewer iterations in certain cases, while the straight-line initialization fails to converge in scenarios where the initial position of the quadrotor lies below the AVC. This demonstrates the necessity of LC initialization for SCP convergence. Furthermore, the computational cost of the LC-based 3-DoF trajectory planning is only approximately 10 ms, whereas the SCP iteration takes 40-50 ms per iteration. Considering this, there is no reason to overlook its adoption as an initialization method. Moreover, the proposed 3-DoF trajectory planning method can serve as a standalone planning approach, enabling practitioners to deploy it on low-cost UAVs with limited computational resources, especially in tasks that do not necessitate collision avoidance.

V. CONCLUSION

This article proposes a safe, agile, and accurate trajectory planning algorithm for quadrotor landing on AVCs using a combination of a 3-DoF planning method based on LC theory and an SCP algorithm. The 3-DoF planning method, which can compute the trajectory in approximately 10 ms, incorporates landing cone constraints to ensure the safety of the drone during landing. By solving a single convex optimization problem, a 3-DoF safe landing trajectory for the quadrotor is obtained. The SCP algorithm, which can compute trajectories at a rate of 4-5 Hz for 6-DoF dynamics, iteratively solves convex subproblems considering collision-avoidance constraints to ensure the safety of the entire flight process. Both simulation and experimental results demonstrate the dynamical feasibility of the trajectories obtained by the proposed methods. The nominal NMPC is used for trajectory tracking, resulting in a position error of approximately 10 cm in the experiments, which highlights the effectiveness and

accuracy of the proposed algorithm in real-world scenarios. This article develops a practical trajectory planning approach for quadrotors landing on AVCs. The computational efficiency of the methods allows for real-time trajectory generation, making them suitable for practical applications.

REFERENCES

[1] F.-Y. Wang, Y. Lin, P. A. Ioannou, L. Vlacic, X. Liu, A. Eskandarian, Y. Lv, X. Na, D. Cebon, J. Ma, L. Li, and C. Olaverri-Monreal, "Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[2] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2023.

[3] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, X. Zhe, F. Zhu, and L. Chen, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3692–3711, 2023.

[4] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 970–11 981, 2022.

[5] B. Li, L. Li, T. Acarman, Z. Shao, and M. Yue, "Optimization-based maneuver planning for a tractor-trailer vehicle in a curvy tunnel: A weak reliance on sampling and search," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 706–713, 2022.

[6] H. Zhong, Y. Wang, Z. Miao, L. Li, S. Fan, and H. Zhang, "A homography-based visual servo control approach for an underactuated unmanned aerial vehicle in GPS-denied environments," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1119–1129, 2023.

[7] W. Yao, Y. Chen, J. Fu, D. Qu, C. Wu, J. Liu, G. Sun, and L. Xin, "Evolutionary utility prediction matrix-based mission planning for unmanned aerial vehicles in complex urban environments," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1068–1080, 2023.

[8] F. Wang, J. Huang, K. H. Low, and T. Hu, "Collective navigation of aerial vehicle swarms: A flocking inspired approach," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2023.

[9] H. Huang, Z. Shen, C. Huang, Y. Wang, and F.-Y. Wang, "Intelligent vehicle carriers to support general civilian purposes," *IEEE Transactions on Intelligent Vehicles*, pp. 1–4, 2023.

[10] H. Huang and A. V. Savkin, "Aerial surveillance in cities: When UAVs take public transportation vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1069 – 1080, 2023.

[11] P. Sun, B. Zhu, and S. Li, "Vision-based prescribed performance control for UAV target tracking subject to actuator saturation," *IEEE Transactions on Intelligent Vehicles*, pp. 1–8, 2023.

[12] H. Huang, A. V. Savkin, and C. Huang, "Reliable path planning for drone delivery using a stochastic time-dependent public transportation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4941–4950, 2021.

[13] Y. Lin, X. Na, D. Wang, X. Dai, and F.-Y. Wang, "Mobility 5.0: Smart logistics and transportation services in cyber-physical-social systems," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3527–3532, 2023.

[14] A. V. Savkin and H. Huang, "Asymptotically optimal path planning for ground surveillance by a team of UAVs," *IEEE Systems Journal*, vol. 16, no. 2, pp. 3446–3449, 2022.

[15] S. Zhou, X. Dong, Y. Hua, J. Yu, and Z. Ren, "Predefined formation-containment control of high-order multi-agent systems under communication delays and switching topologies," *IET Control Theory & Applications*, vol. 15, no. 12, pp. 1661–1672, 2021.

[16] B. Herissé, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[17] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2202–2207.

[18] T. Baca, P. Stepan, V. Spurny, D. Hert, R. Penicka, M. Saska, J. Thomas, G. Loianno, and V. Kumar, "Autonomous landing on a moving vehicle with an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 5, pp. 874–891, 2019.

[19] C. R. Hayner, S. C. Buckner, D. Broyles, E. Madewell, K. Leung, and B. Açikmeşe, "HALO: Hazard-aware landing optimization for autonomous systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3261–3267.

[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2520–2525.

[21] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[22] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[23] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[24] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 966–984, 2018.

[25] S. Liu, K. Mohta, N. Atanasov, and V. Kumar, "Search-based motion planning for aggressive flight in SE(3)," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2439–2446, 2018.

[26] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[27] R. Liu, Z. Qu, G. Huang, M. Dong, T. Wang, S. Zhang, and A. Liu, "DRL-UTPS: DRL-based trajectory planning for unmanned aerial vehicles for data collection in dynamic IoT network," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1204–1218, 2023.

[28] S. Chen, Y. Li, Y. Lou, K. Lin, and X. Wu, "Learning real-time dynamic responsive gap-traversing policy for quadrotors with safety-aware exploration," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2271–2284, 2023.

[29] T. Hickling, N. Aouf, and P. Spencer, "Robust adversarial attacks detection based on explainable deep reinforcement learning for UAV guidance and planning," *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2023.

[30] V. Freire and X. Xu, "Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2023.

[31] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.

[32] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Science Robotics*, vol. 6, no. 56, p. eabh1221, 2021-07-21.

[33] D. Dueri, Y. Mao, Z. Mian, J. Ding, and B. Açikmeşe, "Trajectory optimization with inter-sample obstacle avoidance via successive convexification," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 1150–1156.

[34] R. Bonalli, T. Lew, and M. Pavone, "Analysis of theoretical and numerical properties of sequential convex programming for continuous-time optimal control," *IEEE Transactions on Automatic Control*, vol. 68, no. 8, pp. 4570–4585, 2023.

[35] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Açıkmeşe, "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Systems Magazine*, vol. 42, no. 5, pp. 40–113, 2022.

[36] M. Eskandari, H. Huang, A. V. Savkin, and W. Ni, "Model predictive control-based 3D navigation of a RIS-equipped UAV for LoS wireless communication with a ground intelligent vehicle," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2371–2384, 2023.

[37] B. Açıkmeşe and L. Blackmore, "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.

[38] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[39] M. W. Harris and B. Açıkmeşe, "Lossless convexification for a class of optimal control problems with quadratic state constraints," in *2013 American Control Conference (ACC)*, 2013, pp. 3415–3420.

[40] ——, "Lossless convexification of non-convex optimal control problems for state constrained linear systems," *Automatica*, vol. 50, no. 9, pp. 2304–2311, 2014.

[41] S. Kunhippurayil, M. W. Harris, and O. Jansson, "Lossless convexification of optimal control problems with annular control constraints," *Automatica*, vol. 133, p. 109848, 2021.

[42] M. Szmuk, D. Malyuta, T. P. Reynolds, M. S. Mceowen, and B. Açikmeşe, "Real-time quad-rotor path planning using convex optimization and compound state-triggered constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7666–7673.

[43] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, "Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 8, pp. 1399–1413, 2020.

[44] P. Scheffe, T. M. Henneken, M. Kloock, and B. Alrifaee, "Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 661–672, 2023.

[45] A. Hanson, *Visualizing Quaternions*, ser. Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann, 2006.

[46] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[47] C.-T. Chen, *Linear system theory and design*, 3rd ed., ser. The Oxford series in electrical and computer engineering. New York: Oxford University Press, 1999.

[48] D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, and J. M. Carson, "Discretization performance and accuracy analysis for the rocket powered descent guidance problem," in *AIAA Scitech 2019 Forum*, 2019, pp. 1–20.

[49] M. Szmuk and B. Acikmese, "Successive convexification for 6-dof mars rocket powered landing with free-final-time," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, pp. 1–15.

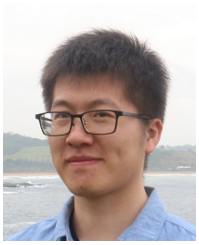
[50] P. Foehn, E. Kaufmann, A. Romero, R. Penicka, S. Sun, L. Bauersfeld, T. Laengle, G. Cioffi, Y. Song, A. Loquercio, and D. Scaramuzza, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight," *Science Robotics*, vol. 7, no. 67, p. eabl6259, 2022.

[51] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[52] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven MPC for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

Zhipeng Shen received the M.Eng. degree in control engineering from the Beihang University, Beijing, China, in 2022. Previously, he received the B.Eng. degree in aircraft design and engineering from the Northwestern Polytechnical University, Xi'an, China, in 2019. He is currently pursuing the Ph.D. degree at the Hong Kong Polytechnic University. His research interests include trajectory planning, machine learning, and optimal control in robotics.

Guanzhong Zhou received the B.Eng. degree from the Harbin Institute of Technology, Weihai, China, in 2019. He is now pursuing his master's degree at the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. His research interests include quadruped robots, mobile robots (UGV, UAV), and swarm robotics.



Hailong Huang received his Ph.D degree in Systems and Control from the University of New South Wales, Sydney, Australia, in 2018. He was a post-doctoral research fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia.
He is now an Assistant Professor at the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. His current research interests include guidance, navigation, and control of UAVs and mobile robots.
He is an Associate Editor of IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Vehicles, Intelligent Service Robotics, and International Journal of Advanced Robotic Systems, an editorial board member of the International Journal of Dynamics and Control, a guest editor of IEEE Transactions on Automation Science and Engineering.

Chao Huang received her Ph.D. degree in control engineering from the University of Wollongong, Wollongong, NSW, Australia, in 2018. She is currently a Research Assistant Professor at the Department of Industrial and System Engineering, The Hong Kong Polytechnical University (PolyU), Hong Kong. Prior to PolyU, she was a Research Fellow with Nanyang Technological University (NTU), Singapore, and the National Institutes of Informatics (NII), Tokyo, Japan. Her research interests include human–machine collaboration, fault-tolerant control, mobile robot (EV, UAV), and path planning and control. She is an Associate Editor of IEEE Transactions on Intelligent Vehicles, IEEE Transactions on Transportation Electrification, and IET Intelligent Transport Systems.

Yutong Wang received her Ph.D. degree in control theory and control engineering from the University of Chinese Academy of Sciences in 2021. She is currently an Assistant Professor at the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. Her research interests cover computer vision, intelligent vehicles, foundation models, decision intelligence, and CPSS.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined The University of Arizona, Tucson, AZ, USA, in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS, and the Vice President of Institute of Automation, CAS, in 2006. He found CAS Center for Social Computing and Parallel Management in 2008 and became the State Specially Appointed Expert and the Founding Director of the State Key Laboratory for Management and Control of Complex Systems in 2011. His research interests include parallel intelligence, social computing, and knowledge automation.