

Robust Trajectory Planning for Spatial-Temporal Multi-Drone Coordination in Large Scenes

Zhepei Wang, Chao Xu, and Fei Gao

Abstract—In this paper, we describe a robust multi-drone planning framework for high-speed trajectories in large scenes. It uses a free-space-oriented map to free the optimization from cumbersome environment data. A capsule-like safety constraint is designed to avoid reciprocal collisions when vehicles deviate from their nominal flight progress under disturbance. We further show the minimum-singularity differential flatness of our drone dynamics with nonlinear drag effects involved. Leveraging the flatness map, trajectory optimization is efficiently conducted on the flat outputs while still subject to physical limits considering drag forces at high speeds. The robustness and effectiveness of our framework are both validated in large-scale simulations. It can compute collision-free trajectories satisfying high-fidelity vehicle constraints for hundreds of drones in a few minutes.

I. INTRODUCTION

Multi-drone coordination is receiving increasing attention as a fundamental problem in various applications such as urban delivery, exploration, and inspection. It often requires robust planning for concurrent long-distance flights in vast spaces. The coordination should also be tolerant of realistic factors from both the environment and vehicles. The high problem dimension and the huge environment data further prevent existing algorithms from being applicable to large scenes. Most of them consider over-simplified safety criteria and system dynamics in relatively short-range flights.

Several practical problems exist in multi-drone planning. Firstly, planning algorithms frequently need accessing map data for obstacle information [1]. In large scenes, an obstacle-oriented map can become quite cumbersome, making itself a computational bottleneck. Secondly, robust planning for multi-drone should exploit the flexibility in both space and time aspects while the latter is often ignored in the literature. Thirdly, unexpected disturbances can make vehicles deviate a lot from nominal trajectories. In this case, potential reciprocal collisions pose threats to the whole system. Therefore, mismatches between actual flight progress and the planned one easily invalidate distance-based safety criteria ensured in the planning phase. Fourthly, high-speed flights admittedly improve task efficiency in large scenes. However, aerodynamic drag effects cannot be ignored when ensuring the physical limits of vehicles. These also make oversimplified feasibility criteria insufficient here.

In this paper, we propose a robust framework for multi-drone planning in large scenes. The framework is built upon

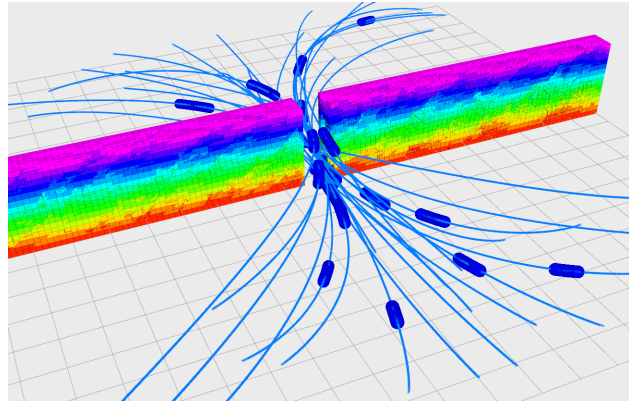


Fig. 1. These are 20 drones flying through a narrow gap in opposite directions concurrently. The blue “capsules” indicate the space-time uncertainty of vehicles. Each vehicle can deviate from the timestamp and position of its nominal trajectory. Our robustness is guaranteed by collision avoidance between capsules via spatial-temporal planning.

three criteria for flight coordination. An obstacle avoidance criterion ensures that no collision occurs between vehicles and environments. We use a free-space-oriented map instead where a union of polyhedra tightly approximates all free configurations. The MINCO [2] trajectory is then adopted for online spatial-temporal optimization within polyhedron-shaped corridors. We also design a reciprocal safety criterion via a space-time “capsule” constraint. It makes large flight errors tolerable in time and position. Therefore, our planning results are robust against reasonable perturbations. To ensure a dynamic feasibility criterion, we show the differential flatness for our drones subject to nonlinear drags. The flatness map makes it possible to enforce user-defined physical limits via penalty functionals supported by MINCO.

Summarizing our contributions in this work:

- A map polyhedronization scheme with corridor generation is proposed for online free space query;
- A space-time capsule constraint is designed for robust reciprocal safety against large flight errors;
- Minimum-singularity differential flatness is shown for our drones subject to nonlinear drag effects;
- A systematic way to robust trajectory planning is provided for multi-drone coordination with physical limits.

II. RELATED WORK

Planning for multi-drone coordination is inherently a high-dimensional problem even if safety is the only requirement. To reduce difficulty, reciprocal collision avoidance [3] computes the feasible velocity in a decentralized way. Conflict-based search [4] conducts centralized graph search instead,

with complexity dominated by the number of conflicts rather than agents. Their nonsmooth results do not suit high-order dynamics. Some methods formulate the joint planning into a Mixed Integer Quadratic Program (MIQP) [5] or a Sequential Convex Program (SCP) [6] to generate smooth trajectories in a centralized way, while they are not easily scaled to a large vehicle number. In [7], separating hyperplanes are applied to convex hulls of trajectories, forming a decoupled QP for each individual in every refinement iteration. A drawback is that obstacle-free environments are assumed. In [8], separating planes also form the safe flight corridor for a vehicle, which excludes all other dynamic and static obstacles. The planning is done by iterative refinement based on guaranteed safe results from [4]. To reduce conservativeness, these hyperplanes are treated as decision variables of a Nonlinear Program (NLP) for planning [9]. However, all these methods do not consider real physical limits and temporal optimization. To assure higher-fidelity dynamics, the learned aerodynamic interactions are incorporated into full quadrotor dynamics [10], showing the high stability in close-proximity coordination flights. Optimization-based temporal scheduling is performed in [11] while it incorporates integer variables and assumes obstacle-free environments. Different from existing work, our work accomplishes spatial-temporal coordination for multi-drone in an incremental way. It further handles state-input limits and nonlinear drag effects during high-speed flights.

III. PRELIMINARIES

A. Map Polyhedronization

Safe flight corridors are convenient to encode free-space information into trajectory planning while conquering non-smoothness in discrete environment data. Instead of online construction, we propose map polyhedronization as a pre-processing step for corridor-based trajectory planning. The process is done in advance for any large-scale fixed scene, such as occupancy grids, point clouds, or triangle meshes of digital maps.

Let \mathcal{F} denote the free configuration in a given map. By map polyhedronization we mean finding a set of polyhedra $\mathcal{P}_i \subseteq \mathcal{F}$ such that their union approximates \mathcal{F} at a satisfactory filling rate, i.e.,

$$\text{vol} \left(\bigcup_{i=1}^{M_P} \mathcal{P}_i \right) > (1 - \epsilon) \text{vol}(\mathcal{F}), \quad (1)$$

where $\text{vol}(\cdot)$ means the volume and

$$\mathcal{P}_i = \left\{ x \in \mathbb{R}^3 \mid A_i x \preceq b_i \right\}. \quad (2)$$

There are many algorithms [12]–[15] able to generate a free polyhedron wrapping a given seed. To accomplish the map polyhedronization, we first select a random position $x \in \mathcal{F} \setminus \bigcup_{i=1}^j \mathcal{P}_i$ by rejection sampling. Any algorithm mentioned is applied to generate a \mathcal{P}_{j+1} . This procedure repeats for $j \leftarrow j + 1$ until its rejection rate exceeds $1 - \epsilon$. Two examples are also provided in Figure 2. Axis-aligned bounding boxes of these polyhedra are managed by a multi-level segment tree for three-dimensional stabbing queries [16]. Consequently,

for any position x in the covered free space, it becomes efficient to obtain an outer polyhedron, denoted by $\mathcal{P}(x)$.

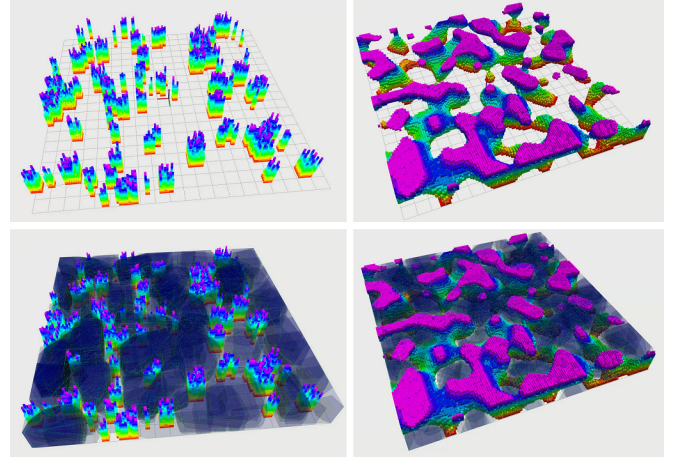


Fig. 2. The top figures show two kinds of obstacle environments. The bottom figures show polyhedronization of these maps. All free spaces are tightly filled by unions of convex polyhedra, colored in dark blue.

B. Problem Statement

Consider a set of drones conducting concurrent flights in one scene \mathcal{F} . The i -th vehicle's position is $r_i(t) : \mathbb{R} \mapsto \mathbb{R}^3$ at an absolute timestamp t . A new flight mission from p_o to p_f is to start at t_o , while its trajectory $r(t)$ should not conflict with early existed missions. Therefore, it must follow some criteria for obstacle avoidance, reciprocal safety, and dynamic feasibility.

The obstacle avoidance criterion requires the vehicle to fly in the covered free configuration $\tilde{\mathcal{F}} = \bigcup_{i=1}^{M_P} \mathcal{P}_i$, i.e.,

$$r(t) \in \tilde{\mathcal{F}}, \quad \forall t \in [t_o, t_f], \quad (3)$$

where $T_\Sigma \in \mathbb{R}_{>0}$ is the total duration and $t_f = t_o + T_\Sigma$. The reciprocal safety criterion requires the robustness against unexpected disturbance. Thus we consider the space-time capsule constraint, ensuring safety margins in space and time aspects are both considered. Specifically,

$$\|r(\alpha) - r_i(\beta)\|_W \geq 2M_r, \quad (4a)$$

$$\forall \alpha \in [t - M_d, t + M_d] \cap [t_o, t_f], \quad (4b)$$

$$\forall \beta \in [t - M_d, t + M_d], \quad (4c)$$

$$\forall t \in [t_o, t_f], \quad (4d)$$

where $W = \text{diag}\{1, 1, w\}$ enlarges the margin vertically to prevent downwash interference if $w < 1$. The scalars M_r and M_d are spatial and temporal margins for individuals, respectively. The dynamic feasibility criterion requires vehicle states and inputs such as the collective thrust and body rate to fulfill physical limits when the drone tracks $r(t)$ even with significant air drag at a high speed.

IV. METHOD

A. Just-In-Time Corridor Generation

Safe flight corridors are essentially sequences of polyhedra. Corridors ensure collision-free flights by excluding static obstacles while keeping parameters as compact as possible.

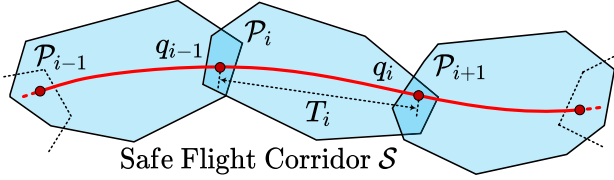


Fig. 3. Every waypoint q_i is sequentially assigned in to the intersection of consecutive polyhedra $\mathcal{P}_i \cap \mathcal{P}_{i+1}$. This assignment forms a convex constraint for each piece, instead of the nonconvex constraint \mathcal{S} . The trajectory is only parameterized by waypoints and durations T_i .

Thus, it is well-suited for large-scale maps with redundant data for navigation. We conduct just-in-time corridor generation for a newly occurred flight mission based on the interface $\mathcal{P}(x)$ of a previously polyhedronized map.

Firstly, we apply the Informed RRT* [17] to obtain an approximately shortest path from p_o to p_f in $\tilde{\mathcal{F}}$. Our main concern here is that in large-scale scenes, energy consumption is more relevant to the distance covered. Denote by $\bar{r}(l) : [0, L] \mapsto \mathbb{R}^3$ a arc-length-parameterized path. Based on the collision-free path, the generation of a homotopic safe flight corridor \mathcal{S} is given by the algorithm below, also visualized in Figure 6.

Algorithm 1: Homotopic Corridor Generation

Input: Path $\bar{r}(l)$ and Polyhedronized Map $\mathcal{P}(x)$

Output: Safe Flight Corridor \mathcal{S}

begin

$\mathcal{S} \leftarrow \{\}, l \leftarrow 0;$
while $l \leq L$ **do**
 $x \leftarrow \bar{r}(l);$
 $\bar{\mathcal{P}} \leftarrow \mathcal{P}(x);$
 $\mathcal{S}.\text{append}(\bar{\mathcal{P}});$
 $l \leftarrow \max_{\theta \in [l, L]} \theta, \text{ s.t. } \bar{r}(\theta) \in \bar{\mathcal{P}}, \forall \theta \in [l, \theta];$

return $\mathcal{S};$

Algorithm 1 actually finds a convex cover of the guiding path as soon as a new flight mission occurs. All polyhedra have been precomputed and can be online accessed directly. Thus the corridor construction is just-in-time and efficient. \mathcal{S} provides a large room for consequent trajectory planning subject to various constraints. For convenience, we denote \mathcal{S} by a polyhedron sequence $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_M\}$ hereafter.

B. Spatial-Temporal Planning in Flight Corridors

For efficiency, we conduct optimization in the flat-output space of multicopters such that all differential constraints from dynamics are fulfilled by default. We adopt the MINCO representation [2] to conduct spatial-temporal deformation of the flat-output trajectory.

An s -order MINCO trajectory is indeed a $2s$ -order polynomial spline with constant boundary conditions. It provides a linear-complexity smooth map from intermediate points q and a time allocation T to the coefficients of splines, which is denoted as $\mathcal{M}(q, T) : \mathbb{R}^{3 \times (M-1)} \times \mathbb{R}_{>0}^M \mapsto \mathbb{R}^{2Ms \times 3}$. A spline with $c = \mathcal{M}(q, T)$ is exactly the unique control effort

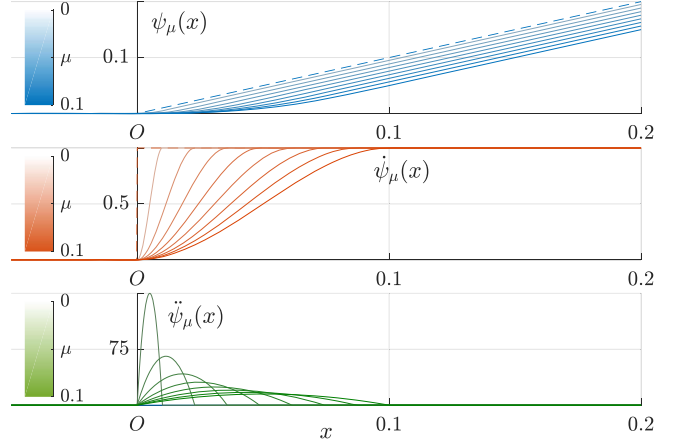


Fig. 4. The function ψ_μ has continuous first and second derivatives for $\mu > 0$. As μ approaches 0, the function also approaches the exact penalty $\max(x, 0)$. By iteratively shrinking μ , constraints are enforced within any desired precision while keeping a bounded weight and the C^2 -smoothness.

minimizer of an s -integrator that passes q . Moreover, given with any function $\mathcal{K}(c, T)$, MINCO can also serve as a linear-complexity differentiable layer $\mathcal{W}(q, T) := \mathcal{K}(\mathcal{M}(q, T), T)$, such that $\partial \mathcal{W} / \partial q$ and $\partial \mathcal{W} / \partial T$ can be efficiently computed from any $\partial \mathcal{K} / \partial c$ and $\partial \mathcal{K} / \partial T$.

The corridor \mathcal{S} provides natural constraints for the obstacle avoidance criterion in (3). We sequentially assign trajectory pieces into \mathcal{S} as is shown in Figure 3. For the case where a polyhedron \mathcal{P}_i has only one piece, we have:

$$q_i \in \mathcal{P}_i \cap \mathcal{P}_{i+1}, \forall 1 \leq i \leq M, \quad (5)$$

where q_i is the i -th column in q . Unconstrained coordinates $q(\xi)$ and $T(\tau)$ are adopted such that (5) and the positiveness of time are both satisfied by default [2]. The continuous-time safety is enforced via the penalty functional below.

$$\mathcal{I}_1(c, T) = \sum_{i=1}^M \int_0^{T_i} \mathbf{1}^T \phi_\mu [A_i c_i^T \beta(t) - b_i] dt, \quad (6)$$

where $\beta(t) = (1, t, t^2, \dots, t^{2s-1})^T$ is the power basis, $\phi_\mu[\cdot]$ an entry-wise operator for ϕ_μ , and $\phi_\mu : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$ an C^2 -smoothing of the exact penalty. ϕ_μ is defined as

$$\phi_\mu(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ (\mu - x/2)(x/\mu)^3 & \text{if } 0 < x < \mu, \\ x - \mu/2 & \text{if } x \geq \mu. \end{cases} \quad (7)$$

Figure (4) shows that function ϕ_μ approximates $\max(x, 0)$ as μ approaches 0, thus a finite weight for penalty can enforce the constraint at any specified precision.

The obstacle avoidance has been ensured by (5) and (6). Now that they are defined via either $\{q, T\}$ or $\{c, T\}$, we can utilize the property of MINCO to formulate them on unconstrained coordinates ξ and τ . Note that there is no need for heuristic time allocation as done in traditional corridor-based methods. Optimization of decoupled time parameters are directly supported by MINCO.

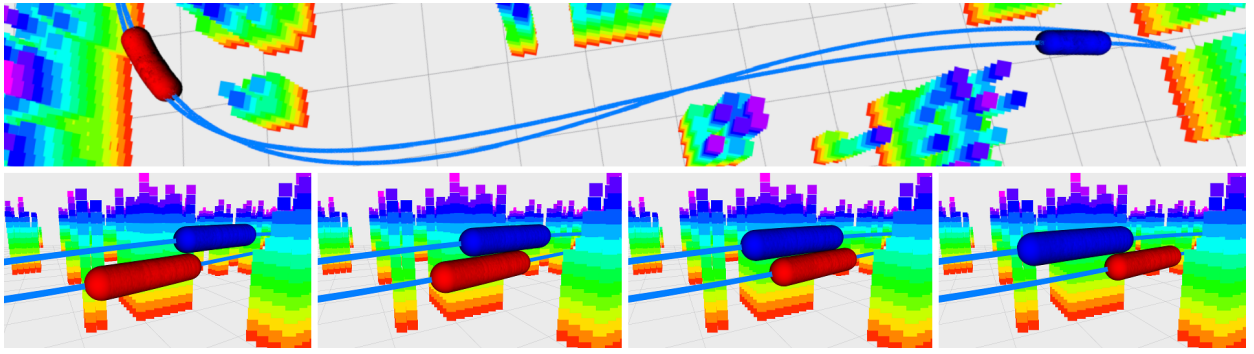


Fig. 5. The top figure shows two vehicles fly in the opposite direction. The bottom figures show snapshots of their meeting. The red and blue capsules represent their spatial and temporal uncertainties. A robust planner should guarantee the condition (4), i.e., no intersection occurs between them. In these figures, all trajectories are generated using the equivalent condition (8) instead.

C. Robust Reciprocal Safety via Space-Time Capsules

The reciprocal safety criterion (4) ensures robustness in accidents that vehicles may fail to accurate trajectory tracking under wind disturbance or so on. Large errors can occur relative to the nominal position-stamp tuple. In this case, only considering the safe distance margin is insufficient because high-speed flights are common in large scenes.

The condition (4) views vehicles as space-time capsules instead. Representing all uncertainties of a stamped position, a capsule is a sphere-swept volume of the trajectory segment on $[t - M_d, t + M_d]$ with sphere radius M_r . Robust reciprocal safety requires that all pairs of capsules be collision-free at any t . This condition is difficult to enforce since two time-varying nonconvex volumes are required to be collision-free all the time. Fortunately, an equivalent but more convenient condition exists:

$$\|r(t) - r_i(\gamma)\|_W \geq 2M_r, \quad (8a)$$

$$\forall \gamma \in [t - 2M_d, t + 2M_d], \quad (8b)$$

$$\forall t \in [t_o, t_f]. \quad (8c)$$

Here we give a proof of the equivalence between (4) and (8). For a given $r_i(t)$, if (4) holds while there is a $\zeta \in [t_o, t_f]$ and a $\gamma \in [\zeta - 2M_d, \zeta + 2M_d]$ such that $\|r(\zeta) - r_i(\gamma)\|_W < 2M_r$, then letting $\alpha = \zeta$, $\beta = \gamma$, and $t = (\zeta + \gamma)/2$ in (4) gives a contradiction. If (8) holds while there is an $\alpha \in [\zeta - M_d, \zeta + M_d] \cap [t_o, t_f]$ and a $\beta \in [\zeta - M_d, \zeta + M_d]$ such that $\|r(\alpha) - r_i(\beta)\|_W < 2M_r$, then letting $t = \alpha$ and $\gamma = \beta$ in (8) gives a contradiction.

It now becomes tractable to incorporate the robust reciprocal safety into our planning problem. Because (8) only forbids collisions between a single volume and a dimensionless point at each timestamp. We enforce the space-time capsule constraint via the following penalty functional,

$$\mathcal{I}_2(c, T) = \sum_{i=1}^N \int_{t_o}^{t_f} \int_{-2M_d}^{2M_d} \phi_\mu \left(4M_r^2 - \|r(t) - r_i(t+v)\|_W^2 \right) dv dt, \quad (9)$$

where t_f and $r(\cdot)$ are only determined by $\{c, T\}$. A planning result is given in Figure 5, showing the effectiveness of (9).

D. Physical Limits on Vehicle Dynamics with Drag Effects

High-speed flights in large-scale scenes put forward higher requirements on dynamic feasibility. Unlike existing flatness-based methods using oversimplified dynamics, we enforce physical limits under nonlinear drag effects while still conducting optimization in the flat-output space.

Consider the vehicle state $x = \{r, \dot{r}, R\}$ where $r \in \mathbb{R}^3$ and $R \in \text{SO}(3)$ are its translation and rotation, respectively. The input is $u = \{f, \omega\}$ where $f \in \mathbb{R}_{\geq 0}$ is the thrust and $\omega \in \mathbb{R}^3$ the body rates. The vehicle dynamics are defined as

$$\begin{cases} m\ddot{r} = -mge_3 - RDR^T\sigma(\|\dot{r}\|)\dot{r} + Rfe_3, \\ \dot{R} = R\hat{\omega}, \end{cases} \quad (10a)$$

$$(10b)$$

where m is the vehicle mass, g the gravitational acceleration, $e_3 = (0, 0, 1)^T$, $D = \text{diag}\{d_h, d_h, d_v\}$ a horizontally symmetric drag coefficient matrix, $\sigma: \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ a nonlinear term, and $\hat{\omega}$ a skew-symmetric matrix. Note we assume the vehicle to be horizontally symmetric. This decouples the yaw heading from drag effects and is common for multicopters as assumed in [18] and [19]. According to the analysis in [20], we adopt $\sigma(x) = 1 + C_p x$ in our lumped parameter model to incorporate both the linear drag [21] and the parasitic drag.

The physical limits $\mathcal{G}(x, u) \leq 0$ for (10) are defined as

$$\mathcal{G}(x, u) = \begin{pmatrix} \|\dot{r}\|^2 - v_{max}^2 \\ \|\omega\|^2 - \omega_{max}^2 \\ \arccos(e_3^T R e_3) - \theta_{max} \\ (f - f_m)^2 - f_r^2 \end{pmatrix} \leq 0, \quad (11)$$

where $f_m = (f_{max} + f_{min})/2$ and $f_r = (f_{max} - f_{min})/2$ are intermediate constants. Maximum flight speed, body rate, tilt angle, and thrust are specified by v_{max} , ω_{max} , θ_{max} , and f_{max} , respectively. Besides, restricting the tilt angle prevents excessively aggressive maneuvers. Lower bounding the thrust by f_{min} benefits attitude stabilization under disturbance.

The basic idea to incorporate physical limits on x and u into a flat trajectory is to utilize the algebraic transformation of differential flatness together with its differentiation. We denote by $r^{[s]}$ the stack of finite derivatives $(r, \dot{r}, \dots, r^{(s)})$, and by ψ the yaw. The flatness transformation is given by

$$(x, u) = \Psi(r^{[s]}, \psi^{[s]}). \quad (12)$$

All physical limits are enforced by the penalty functional,

$$\mathcal{I}_3(c, T) = \int_{t_o}^{t_f} \mathbf{1}^T \phi_\mu \left[\mathcal{G} \circ \Psi \left(r^{[s]}(t), \psi^{[s]}(t) \right) \right] dt, \quad (13)$$

where $\psi(\cdot)$ is any given planning of yaw. An energy functional is also incorporated to ensure a smooth flight,

$$\mathcal{I}_0(c, T) = \int_{t_o}^{t_f} \left(\|r^{(s)}(t)\|_2^2 + \rho \right) dt, \quad (14)$$

where both t_f and $r(\cdot)$ are still determined by $\{c, T\}$.

Now we give all details about the algebraic function Ψ for the concerned vehicle dynamics (10). We left multiply (10a) by body axes $x_b = Re_1$ and $y_b = Re_2$, then

$$(Re_i)^T (\ddot{r} + \frac{d_h}{m} \sigma(\|\dot{r}\|) \dot{r} + ge_3) = 0, \quad \forall i \in \{1, 2\}. \quad (15)$$

The consistency of $z_b = Re_3$ as d_h vanishes implies

$$z_b = \mathcal{N}(\ddot{r} + \frac{d_h}{m} \sigma(\|\dot{r}\|) \dot{r} + ge_3), \quad (16)$$

where $\mathcal{N}(x) = x/\|x\|_2$. Multiplying (10a) by z_b gives

$$f = z_b^T (m\ddot{r} + d_v \sigma(\|\dot{r}\|) \dot{r} + mge_3). \quad (17)$$

We use Hopf fibration [22] to decompose the yaw quaternion q_ψ and the tilt quaternion q_z without involving Euler angles,

$$q_\psi = (\cos(\psi/2), 0, 0, \sin(\psi/2))^T, \quad (18)$$

$$q_z = (1 + z_b(3), -z_b(2), z_b(1), 0)^T / \sqrt{2(1 + z_b(3))}. \quad (19)$$

Thus the rotation and body rates are given by

$$R = \mathcal{R}_{quat}(q_z \otimes q_\psi), \quad (20)$$

$$\omega = 2(q_z \otimes q_\psi)^{-1} \otimes (\dot{q}_z \otimes q_\psi + q_z \otimes \dot{q}_\psi), \quad (21)$$

where \otimes and the conversion $\mathcal{R}_{quat}(\cdot)$ are both given in [23], \dot{q}_z and \dot{q}_ψ are both evident from (18) and (19), respectively.

Algebraic procedures (16)-(21) exactly define the function Ψ in (12), whose gradient is as cheap as Ψ itself [24]. Also, we know that $s = 3$ is needed by the system (10) thus $r(t)$ should at least be jerk-controlled. There are only two intrinsic singularities in Ψ , which can be easily avoided by setting $\theta_{max} < \pi$ and $f_{min} > (d_v - d_h)\sigma(v_{max})v_{max}$. Our flatness map with nonlinear drag effects does not produce extra unnecessary singularities which occur in [21].

E. Incremental Planning for Multi-Drone Coordination

As for a newly occurred flight mission, we conduct the optimization below to accomplish trajectory planning subject to three concerned criteria in previous sections.

$$\min_{\xi, \tau} \sum_{\nu=0}^3 w_\nu \cdot \mathcal{I}_\nu(\mathcal{M}(q(\xi), T(\tau)), T(\tau)), \quad (22)$$

where $w_0 = 1$, w_1, w_2, w_3 are all finite weights for (7), variables $\{\xi, \tau\}$ are unconstrained coordinates for $\{q, T\}$ in a fixed corridor. This unconstrained NLP can be solved efficiently and reliably by quasi-Newton methods. Due to the fact that optimization usually focuses on high-quality local solutions, the initial values for points q and the time allocation T need to be further specified.

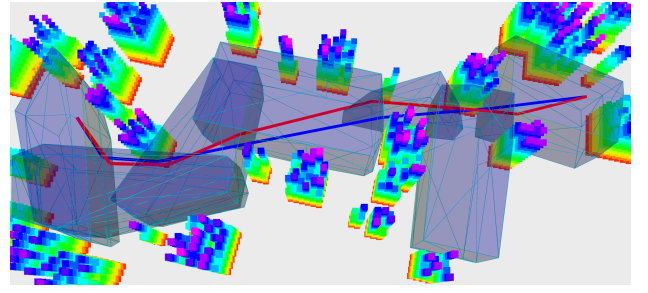


Fig. 6. The red path is generated by Informed RRT* [17]. The blue path is generated by distance-minimization (23) with $\sigma = 10^{-2}$. The blue one is a high-quality homotopic refinement computed within 0.025 ms.

As for q , we take the waypoints from a shortest path as its initial value. Denote $x_0 = p_o$ and $x_M = p_f$. The path is given by a lightweight distance minimization, i.e.,

$$\min_{\xi} \sum_{i=1}^M \sqrt{\|x_i - x_{i-1}\|_2^2 + \delta}, \quad (23a)$$

$$s.t. \quad x_i = q_i(\xi_i), \quad 1 \leq i < M. \quad (23b)$$

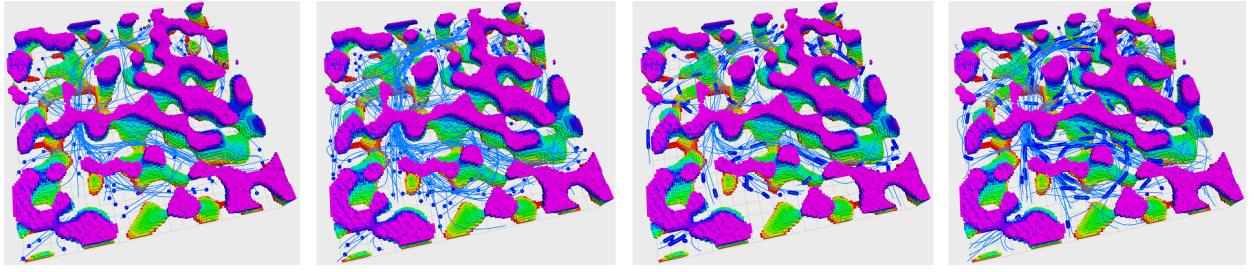
Here we use $\sqrt{\|\cdot\|_2^2 + \delta}$ as a δ -smooth approximation [25] to the Euclidean distance, where δ takes the desired precision. Note that coordinates ξ preserve local minima of the original convex problem for q . This minimization efficiently refines the rough solution as depicted in Figure 6.

As for T , we set the initial guess using trapezoidal velocity profiles [26] for the obtained q . A $p(t) : [0, \bar{T}] \mapsto \mathbb{R}^3$ is then generated by (22) with $w_2 = 0$, i.e., reciprocal safety is not considered. If $p(t)$ happens to be safe by checking (8), the planning is completed. Otherwise, temporal scheduling should be done for the length-parameterized curve of $p(t)$. We conduct kinodynamic RRT* [27] for a one-dimensional double integrator along the curve length, with limited speed and acceleration. Its cost and steering function are all determined by a 1-dimensional time-optimal curve whose closed-form solution is trivial for a double integrator. Note that the time scheduling never fails since a feasible solution always exists if the vehicle waits for a long enough time. The result provides a feasible time allocation T . Along with waypoints of $p(t)$, we finally obtain an initial feasible guess for (22).

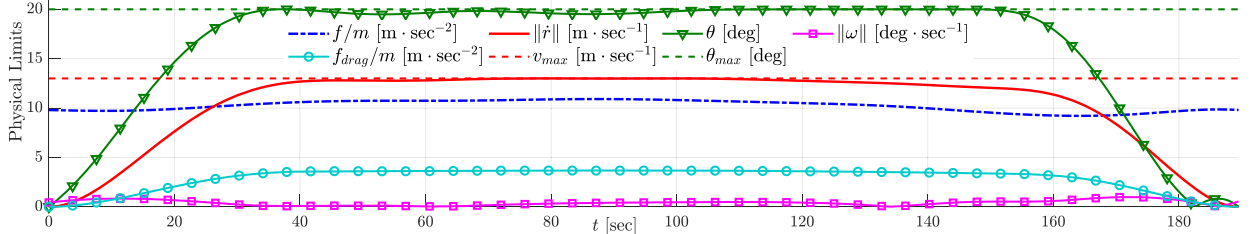
V. RESULTS

A. Implementation Details

We conduct multi-drone trajectory planning in an $2000 \times 2000 \times 200m^3$ space where obstacles are randomly generated by [28]. The occupancy map is firstly polyhedronized with $\epsilon = 10^{-5}$, implying the filling rate to be 99.999%. In safety criteria, we use $M_d = 4s$ and $M_r = 15m$. In vehicle dynamics, we use $m = 1.9kg$, $d_h = d_v = 0.475s^{-1}$, $C_p = 0.01m^{-1}s$. We set physical limits as $v_{max} = 13ms^{-1}$, $\omega_{max} = 2\pi/3rad s^{-1}$, $\theta_{max} = \pi/9rad$, $f_{min} = 9.5N$, and $f_{max} = 28.5N$. This implies the thrust-to-weight ratio to be only 1.53. Besides, $\mathcal{I}_1, \mathcal{I}_2$, and \mathcal{I}_3 are evaluated via equally-spaced quadrature with fixed node numbers. In optimization, we use $\rho = 10^{-3}$ for time regularization, $\mu = 10^{-2}$ as the initial smooth factor, and $w_1 = w_2 = w_3 = 10^5$ as weights,



(a) Multi-drone planning for 80 and 160 vehicles in a 4 km² dense-obstacle environment. Two left figures use distance-based safety constraints. Two right figures use space-time capsule constraints instead. All trajectories fulfill the physical limits under drag effects.



(b) The dynamic profile of a vehicle in the fourth figure above. Dashed lines indicate physical limits. The mass normalized thrust, speed, tilt angle, and the magnitude of body rate are all provided. The mass-normalized magnitude of the drag force is also given in the light-blue curve.

Fig. 7. Two subfigures give both the planning results in a large scene and the dynamic profile of a selected vehicle.

which ensure a high precision for constraints. Based on these settings, we plan for every newly-generated pair of start and goal in the map.

B. Multi-Drone Planning - Physical Limits

We enforce both the distance-based constraints and our space-time capsule constraints in (4). The results are shown in Figure 7. All trajectories are generated in several minutes. In Figure 7(a), our scheme ensures the safety of concurrent flights for up to 160 vehicles. Moreover, physical limits under drag effects are considered even if most vehicles have to cover about a kilometer. We give profiles of physical limits in Figure 7(b) for one of the vehicles, implying that all constraints (11) are satisfied during the entire 3 min flight.

According to Figure 7(b), our planner differs from traditional ones in that it does not assume a point-mass model. For example, some multi-drone delivery missions require a maximum tilt angle of fragile payloads. In this case, our planning directly meets these requirements as the actual tilt angle θ is always below θ_{max} . Moreover, it outperforms traditional ones that assume a drag-free rigid body. For example, whenever a vehicle flies as a large constant velocity, its acceleration becomes zero, and its thrust is exactly the weight if no drag is considered. Actually, the vehicle produces more thrust f and a nonzero tilt angle θ to cancel drag forces f_{drag} as shown in our figure. Therefore, multi-drone flights in large schemes indeed involve more realistic factors.

C. Multi-Drone Planning - Robustness

To validate the robustness of our scheme. We consider a case where 20 vehicles concurrently fly through a narrow gap, as shown in Figure 1. We use both the distance-based safety constraints and our capsule constraints to solve this planning. Moreover, we add disturbance to the vehicle such that tracking error occurs during their high-speed flight.

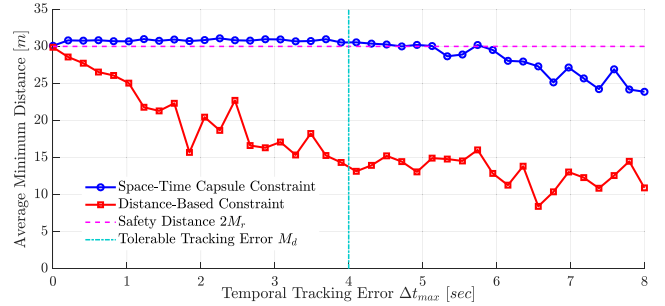


Fig. 8. Average minimum distances among all vehicle pairs for different safety criteria and temporal tracking error.

Denote by Δt_{max} the maximum temporal error for nominal trajectory tracking. For different Δt_{max} , we compute the minimum distance among all vehicle pairs. The entire multi-drone planning is repeated for 20 times. The average minimum distance is counted for both constraints in Figure 8. As Δt_{max} becomes positive, distance-based safety is quickly broken since the red curve goes below the safe distance $2M_r = 30$ m. Our capsule constraints guarantees the safety when $\Delta t_{max} \leq M_d$. Moreover, the safe distance is still maintained even if $4\text{ s} < \Delta t_{max} < 5\text{ s}$.

VI. CONCLUSION

In this paper, we propose a systematic scheme for robust multi-drone planning at high speeds. The free-space-oriented map much eases our planner from the burden of high-volume data accessing. The space-time capsule constraint ensures reciprocal safety even if any vehicle is significantly behind the predefined flight progress. The minimum-singularity flatness of our drone dynamics subject to nonlinear drags plays an essential role in ensuring realistic physical limits at high speeds. We believe this is a practical framework towards robust multi-drone trajectory planning in large scenes.

REFERENCES

- [1] J. Bialkowski, M. Otte, S. Karaman, and E. Frazzoli, "Efficient collision checking in sampling-based motion planning via safety certificates," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 767–796, 2016.
- [2] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *arXiv preprint arXiv:2103.00190*, 2021.
- [3] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *International Symposium of Robotic Research*, 2011, pp. 3–19.
- [4] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.
- [5] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *IEEE International Conference on Robotics and Automation*. Saint Paul, Minnesota, USA: IEEE, 2012, pp. 477–483.
- [6] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura, Algarve, Portugal: IEEE, 2012, pp. 1917–1922.
- [7] S. Tang and V. Kumar, "Safe and complete trajectory generation for robot teams with higher-order dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Daejeon, Korea: IEEE, 2016, pp. 1894–1901.
- [8] W. Hönig, J. A. Preiss, T. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.
- [9] J. Tordesillas and J. P. How, "MADER: Trajectory planner in multi-agent and dynamic environments," *IEEE Transactions on Robotics*, 2021.
- [10] G. Shi, W. Hönig, X. Shi, Y. Yue, and S.-J. Chung, "Neural-Swarm2: Planning and control of heterogeneous multirotor swarms using learned interactions," *IEEE Transactions on Robotics*, 2021.
- [11] W. Wu, F. Gao, L. Wang, B. Zhou, and S. Shen, "Temporal scheduling and optimization for multi-MAV planning," in *International Symposium of Robotic Research*, 2019.
- [12] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 109–124.
- [13] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robotics and Automation Letters*, pp. 1688–1695, 2017.
- [14] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-Repeat-Replan: A complete and robust system for aggressive flight in complex environments," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1526–1545, 2020.
- [15] X. Zhong, Y. Wu, D. Wang, Q. Wang, C. Xu, and F. Gao, "Generating large convex polytopes directly on point clouds," *arXiv preprint arXiv:2010.08744*, 2020.
- [16] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer, 2008.
- [17] 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.
- [18] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, "Nonlinear control of vtol uavs incorporating flapping dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 2419–2425.
- [19] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," in *IFAC World Congress*, vol. 50, no. 1. Elsevier, 2017, pp. 8189–8195.
- [20] M. Bangura, R. Mahony *et al.*, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Australasian Conference on Robotics and Automation*, Wellington, New Zealand, 2012.
- [21] 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.
- [22] M. Watterson and V. Kumar, "Control of quadrotors using the Hopf fibration on $SO(3)$," in *International Symposium on Robotics Research*, Hanoi, Vietnam, 2019.
- [23] J. Vince, *Quaternions for Computer Graphics*. Springer, 2011.
- [24] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.
- [25] A. Beck and M. Teboulle, "Smoothing and first order methods: A unified framework," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 557–580, 2012.
- [26] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.
- [27] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *IEEE Conference on Decision and Control*, Atlanta, Georgia, USA, 2010, pp. 7681–7687.
- [28] W. Wu, <https://github.com/HKUST-Aerial-Robotics/mockamap>.