2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Autonomous Aerial Vehicle Carrier and MAV Collaboration: System Design, Trajectory Optimization, and Real-world Implementation

Zhipeng Shen†, Guanzhong Zhou†, Peng Sun, Bowen Lan, Qingyang Meng, Hao Cao, Shiyu Zhou, Jinliang Shao, Hailong Huang*

***Abstract*— The rapid growth of the low-altitude economy has spotlighted the utility of Micro Aerial Vehicles (MAVs) in various applications. However, the limited operational range of MAVs due to their size and weight constraints poses significant challenges. This paper investigates the concept of the Aerial Vehicle Carrier (AVC), a larger vehicle designed to enhance the mission radius of MAVs by transporting them to designated task areas. The synergy between the maneuverable MAVs and the enduring AVC creates a collaborative system that effectively overcomes individual limitations and significantly boosts operational efficiency in practical applications. A novel trajectory optimization algorithm has also been developed for the AVC collaboration. This study highlights the development and real-world deployment of a fully autonomous AVC collaboration system prototype, representing a significant leap over previous experimental setups. This paper details the deployment of the AVC system in real-world environments, demonstrating its effectiveness and robustness for AVC-MAV collaboration.**

Fig. 1. The AVC and MAV collaboration system operates in a real-world environment.

I. INTRODUCTION

Due to the burgeoning low-altitude economy, Micro Aerial Vehicles (MAVs) have garnered significant attention for their exceptional maneuverability and ability to traverse difficult terrains. MAVs are employed in a variety of applications including parcel delivery ([1]), traffic monitoring ([2]), and surveillance ([3]). However, the size and weight constraints of MAVs typically limit their operational range, which in turn restricts their application scenarios. This article introduces a novel concept, the Aerial Vehicle Carrier (AVC) ([4]), which is designed to extend the mission radius of MAVs. Compared to MAVs, the AVC is a larger vehicle with a design focus on enhancing operational range.

By using the AVC to transport MAVs to the designated task area as demonstrated in Fig. 1, the AVC-MAV collaboration system leverages the maneuverability and passability of MAVs to execute tasks effectively, particularly in time-critical operations such as search and rescue. This solution capitalizes on the maneuverability and passability of MAVs while also utilizing the endurance capabilities of the AVC. Due to its considerable size and weight, the AVC exhibits limited maneuverability and passability; however, in conjunction with MAVs, the two systems complement each other's limitations, creating a synergistic effect that enhances overall operational efficiency.

A. Related works

MAV-related technologies have seen substantial advancements. For example, a trajectory optimization method for multi-rotor systems has been proposed in [5]. [6] introduced a method for constructing a low-cost quadcopter experimental platform. [7] and [8] explored using Incremental Nonlinear Dynamic Inversion (INDI) for robust anti-interference flight. [9] conducted a comparative study that analyzed common control methods for MAVs, such as Nonlinear Model Predictive Control (NMPC) and control based on differential flatness.

As MAV technology has matured, the focus has shifted toward collaboration among multiple MAVs. A fault-tolerant task planning method for groups of MAVs was proposed in [10]. [11] developed an indoor formation motion planning method for MAVs based on B-splines and evolutionary optimization. The control of flying parallel robots, suitable for collaborative transportation of MAVs, was studied in [12]. A virtual tube-based MAV swarm planning and control method was introduced in [13]. [14] studied the relative positioning of air-ground robot formations using onboard perception.

This work was supported by the Research Centre for Low Altitude Economy (RCLAE), the Hong Kong Polytechnic University. ∗*Corresponding author: Hailong Huang.*

†Equal contribution

Z. Shen, G. Zhou, B. Lan, Q. Meng, H. Cao, and H. Huang are with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, China (e-mail: hailong.huang@polyu.edu.hk, {zhipeng.shen, newtt.zhou, bowen.lan, qingyang.meng, hao.cao}@connect.polyu.hk).

P. Sun and J. Shao are with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, China (e-mail: jinliangshao@uestc.edu.cn, pengsun@std.uestc.edu.cn).

S. Zhou is with the Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: shiyuzhou9-c@my.cityu.edu.hk).

[15] introduced the RACER approach, which leverages a fleet of decentralized MAVs for efficient and autonomous exploration, using innovative techniques in spatial partitioning, workload balancing, and hierarchical planning to achieve unprecedented real-world collaborative exploration. [16] presented a software-defined, cooperative multiple UAV-enabled aerial computing system that enhances resource utilization and load balancing, significantly improving task processing efficiency and reducing energy consumption in advanced 5G environments. This article will explore collaborative operations between the AVC and the MAV to expand the operational scope of MAVs.

Trajectory optimization is a critical technology for ensuring the effective performance of MAVs in various applications. Regarding the AVC collaborative system, [4] focused on trajectory optimization for a MAV landing on the AVC, achieving the successful landing on a hovering or moving AVC. In [17], the trajectory optimization for the MAV landing on the AVC was further developed based on [5], incorporating the landing cone constraint from [4] to minimize the aerodynamic interference. This article will develop a Sequential Convex Programming (SCP) algorithm for trajectory optimization in the AVC collaboration. The SCP algorithm transforms the original trajectory optimization problem into convex subproblems, iteratively solving these to approximate the solution of the original problem. Due to its wide applicability, the SCP algorithm has been employed in various applications ([18]). [19] and [20] designed trajectory optimization methods for six-degrees-of-freedom powered descent landing. In [21], a rigorous theoretical analysis of the SCP algorithm was conducted. An SCP algorithm with guaranteed feasibility based on the augmented Lagrangian method was proposed in [22]. In [23], the optimal control problem of stochastic dynamic systems based on SCP was studied. The versatility of the SCP algorithm allows the method proposed in this paper to be adaptable to various configurations, e.g., tiltable quadrotors ([24]).

B. Contributions

This work designed, constructed, and tested a prototype of the AVC-MAV collaboration system in real-world scenarios to validate its effectiveness. The main contributions of this study are outlined as follows:

1) This work designs and builds the first fully autonomous AVC-MAV collaboration system prototype. Compared with previous experimental setups in [4] and [17], this prototype eliminates the dependence on motion capture systems, enabling its rapid field deployment in GPS-denied environments.
2) This research develops a spatial-temporal trajectory optimization algorithm for the task of AVC-MAV collaboration. Unlike the previous study in [4], which relied on an assumption of ellipsoidal obstacles, the algorithm proposed in this paper ensures safe navigation through the flight corridor, thereby eliminating the need for this restrictive assumption. Besides, the adaptive time mesh is designed to dynamically allocate convex polyhedrons along the trajectory, thus configuring the flight corridor (see also Remark 1).
3) Moving beyond the controlled conditions of laboratory tests, this work successfully implements the AVC-MAV collaboration system in outdoor field tests. This achievement demonstrates the system's robustness and potential for broader applications.

The remainder of this paper is organized as follows. Section II introduces the AVC-MAV collaboration system. Section III presents the trajectory optimization algorithm for the AVC-MAV collaboration. Section IV shows the experiment results. Section V summarizes this paper.

II. THE PROTOTYPE OF THE AVC-MAV COLLABORATION SYSTEM

This section presents the prototype of the AVC-MAV collaboration system and provides a detailed description of its framework.

Fig. 2. The prototype of the AVC.

The prototype of the AVC-MAV collaboration system is depicted in Fig. 1. The AVC is detailed in Fig. 2. The system consists of one MAV and one AVC. The MAV is equipped with a LiDAR sensor, a flight control unit with PX4 firmware, and an onboard computer featuring the RK3588s chip. Similarly, the AVC includes a custom-designed deck, specifically tailored to support MAV operations. The AVC itself weighs 10.2 kg, while the deck adds an additional 2.9 kg, bringing the total weight to 13.1 kg. The AVC is engineered for longer ranges but lacks the MAV's maneuverability. The MAV, lighter in comparison, weighs only 0.928 kg and has a diagonal size of 15.2 cm. The AVC has a diagonal size of 106.8 cm and a battery life of approximately 19 minutes, significantly outlasting the MAV's 5-minute battery duration.

For the software framework of the AVC-MAV collaboration system, the perception employs the Point-LIO algorithm ([25]), while the control strategy utilizes NMPC similar to [4]. The trajectory optimization algorithm will be detailed in Section III.

III. TRAJECTORY OPTIMIZATION FOR THE AVC-MAV COLLABORATION

This section describes the SCP algorithm for trajectory optimization. The SCP algorithm can be applied to the AVC-MAV collaboration task. This section also introduces how

to generate a satisfactory initial guess by solving a convex optimization problem.

A. Notation

$\|\cdot\|$ denotes the Euclidean norm. $\mathscr{N}_p^q$ is the integer set from p to q, i.e., $\{p, p+1, \cdots, q-1, q\}$. $t \in \mathbb{R}$ denotes the time. $\boldsymbol{p}(t)$, $\boldsymbol{v}(t)$, $\boldsymbol{q}(t)$, $\boldsymbol{\omega}(t)$ are position, velocity, quaternion and angular velocity of the MAV, respectively. $\boldsymbol{x}(t) \in \mathbb{R}^{13}$ and $\boldsymbol{u}(t) \in \mathbb{R}^4$ denote the state and control of the nonlinear dynamics of the MAV, respectively. $f : \mathbb{R}^{13} \times \mathbb{R}^4 \to \mathbb{R}^{13}$ is the continuous-time nonlinear dynamics. $\boldsymbol{a}(t)$ is the acceleration of the MAV. $\boldsymbol{x}^{\vee}(t)$ is the state of the linear dynamics, consisting of the position and velocity of the MAV. I is an identity matrix. $\Omega(\cdot)$ is a skew-symmetric matrix. $J \in \mathbb{S}_+^3$ is the moment of inertia of the MAV. $\boldsymbol{g}$ is the gravity vector. m is the mass.

B. Trajectory optimization based on SCP

The rigid-body dynamics of the MAV are given as

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

$$\dot{\boldsymbol{v}}(t) = \frac{1}{m} R(\boldsymbol{q}(t)) \boldsymbol{T}(t) + \boldsymbol{g}, \tag{2}$$

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2} \Omega(\boldsymbol{\omega}(t)) \boldsymbol{q}(t), \tag{3}$$

$$J \dot{\boldsymbol{\omega}}(t) = \boldsymbol{\sigma}(t) - \boldsymbol{\omega}(t) \times J \boldsymbol{\omega}(t). \tag{4}$$

The total thrust $\boldsymbol{T}(t)$ and the torque $\boldsymbol{\sigma}$ can be presented as

$$\boldsymbol{T}(t) = M_T \boldsymbol{u}(t), \tag{5}$$

$$\boldsymbol{\sigma}(t) = M_\sigma \boldsymbol{u}(t), \tag{6}$$

where M_T and M_σ are matrices decided by the configuration of the MAV.

The nonlinear dynamics (1)-(6) can be represented as

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall t \in \left[t_0, t_f\right], \tag{7}$$

We consider a free-Final-Time (free-FT) Optimal Control Problem (OCP) as follows

$$\min_{\boldsymbol{u}(t), t_f} w_u \int_{t_0}^{t_f} \|\boldsymbol{u}(t)\| dt \quad + w_t \left(t_f - t_0\right), \tag{8a}$$

subject to:

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \quad \forall t \in [t_0, t_f], \tag{8b}$$

$$g(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leqslant 0_{n_g} \quad \forall t \in [t_0, t_f], \tag{8c}$$

$$h(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) = 0_{n_h} \quad \forall t \in [t_0, t_f], \tag{8d}$$

where $g(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$ is a convex function, and $h(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$ is affine. Here $g : [t_0, t_f] \times \mathbb{R}^{13} \times \mathbb{R}^4 \to \mathbb{R}^{n_g}$ and $h : [t_0, t_f] \times \mathbb{R}^{13} \times \mathbb{R}^4 \to \mathbb{R}^{n_h}$ denote the path inequality and equality constraint functions, respectively. The vectors $0_{n_g} \in \mathbb{R}^{n_g}$ and $0_{n_h} \in \mathbb{R}^{n_h}$ denote the zero vectors of dimensions n_g and n_h.

To solve this continuous-time problem using an optimization algorithm, the problem must be discretized. Since the problem is free-FT, it cannot be discretized. The problem can be equivalently transformed into a fixed-final-time (fixed-FT) problem. A scaled time is defined by $\tau \in [\tau_0, \tau_f]$, where τ_0 and τ_f are the fixed initial and final scaled times, respectively. The original t can be determined by

$$s(\tau) \triangleq \frac{dt(\tau)}{d\tau}, \ t(\tau_0) = t_0, \ t(\tau_f) = t_f. \tag{9}$$

Let $\dot{\varphi}(t) \triangleq \|\boldsymbol{u}(t)\|$, and $\breve{\boldsymbol{x}}(\tau) \triangleq [\boldsymbol{x}^T(t(\tau)), \varphi(t(\tau)), t(\tau)]^T$. The dynamics of the extended system is

$$\breve{\boldsymbol{x}}'(\tau) \triangleq \frac{d\breve{\boldsymbol{x}}(\tau)}{d\tau} = F(\breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau)) = \begin{bmatrix} f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \|\boldsymbol{u}(t)\| \\ 1 \end{bmatrix} s(\tau), \tag{10}$$

where $\breve{\boldsymbol{u}}(\tau) \triangleq [\boldsymbol{u}^T(t(\tau)), s(\tau)]^T$. The transformed fixed-FT OCP is as

$$\min_{\breve{\boldsymbol{u}}(\tau)} w_u \varphi\left(\tau_f\right) + w_t t\left(\tau_f\right), \tag{11a}$$

subject to:

$$\breve{\boldsymbol{x}}'(\tau) = F(\breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau)) \quad \forall \tau \in [\tau_0, \tau_f], \tag{11b}$$

$$\breve{g}(\tau, \breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau)) \leqslant 0_{n_{\breve{g}}} \quad \forall \tau \in [\tau_0, \tau_f], \tag{11c}$$

$$\breve{h}(\tau, \breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau)) = 0_{n_{\breve{h}}} \quad \forall \tau \in [\tau_0, \tau_f], \tag{11d}$$

where $\breve{g}(\tau, \breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau))$ is convex, $\breve{h}(\tau, \breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau))$ is affine.

The dynamics (10) is linearized as

$$\breve{\boldsymbol{x}}'(\tau) \approx A(\tau) \breve{\boldsymbol{x}}(\tau) + B(\tau) \breve{\boldsymbol{u}}(\tau) + \boldsymbol{c}(\tau), \tag{12}$$

$$A(\tau) \triangleq \left.\frac{\partial F(\breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau))}{\partial \breve{\boldsymbol{x}}(\tau)}\right|_{\tilde{\boldsymbol{z}}(\tau)}, B(\tau) \triangleq \left.\frac{\partial F(\breve{\boldsymbol{x}}(\tau), \breve{\boldsymbol{u}}(\tau))}{\partial \breve{\boldsymbol{u}}(\tau)}\right|_{\tilde{\boldsymbol{z}}(\tau)}, \tag{13}$$

$$\boldsymbol{c}(\tau) \triangleq -A(\tau) \tilde{\breve{\boldsymbol{x}}}(\tau) - B(\tau) \tilde{\breve{\boldsymbol{u}}}(\tau), \tag{14}$$

where $\tilde{\boldsymbol{z}}(\tau) \triangleq [\tilde{\breve{\boldsymbol{x}}}^T(\tau), \tilde{\breve{\boldsymbol{u}}}^T(\tau)]^T$ is the reference trajectory. $\tilde{\breve{\boldsymbol{x}}}(\tau)$ and $\tilde{\breve{\boldsymbol{u}}}(\tau)$ are the reference values of $\breve{\boldsymbol{x}}(\tau)$ and $\breve{\boldsymbol{u}}(\tau)$, respectively, which can be obtained from the last iteration. To solve the optimization problem, the continuous problem must be discretized.

To convert the original continuous-time problem into a discrete form, we introduce N uniformly distributed nodes. This process segments the problem into $N-1$ subintervals. Within each of these subintervals, we approximate the control input using first-order hold.

$$\breve{\boldsymbol{u}}(\tau) = \hat{\eta}_k(\tau) \breve{\boldsymbol{u}}_k + \eta_k(\tau) \breve{\boldsymbol{u}}_{k+1} \quad \forall \tau \in [\tau_k, \tau_{k+1}], \tag{15}$$

$$\hat{\eta}_k(\tau) = \frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}, \quad \eta_k(\tau) = \frac{\tau - \tau_k}{\tau_{k+1} - \tau_k}, \tag{16}$$

where subscript k represents the variable at the scaled time τ_k, i.e., $(\cdot)_k \triangleq (\cdot)(\tau_k)$.

The linearized dynamics, as described in (12), constitute a linear-time-varying (LTV) system. Utilizing the characteristics of the state transition matrix, we derive the discrete-time LTV dynamics for each index i within the set $\mathscr{N}_1^{N-1}$.

$$\breve{\boldsymbol{x}}_{i+1} = A_i \breve{\boldsymbol{x}}_i + \mathring{B}_i \breve{\boldsymbol{u}}_i + B_i \breve{\boldsymbol{u}}_{i+1} + \boldsymbol{c}_i + \boldsymbol{v}_i, \tag{17}$$

$$\Phi(\tau,\tau_i) \triangleq I + \int_{\tau_i}^{\tau} A(\zeta)\Phi(\zeta,\tau_i)\,\mathrm{d}\zeta, \tag{18a}$$

$$A_i \triangleq \Phi(\tau_{k+1},\tau_i), \tag{18b}$$

$$\check{B}_i \triangleq A_i \int_{\tau_i}^{\tau_{k+1}} \Phi^{-1}(\tau,\tau_i)\,\hat{\eta}(\tau)B(\tau)\mathrm{d}\tau, \tag{18c}$$

$$B_i \triangleq A_i \int_{\tau_i}^{\tau_{k+1}} \Phi^{-1}(\tau,\tau_i)\,\eta(\tau)B(\tau)\mathrm{d}\tau, \tag{18d}$$

$$\boldsymbol{c}_i \triangleq A_i \int_{\tau_i}^{\tau_{k+1}} \Phi^{-1}(\tau,\tau_i)\,\boldsymbol{c}(\tau)\mathrm{d}\tau, \tag{18e}$$

Linearization may inadvertently lead to artificial infeasibility, as noted in [19]. There are instances where the linearized version of a problem becomes infeasible, despite the feasibility of the original problem. This typically happens when the linearized constraints are at odds with each other due to the reliance on an impractical reference trajectory. To address this challenge, a virtual control term $\boldsymbol{v}_i \in \mathbb{R}^{13}$ is integrated into the linearized dynamics (17).

The OCP can be discretized as (19) for all $k \in \mathcal{N}_1^N$ and $i \in \mathcal{N}_1^{N-1}$.

$$\min_{\boldsymbol{z}_k,\boldsymbol{v}_i,\zeta_i,\xi_k} w_u\varphi(\tau_f) + w_t t(\tau_f) + w_{vc}\sum\zeta_i + w_{tr}\sum\xi_k, \tag{19a}$$

subject to:

$$\check{\boldsymbol{x}}_{i+1} = A_i\check{\boldsymbol{x}}_i + \check{B}_i\check{\boldsymbol{u}}_i + B_i\check{\boldsymbol{u}}_{i+1} + \boldsymbol{c}_i + \boldsymbol{v}_i, \tag{19b}$$

$$\check{g}(\tau_k,\check{\boldsymbol{x}}_k,\check{\boldsymbol{u}}_k) \leqslant 0_{n_{\check{g}}}, \tag{19c}$$

$$\check{h}(\tau_k,\check{\boldsymbol{x}}_k,\check{\boldsymbol{u}}_k) = 0_{n_{\check{h}}}, \tag{19d}$$

$$\|\boldsymbol{v}_i\| \leqslant \zeta_i, \tag{19e}$$

$$\|\boldsymbol{z}_k - \tilde{\boldsymbol{z}}_k\| \leqslant \xi_k. \tag{19f}$$

$\boldsymbol{z}_k \triangleq [\check{\boldsymbol{x}}_k^T, \check{\boldsymbol{u}}_k^T]^T$ is the trajectory to optimize. The variable $\boldsymbol{i}$ is introduced to prevent the infeasibility caused by linearization. The constraint (19e) and $w_{vc}\sum\zeta_i$ in (19a) ensure $\boldsymbol{v}_i$ only present when necessary. The trust region constraint (19f) and $w_{tr}\sum\xi_k$ in (19a) ensure the linearization is valid.

The problem (19) is a convex optimization problem. An SCP algorithm, Algorithm 1, is used to iteratively solve (19) to get a solution of (8). i_{max} is the maximum iteration. $\boldsymbol{z}$ consists of all the $\boldsymbol{z}_k$, i.e., $[\boldsymbol{z}_1^T,\cdots,\boldsymbol{z}_N^T]^T$. Similarly, $\tilde{\boldsymbol{z}}$, $\boldsymbol{v}$, $\boldsymbol{\zeta}$, and $\boldsymbol{\xi}$ consist of all the $\tilde{\boldsymbol{z}}_k$, $\boldsymbol{v}_i$, ζ_i, and ξ_k, respectively. ε_v and ε_{tr} are the thresholds of the convergence criteria.

Algorithm 1 Trajectory Optimization.

SCP $(\tilde{\boldsymbol{z}},w_t,w_u,w_{vc},w_{tr})$
  $i \leftarrow 0$
  while $\varepsilon_v \leqslant \|\boldsymbol{v}\|$ and $\varepsilon_{tr} \leqslant \|\boldsymbol{z}-\tilde{\boldsymbol{z}}\|$ and $i \leq i_{max}$ **do**
    solve problem (19) to get $(\boldsymbol{z},\boldsymbol{v},\boldsymbol{\zeta},\boldsymbol{\xi})$
    $\tilde{\boldsymbol{z}} \leftarrow \boldsymbol{z}$
    $i \leftarrow i+1$
  end while
  return $\boldsymbol{z}$

Remark 1: By transforming the OCP from Equation (8) to Equation (11), we introduce a new control variable $s(\tau)$ to adjust the time mesh. Consequently, the flight corridor, which consists of multiple connected convex polyhedra, does not require precise allocation. Instead, it utilizes the SCP algorithm to optimize the time mesh for time allocation. The constraints (19c) and (19d) will be detailed in Section III-C.

C. Trajectory optimization for safe AVC-MAV collaboration

The Safe Flight Corridor (SFC) ensures MAVs do not collide with obstacles. The SFC can be obtained using existing methods, e.g., [26]. The SFC consists of connected convex polyhedrons as shown in Fig. 3. Thus, the SFC can be described as

$$A_c(t)\boldsymbol{p}(t) \leqslant b_c(t). \tag{20}$$

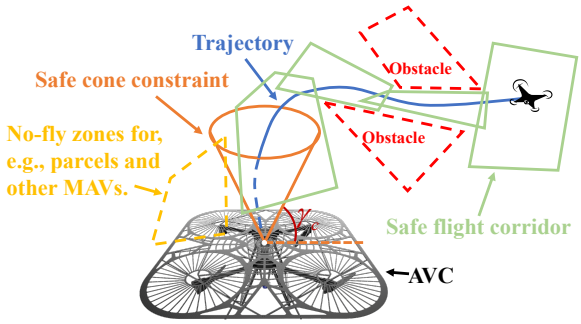

Fig. 3. Safe trajectory for AVC-MAV collaboration.

To ensure the safe departure and return of MAVs to the AVC, they must utilize a designated landing area. The following constraints are established to guarantee a safe landing at the specified point.

$$\left\|H_c(\boldsymbol{p}(t)-\boldsymbol{p}_f)\right\| \leqslant \cot(\gamma_c)\boldsymbol{e}(\boldsymbol{p}(t)-\boldsymbol{p}_f), \tag{21}$$

$$\boldsymbol{e} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, H_c \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{22}$$

where $\boldsymbol{p}_f$ is the landing point. This constraint enforces the trajectory to lie inside a cone defined by the angle γ_c as shown in Fig. 3.

The actuator limits can be enforced by

$$u_{min} \leqslant \boldsymbol{u}(t) \leqslant u_{max}, \tag{23}$$

where u_{min} and u_{max} are the lower and upper limits, respectively.

The boundary conditions are defined by

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \boldsymbol{x}(t_f) = \boldsymbol{x}_f. \tag{24}$$

The constraints (20)-(23) can be incorporated into (8c). The constraint (24) can be incorporated into (8d). Therefore, the SCP algorithm can seamlessly apply to safe AVC-MAV collaboration.

D. Initial guess for SCP

The SCP algorithm requires an initial guess to start the iteration. To utilize the convexity of the MAV system, we consider the following Linear Time-Invariant (LTI) system with a fixed final time ([4]).

$$\dot{\boldsymbol{x}}^{\vee}(t) = A^{\vee}\boldsymbol{x}^{\vee}(t) + B^{\vee}\boldsymbol{a}(t), \tag{25}$$

where $A^{\vee}$ and $B^{\vee}$ are constant matrices defining the LTI system.

By discretizing the system, an OCP can be constructed as (26) for all $k \in \mathcal{N}_1^{N-1}$, $j \in \mathcal{N}_{N-5}^{N}$, and $i \in \mathcal{N}_1^{N}$.

$$\min_{\boldsymbol{a}(t_k)} \sum \|\boldsymbol{a}(t_k)\|, \tag{26a}$$

subject to:

$$\boldsymbol{x}^{\vee}(t_{k+1}) = A_k^{\vee}\boldsymbol{x}^{\vee}(t_k) + B_k^{\vee}\boldsymbol{a}(t_k), \tag{26b}$$

$$\boldsymbol{x}^{\vee}(t_0) = x_0^{\vee}, \boldsymbol{x}^{\vee}(t_f) = x_f^{\vee}, \tag{26c}$$

$$A_c(t_i)\boldsymbol{p}(t_i) \leqslant b_c(t_i), \tag{26d}$$

$$\left\|H_c(\boldsymbol{p}(t_j) - \boldsymbol{p}_f)\right\| \leqslant \cot(\gamma_c)\boldsymbol{e}(\boldsymbol{p}(t_j) - \boldsymbol{p}_f). \tag{26e}$$

A satisfactory initial guess can be obtained by solving the convex optimization problem (26), which can be solved in several milliseconds as shown in Section IV.

IV. Experiments

This section presents the results of experiments conducted in both simulated and real-world scenarios.

A. Simulation experiments

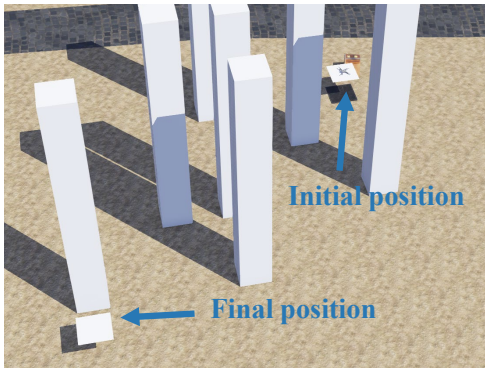

Fig. 4. Simulation scenario for Case 1, illustrating the MAV navigating between platforms in an obstacle-rich environment.

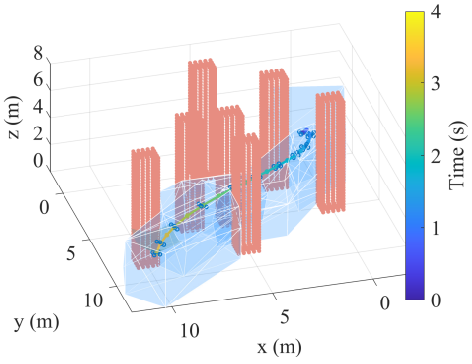

Fig. 5. Trajectory optimization results for Case 1.

We conducted tests of the trajectory optimization algorithm within the high-fidelity simulation environment provided by Webots[1]. Flight corridors were generated by simulating point clouds and utilizing the method proposed by [26]. The simulation results were obtained on a computer equipped with an Intel Core i7-12700KF processor. The convex optimization problems in Algorithm 1 were solved using CVXPY and CLARABEL ([27], [28]). Additionally, the algorithm's performance was enhanced by Numba and the disciplined parametrized programming framework of CVXPY[2] ([29]). The detailed parameters of the simulation experiment are presented in Table I.

TABLE I
Parameters for Trajectory Optimization

Parameter	γ_c (deg)	w_t	w_u	w_{vc}	w_{tr}
Value	45	0	1	1000	10

The initial simulation scenario, depicted in Fig. 4, involves a MAV navigating from one platform to another amidst numerous obstacles. The trajectory, optimized by the proposed algorithm, is illustrated in Fig. 5, effectively demonstrating compliance with various safety restrictions. The trajectory optimization algorithm successfully converges after eight iterations of solving the convex subproblem, achieving a computational time of 0.171 seconds.

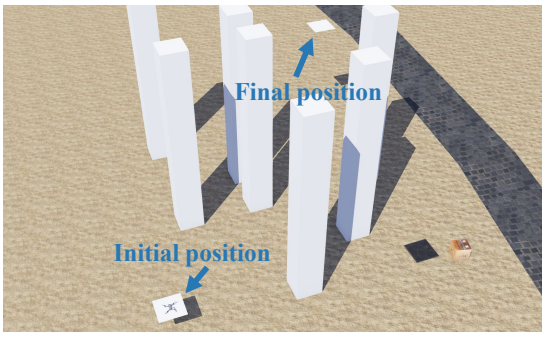

Fig. 6. Simulation scenario for Case 2, illustrating the MAV navigating between platforms in an obstacle-rich environment.

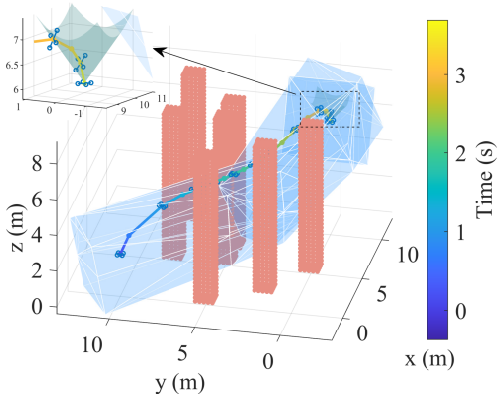

Fig. 7. Trajectory optimization results for Case 2.

In the second scenario of the simulation, depicted in Fig. 6, the MAV is required to ascend to a high platform. The

[1]http://www.cyberbotics.com
[2]https://www.cvxpy.org/tutorial/dpp/index.html

results of the trajectory optimization are presented in Fig. 7, demonstrating that the MAV adheres to both the landing cone and flight corridor constraints. The landing cone constraint is crucial not only for ensuring the MAV's safety during its approach to the AVC but also for mitigating the effects of the AVC's airflow when approaching laterally. Based on these results, we successfully implemented the algorithm in the real-world environment. The trajectory optimization algorithm reached convergence after 12 iterations, with a computation time of 0.270 seconds.

B. Real-world experiments

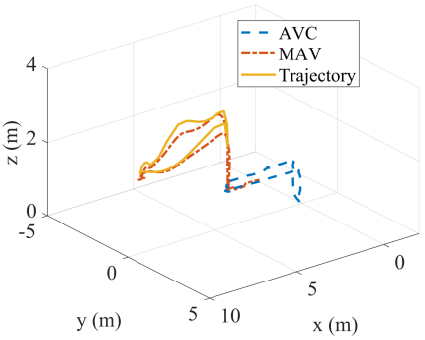

Fig. 8. Real-world experimental trajectory.

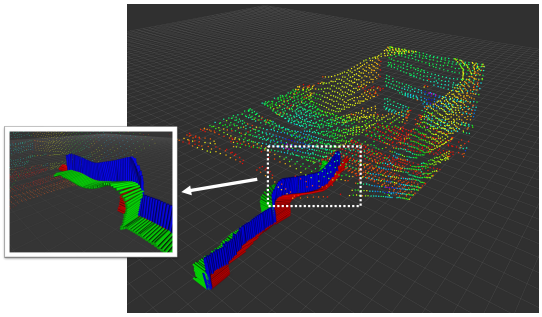

Fig. 9. Indoor experiment point clouds and the odometry of the MAV.

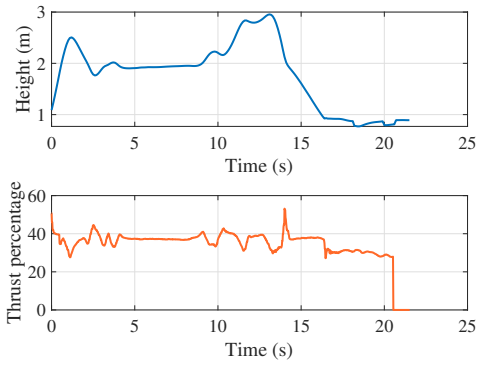

Fig. 10. Thrust percentage and height of the MAV during flight.

We developed a prototype of the AVC-MAV collaboration system and validated the system through physical experiments. All tests were conducted autonomously without reliance on external positioning systems such as motion capture or GPS. Initially, we conducted verification tests in an indoor environment, as illustrated in Figs. 8 and 11. After the AVC transported the MAV over a specific distance, the MAV departed to simulate task execution, followed by a precise landing on the AVC, guided by the optimized trajectory.

During testing, we observed that due to the significantly larger mass of the AVC relative to the MAV, the MAV required approximately 45% of its thrust capacity during takeoff to counteract the suction effect generated by the AVC. This effect proved so substantial that the MAV was unable to initiate takeoff when commanded to ascend only 0.5 m above the AVC. Success was achieved only after increasing the target height to 1.0 m, which prompted the MPC to generate larger thrust commands sufficient to overcome the strong aerodynamic interaction. As shown in Fig. 10, while the ascent rate of the MAV remained relatively stable during takeoff, an observable alteration in the slope of the thrust curve was recorded, indicating a change in thrust requirements during this phase. Interestingly, the aerodynamic forces exerted by the AVC may also act as a passive fastening mechanism for the MAV, helping to secure it on the deck and prevent unintended movement. The proposed algorithm optimizes the landing process by ensuring that the MAV is primarily influenced by vertical forces, which are easier to manage. Surprisingly, the aerodynamic characteristics of the AVC, initially perceived as potential obstacles, appear to offer unexpected benefits, improving system functionality and safety.

We extended the deployment of the AVC-MAV collaboration system to outdoor environments, as illustrated in Figs. 12 and 13. In these settings, the quality of the point cloud data generated by the LiDAR is expected to diminish significantly due to environmental factors such as wind-blown leaves and expansive open spaces, which reduce the density of usable point cloud data for positioning algorithms as shown in Figs. 9 and 14. Despite these challenges, the designed AVC-MAV collaboration system works effectively in such environments, successfully completing collaborative tasks with the MAV.

Observations from Figs. 8 and 13 indicate that the tracking accuracy of outdoor experiments is significantly lower than that of indoor experiments. This discrepancy is primarily due to the challenges posed by the expansive outdoor environment, which complicates the performance of perception algorithms and leads to inaccurate state estimation. Additionally, outdoor wind disturbances significantly affect the performance of lower-quality drones. To address these challenges, our future work will focus on developing perception algorithms tailored to outdoor conditions, including windy environments, enabling drones to effectively collaborate with the AVC-MAV collaboration system in complex settings. This article primarily concentrates on system design, trajectory optimization, and experimental verification, with

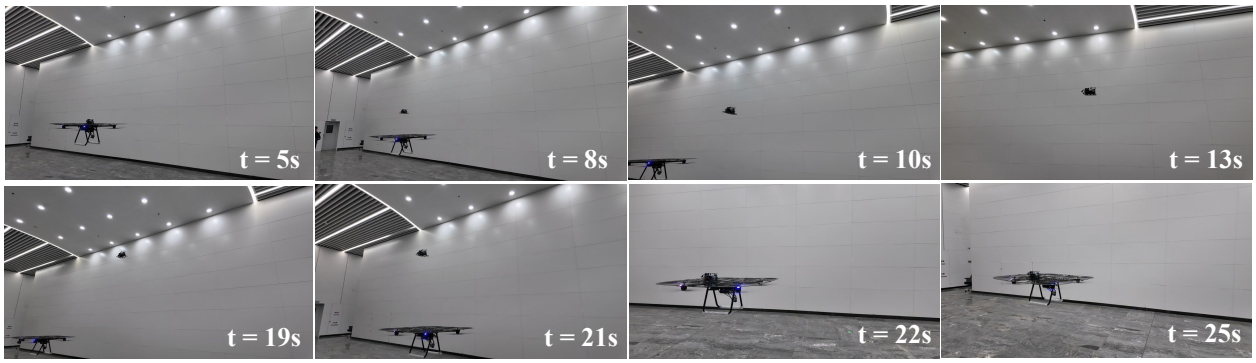

Fig. 11. The real-world indoor experimental process, conducted in a 10m x 20m x 6m testing site at Guangzhou, China. The sequence shows the complete autonomous mission: the AVC transports the MAV to a designated location, the MAV takes off, executes its trajectory, and performs a precise landing back onto the AVC's deck.

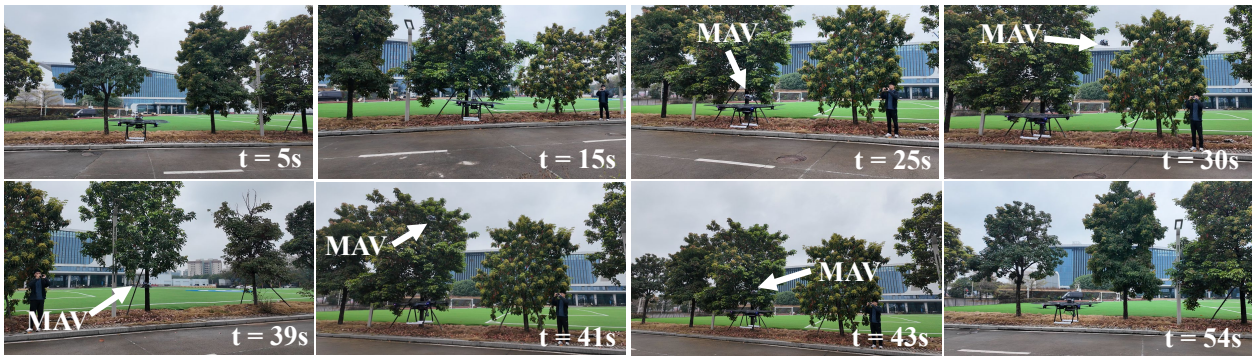

Fig. 12. The real-world outdoor experimental process, conducted in a 50m x 50m open field at a testing site in Guangzhou, China. The sequence demonstrates the complete autonomous mission: the AVC transports the MAV to the task area, the MAV takes off, executes its trajectory, and lands precisely back on the AVC's deck.

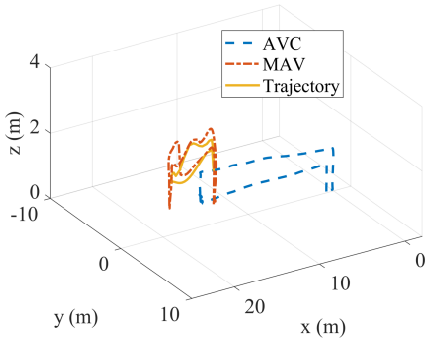

Fig. 13. Outdoor field test trajectory.

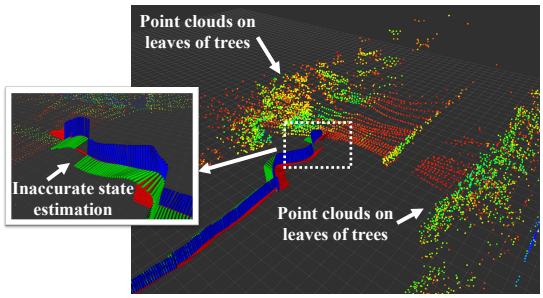

Fig. 14. Outdoor field test point clouds and the odometry of the MAV.

a detailed discussion of perception algorithms considered beyond the scope of this study.

V. Conclusions

This paper has introduced the first fully autonomous AVC system designed to operate with MAVs in real-world environments. Unlike previous efforts heavily dependent on external motion capture, the system presented here can be deployed rapidly and reliably with minimal reliance on external infrastructure. A novel trajectory optimization algorithm based on SCP was developed to ensure the safe and efficient operation of AVC-MAV collaboration, incorporating comprehensive safety constraints such as flight corridors and landing cones.

Extensive tests demonstrated that the AVC system successfully collaborates with the MAV under both internal aerodynamic effects and real-world environmental disturbances. Indoor experiments verified the synergy between the AVC and MAV in a controlled setting, while outdoor deployments validated the adaptability and resilience of the autonomous system in more challenging scenarios without specialized positioning systems. Future efforts will address the development of more advanced perception algorithms, particularly for outdoor environments subject to reduced sensor data quality and wind-induced turbulence, ensuring continued

reliability and scalability of the AVC–MAV collaboration in diverse application domains.

REFERENCES

[1] H. Huang and A. V. Savkin, "Deployment of charging stations for drone delivery assisted by public transportation vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 043–15 054, 2022.

[2] H. Huang, A. V. Savkin, and C. Huang, "Decentralized autonomous navigation of a UAV network for road traffic monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.

[3] 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.

[4] Z. Shen, G. Zhou, H. Huang, C. Huang, Y. Wang, and F.-Y. Wang, "Convex optimization-based trajectory planning for quadrotors landing on aerial vehicle carriers," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 138–150, 2024.

[5] 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.

[6] I. Domínguez, R. Miranda-Colorado, L. T. Aguilar, and D. A. Mercado-Ravell, "A methodology for setting-up a low-cost quadrotor experimental platform," *Control Engineering Practice*, vol. 143, p. 105803, 2024.

[7] E. Smeur, G. de Croon, and Q. Chu, "Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection," *Control Engineering Practice*, vol. 73, pp. 79–90, 2018.

[8] Y. Li, X. Liu, P. Lu, Q. He, R. Ming, and W. Zhang, "Angular acceleration estimation-based incremental nonlinear dynamic inversion for robust flight control," *Control Engineering Practice*, vol. 117, p. 104938, 2021.

[9] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.

[10] M. Bolognini, L. Fagiano, and M. P. Limongelli, "A fault-tolerant automatic mission planner for a fleet of aerial vehicles," *Control Engineering Practice*, vol. 135, p. 105501, 2023.

[11] V. Marguet, C. K. Dinh, F. Stoican, and I. Prodan, "Indoor formation motion planning using B-splines parametrization and evolutionary optimization," *Control Engineering Practice*, vol. 152, p. 106066, 2024.

[12] D. Six, S. Briot, A. Chriette, and P. Martinet, "Dynamic modelling and control of flying parallel robots," *Control Engineering Practice*, vol. 117, p. 104953, 2021.

[13] P. Mao, R. Fu, and Q. Quan, "Optimal virtual tube planning and control for swarm robotics," *The International Journal of Robotics Research*, vol. 43, no. 5, pp. 602–627, Apr. 2024.

[14] S. Güler and İ. E. Yıldırım, "A distributed relative localization approach for air-ground robot formations with onboard sensing," *Control Engineering Practice*, vol. 135, p. 105492, 2023.

[15] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-UAV system," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1816–1835, 2023.

[16] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5g advanced and beyond," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 347–359, 2024.

[17] X. Dong, H. Li, Y. Cui, J. Xiang, D. Li, and Z. Tu, "Aerial landing of micro UAVs on moving platforms considering aerodynamic interference," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 089–10 096, 2024.

[18] 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.

[19] 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.

[20] T. P. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Açıkmeşe, and J. M. Carson, "Dual quaternion-based powered descent guidance with state-triggered constraints," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 9, pp. 1584–1599, 2020.

[21] 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.

[22] K. Oguri, "Successive convexification with feasibility guarantee via augmented lagrangian for non-convex optimal control problems," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 3296–3302.

[23] K. Echigo, O. Sheridan, S. Buckner, and B. Açıkmeşe, "Dispersion sensitive optimal control: A conditional value-at-risk-based tail flattening approach via sequential convex programming," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 6, pp. 2468–2475, 2024.

[24] J. Li, J. Sugihara, and M. Zhao, "Servo integrated nonlinear model predictive control for overactuated tiltable-quadrotors," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8770–8777, 2024.

[25] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, 2023.

[26] J. Arrizabalaga and M. Ryll, "Sctomp: Spatially constrained time-optimal motion planning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 4827–4834.

[27] 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.

[28] P. J. Goulart and Y. Chen, "Clarabel: An interior-point solver for conic programs with quadratic objectives," 2024.

[29] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based Python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 1–6.