2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Flexible and Topological Consistent Local Replanning for Multirotors

Dong Wang†, Hongkai Ye†, Neng Pan, Jinxin Huang, Bangyan Zhang,
Yinian Mao, Guoquan Huang, Chao Xu, and Fei Gao

***Abstract*— In many situations such as city delivery and wild inspection, quadrotors are often required to follow a predefined reference trajectory. However, these reference trajectories cannot be perfectly safe, resulting in conflicts between tracking the reference precisely, flying safely, and finishing the mission timely. This paper proposes to solve the above problem, by introducing a replanning framework that first generates a topological consistent collision-free initial path and then flexibly optimizes the rejoin point and trajectory duration to generate a smooth and safe local rejoining trajectory. To avoid local trajectory switching in different directions during high-frequency replanning, we propose a topology-preserving path search algorithm based on kinodynamic RRT*. To satisfy dynamic constraints, avoid delays, and achieve a smooth rejoin of the reference trajectory, we propose an optimization-based approach to refine the initial trajectory. The simulation results confirm that our proposed topological consistency and flexible optimization methods can reduce the risk of local trajectory and decrease obstacle avoidance delay for tracking reference trajectory. We also conduct real-world experiments in challenging environments and verify the effectiveness of our method.**

I. Introduction

In many situations such as city delivery and wild inspection, quadrotors are often required to operate following a predefined reference trajectory. In such scenarios, the global reference is necessary, since it reflects the mission requirement or human intentions. However, these reference trajectories cannot be perfectly safe, since new obstacles may change the prior map and unpredictable objects may interfere with the flight path. Therefore, tracking the reference precisely, flying safely, and finishing the mission timely may conflict with others.

Based on the above analysis, a proper local planner suits these situations is one that can avoid any possible obstacles threatening the flight and flexibly choose local targets to rejoin the global reference trajectory for continuing the mission. Safety is certainly of top priority, to avoid previously unknown obstacles on reference trajectory, the planner should always strictly respect collision avoidance constraints and operate at a high frequency. Moreover, since the quadrotor is guided by a reference path defined according to the application, the local trajectory must catch up with the global reference as soon as possible after avoiding obstacles. Finally, the planner should flexibly determine the rejoining schedule, to let the drone merge back to the reference trajectory smoothly and dynamically feasibly. Therefore, the local planner should have the ability to select a proper planning horizon and its associated terminal point based on quadrotor dynamics, motion status, and local environmental situation.

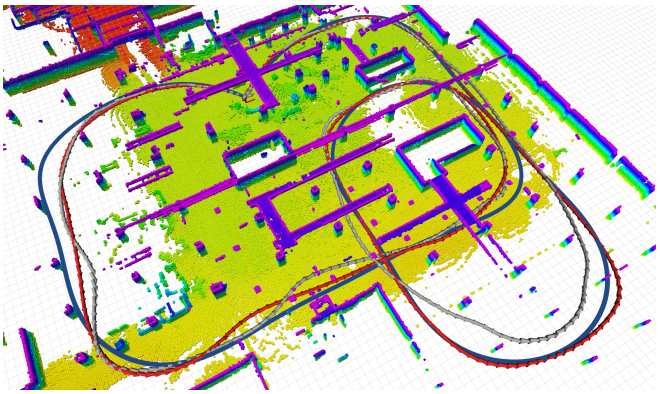


Fig. 1: Comparison of trajectory results between flexible end state and fixed end state. The blue trajectory is the global reference trajectory. The blue, red, and gray trajectories are global trajectory, flexible end-state trajectory, and fixed end-state trajectory respectively.

To satisfy the above-mentioned requirements, we design a two-step planning framework including a sample-based pathfinder for initializing a collision-free path and a trajectory optimizer for generating feasible motions with flexible planning horizons and rejoining states. In scenes where obstacles block the reference trajectory, environmental symmetry often occurs, and general path search algorithms cannot guarantee good consistency. During high-frequency replanning, the trajectory may switch between several local minimums under different topologies, causing the quadrotor to hesitate to avoid obstacles and increasing controller burden. To solve this problem, by increasing the replanning consistency, we propose a topology-preserving path search algorithm based on kinodynamic RRT*. After finding a collision-free path, we propose an optimization approach based on an efficient polynomial trajectory representation to refine it. To achieve smooth and precise access to the reference trajectory, the final state of the local trajectory should be exactly the same as the state of the global trajectory at the rejoin moment. In details, our approach optimizes

†These two authors contribute equally.

Dong Wang, Hongkai Ye, Neng Pan, Chao Xu, and Fei Gao are with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute, Zhejiang University, Huzhou 313000, China.

Jinxin Huang, Bangyan Zhang, Yinian Mao, and Guoquan Huang are with the Meituan UAV, Beijing, China.

This work was supported by the National Key R&D Program of China under grant no. 2023YFB4706600 and the National Natural Science Foundation of China under grant no. 62322314.

Email: (dongwangab, fgaoaa)@zju.edu.cn

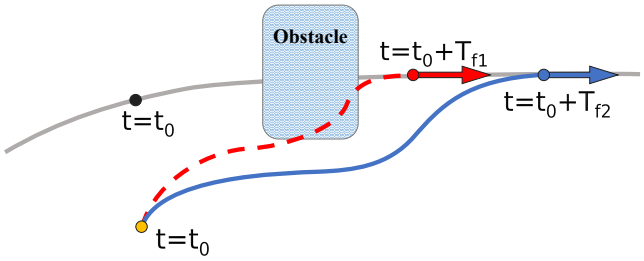


Fig. 2: Planning methods illustration. Our method lets the end state slide along the reference trajectory, and optimize for a reasonable and feasible trajectory duration T.

the trajectory duration and flexibly adjusts the end state. Intuitively, as shown in Fig.2, the terminal point slides along the reference trajectory during the optimization, which slides backward when longer obstacle avoidance time is required to satisfy dynamic constraints, and vice versa.

The contributions are summarized as:

1) We propose a topology-preserving path search algorithm based on kinodynamic RRT*, avoiding trajectory switches between several feasible solutions under different topologies.
2) We propose a flexible trajectory optimizer that generates dynamically feasible and smooth motions and strictly satisfies the rejoining reference trajectory constraints, by sliding the end state along the reference trajectory.
3) We implement our planning system in real-world challenging experiments to validate its effectiveness.

II. Related Works

A. Controller-based method

Considering avoiding obstacles while tracking trajectory is an important issue, some works [1]–[5] address this problem as a state-constrained control problem. Rodrguez-Seda et al. [1, 2] design specific expert control laws for following the reference trajectory and collision avoidance. This method guarantees stability and feasibility at a low computation burden. However, the specific assumptions are often violated in real-world situations. Fukuda et al. [3] propose a convex optimization-based controller using a relaxed control barrier function for trajectory tracking with obstacle avoidance of a two-wheel mobile robot. However, the controller output may violate dynamic constraints and only applies to 2D cases. Peng et al. [4] present a safe guidance scheme with the guarantee of critical obstacle avoidance for intercepting maneuvering targets, by describing the obstacle as high-order control barrier functions. Similarly, this method does not consider actuator constraints and is not adequate for more complex obstacle scenarios. Liu et al. [5] combine adaptive guidance laws with barrier Lyapunov function for addressing the path-following problem of an underwater vehicle in an environment with multiple static and moving obstacles. This method cannot support tracking time-indexed trajectories. In summary, these methods focus on designing controllers with strong stability and are difficult to apply to complex real environments.

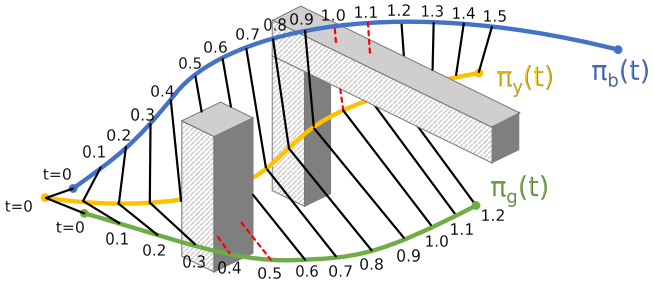


Fig. 3: τ-topological consistency of three trajectories. The starting positions are close and the ending positions are far apart. The yellow trajectory $\pi_y(t)$ can be regarded as the previous planning result and the blue trajectory $\pi_b(t)$ and green trajectory $\pi_g(t)$ are the possible trajectories of this planning. Sampling the positions with a resolution of 0.1 seconds, connecting the positions at corresponding times, and performing collision detection on the connection, we obtain that $\pi_y(t)$ and $\pi_g(t)$ are 0.3s-topological consistency, $\pi_y(t)$ and $\pi_b(t)$ are 0.9s-topological consistency.

B. Local replan trajectory method

More recent works [6]–[11] thus favor local replan a trajectory to adapt to more complex constraints and user-defined goals. Most of the time, selecting the planning endpoint depends on experience and can cause potential problems. For each replan cycle, Oleynikova et al. [6] and enbalar et al. [7] both fix the local target in the reference trajectory, either some certain distances or time durations away from the current state. Liu et al. [8] fix the tracking duration with a fuzzy model considering the current tracking error. These handcraft approaches do not address deviations and delays well, since they ignore the time index of the reference trajectory. Model Predictive Control (MPC) styled planners such as MPCC [9, 10] and MPFTC [11] introduce additional variables as fictitious time stamps, according to which the tracked setpoints slide along the reference trajectory, which can be seen as the reference trajectory waits for the robot, thus adaptively accommodating to the current deviations and lags. However, the standard MPC formulation forces the trajectory to be as close as possible to the reference. When the reference trajectory is unsafe, a risky trajectory that is too close to the obstacle will be generated. In Conclusion, existing methods are difficult to simultaneously handle the conflicts between tracking the reference precisely, flying safely, and finishing the mission timely.

III. Topology Preserving Path Search Algorithm

The concept of path homotopy equivalence has been proposed in many works [12]–[16], most of which are about finding paths belonging to different path topologies. For the replanning consistency problem explored in this section, we specifically propose the concept of τ-topological consistency. We first give the definition of trajectory τ-topological consistency, and then introduce the topology-preserving sampling-based kinodynamic motion planning method.

A. τ-Topological Consistency

Definition 3.1 (τ-topological consistency): Given an obstacle environment $\mathcal{E}$, and two time parameterized trajec-

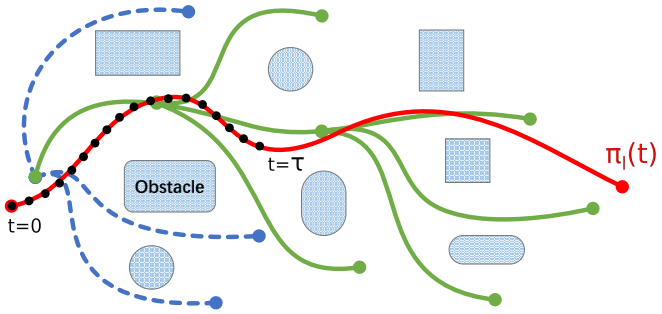


Fig. 4: τ-topological consistency growing tree. The red trajectory $\pi_l(t)$ is the previous result, given parameter τ, the growing tree $\mathcal{T}$ is shown by the green curve. $\mathcal{T}$ has the same direction with $\pi_l(t)$ when $t \in [0, \tau]$ and has no additional restriction on the growth of $\mathcal{T}$ when $t > \tau$. The blue dotted line cannot be added to $\mathcal{T}$ because it does not satisfy the τ-topological consistency with $\pi_l(t)$.

tories $\mathbf{p}_1(t), t \in [0, T_1]$ and $\mathbf{p}_2(t), t \in [0, T_2]$, where T_1, T_2 is the duration of trajectory. If $\|\mathbf{p}_1(0) - \mathbf{p}_2(0)\|_2 < \epsilon$ and line segment $\overline{\mathbf{p}_1(t)\mathbf{p}_2(t)}$ is collision free with $\mathcal{E}$, $\forall t \in [0, min(\tau, T_1, T_2)]$, then $\mathbf{p}_1(t)$ and $\mathbf{p}_2(t)$ is called τ-topological consistency, denoted as $\mathcal{TC}(\mathbf{p}_1(t), \mathbf{p}_2(t), \tau)$.

To detect whether the line connecting any two points on different trajectories collides with obstacles within a certain period, we adopt the approach of uniform visibility deformation similar to Zhou et al. [16, 17]. This method is a simplification of the visibility deformation proposed by Jaillet et al. [13], which greatly reduces the calculation while only introducing a small range of errors.

Different from most definitions of topological equivalence, the above definition does not require trajectories to have the same start and end positions. We require start positions to be close to each other and no requirement for end positions. Fig.3 shows the τ-topological consistency relationship of three trajectories.

Using the above definition, the consistency of replanning can be increased by requiring replan trajectory is τ-topological consistency with previous planning result. Intuitively, this replanning has the same obstacle avoidance direction with the last result within the time interval τ. When $\tau = 0$, each replanning is independent of the other, and the larger τ is, the stronger the consistency during replanning.

B. Topology Preserving Motion Planning Method

Considering the above definition, we propose a τ-topological consistency growing tree. The algorithm details are described in Alg.1. Given a trajectory $\pi_l(t)$, parameter τ and $\pi_l(0)$ is close to the plan start position, this method is used to plan a trajectory which is τ-topological consistent with $\pi_l(t)$.

The algorithm is based on kinodynamic RRT* [18] and with an extra constraint to limit the tree expansion within the specified range. When the tree $\mathcal{T}$ expands a new edge, in addition to collision detection, it also checks whether the new trajectory π_{new} is τ-topological consistent with $\pi_l(t)$, as described in line 11 of Alg.2 and line 6 of Alg.3. Fig.4 shows an example of this algorithm.

Algorithm 1 τ-Topological Consistency Growing Tree

Input: Enviromnent $\mathcal{E}$, Tree $\mathcal{T}$, State $\mathbf{x}$, parameter τ, Trajectory π_{last};
Output: Trajectory π^*;

$\mathcal{T} \leftarrow \{\mathbf{x}_{start}\}$, $FoundSolution \leftarrow False$
while Termination condition not met **do**
  $\mathbf{x}_{new} \leftarrow$ **Sampling**($\mathcal{E}$)
  $\mathcal{X}_{near} \leftarrow$ **NearNeighborSearch**($\mathcal{T}, \mathbf{x}_{new}, r$)
  if $\mathbf{x}_p \leftarrow$ **ChooseTCParent**($\mathcal{X}_{near}, \mathbf{x}_{new}, \mathcal{E}, \tau$) **then**
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{x}_{new}\}$
    if TryConnectGoal($\mathbf{x}_{new}, \mathbf{x}_{goal}$) **then**
      $FoundSolution \leftarrow True$
    end if
    if $FoundSolution$ **then**
      TCRewire($\mathcal{X}_{near}, \mathbf{x}_{new}$)
    end if
  end if
end while
$\pi^* \leftarrow$ **RetrieveTrajectory**($\mathcal{T}, \mathbf{x}_{goal}$)
return π^*

Algorithm 2 Choose τ-Topological Consistency Parent

$\mathbf{x}_p \leftarrow null$, $\mathcal{Q} \leftarrow \emptyset$
for $\mathbf{x}_i \in \mathcal{X}_{near}$ **do**
  $\hat{g}_{\mathbf{x}_i, \mathbf{x}_{new}} \leftarrow g_{\mathbf{x}_i} + \mathcal{J}(\mathbf{x}_i, \mathbf{x}_{new})$
  if $\hat{g}_{\mathbf{x}_i, \mathbf{x}_new} + H(\mathbf{x}_new, \mathbf{x}_goal) < g^*_{current}$ **then**
    $\mathcal{Q} \leftarrow^{+} (\mathbf{x}_i, \hat{g}_{\mathbf{x}_i, \mathbf{x}_new})$
  end if
end for
while $\mathcal{Q}$ is not empty **do**
  $\mathbf{x}_{min} \leftarrow \mathcal{Q}.top()$
  if CheckFeasible($\mathbf{x}_{min}, \mathbf{x}_{new}, \mathcal{E}$) **then**
    $\pi_{new} \leftarrow$ **RetrieveTrajectory**($\mathcal{T}, \mathbf{x}_{min}$) + ($\mathbf{x}_{min}, \mathbf{x}_{new}$)
    if $\mathcal{TC}(\pi_{last}, \pi_{new}, \tau)$ **then**
      $\mathbf{x}_p \leftarrow \mathbf{x}_{min}$
      RETURN $\mathbf{x}_p$
    end if
  end if
end while

Algorithm 3 τ-Topological Consistency Rewire

for $\mathbf{x}_i \in \mathcal{X}_{near}$ **do**
  $\hat{g}_{\mathbf{x}_{new}, \mathbf{x}_i} \leftarrow g_{\mathbf{x}_{new}} + \mathcal{J}(\mathbf{x}_{new}, \mathbf{x}_i)$
  if $\hat{g}_{\mathbf{x}_{new}, \mathbf{x}_i} < g_{\mathbf{x}_i} \wedge \hat{g}_{\mathbf{x}_{new}, \mathbf{x}_i} + H(\mathbf{x}_i, \mathbf{x}_{goal}) < g^*_{current}$ **then**
    if CheckFeasible($\mathbf{x}_{new}, \mathbf{x}_i, \mathcal{E}$) **then**
      $\pi_{new} \leftarrow$ **RetrieveTrajectory**($\mathcal{T}, \mathbf{x}_{new}$) + ($\mathbf{x}_{new}, \mathbf{x}_i$)
      if $\mathcal{TC}(\pi_{last}, \pi_{new}, \tau)$ **then**
        $\mathbf{x}_i.parent \leftarrow \mathbf{x}_{new}$
      end if
    end if
  end if
end for

IV. Flexible Trajectory Optimization Method

A. Preliminaries

Multirotors are typical differentially-flat systems, and their simplified linear dynamics models are widely utilized by the community [19]–[21] to reduce computing. We also build the dynamics as an s-order linear chain model,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= f\big(\mathbf{x}(t),\mathbf{u}(t)\big) = \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t),\\ \mathbf{u}(t) &= \boldsymbol{p}^{(s)}(t)^{\mathsf{T}},\ \mathbf{x}(t) = [\boldsymbol{p}(t),\cdots,\boldsymbol{p}^{(s-1)}(t)]^{\mathsf{T}},\end{aligned} \tag{1}$$

where $\boldsymbol{p}=[p_x,p_y,p_z]^{\mathsf{T}}\in\mathbb{R}^3$ is the position variable vector and $\cdot^{(s)}$ denotes its s-times derivative of time. With $s\geq 3$, the velocity $\boldsymbol{v}$, the rotation $\mathbf{R}$, the angular velocity $\boldsymbol{\omega}$, the thrust f, and other variables that describe the states of the multirotor can then be determined from the chosen flat states through a differential flatness translation,

$$\{\boldsymbol{p},\boldsymbol{v},\mathbf{R},\boldsymbol{\omega},f\} = \Psi(\mathbf{x},\mathbf{u}), \tag{2}$$

and the analytical translation of which can be found in our previous work [22]. For trajectory representation, polynomials prove to be effective [23], and are also the optimal solution form for unbounded Linear Quadratic Minimum Time (LQMT) problems [24]. We use piece-wise polynomials for our problem, and the parameters of an M-piece trajectory are denoted by $\{\mathbf{c},\mathbf{T}\}$, where $\mathbf{c}=(\mathbf{c}_1^{\mathsf{T}},\ldots,\mathbf{c}_M^{\mathsf{T}})^{\mathsf{T}}\in\mathbb{R}^{2Ms\times 3}$ and $\mathbf{T}=(T_1,\ldots,T_M)^{\mathsf{T}}\in\mathbb{R}_+^M$ are the coefficients and the time durations for each piece. For the reference trajectory, it is reasonable to assume that it is planned on the simplified states $\mathbf{x}$, and we thus denote it as $\mathbf{x}_{ref}(t)$.

B. Problem Formulation

For one replan cycle, we denote the planning start time as t_0, and the corresponding state in the reference trajectory is accordingly $\mathbf{x}_{ref}(t_0)$. The trajectory replanning in a cycle is formulated as an optimization problem as follows:

$$\min_{\{\mathbf{c},\ \mathbf{T}\}} \mathcal{J} = \int_0^{T_f}\frac{1}{2}\mathbf{u}(t)^{\mathsf{T}}\mathbf{u}(t)dt + \rho_r(T_f-T_{ref})^2, \tag{3a}$$

$$s.t.\quad \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t), \tag{3b}$$

$$\mathbf{x}(0)=\mathbf{x}_{init},\ \mathbf{x}(T_f)=\mathbf{x}_{ref}(t_0+T_f), \tag{3c}$$

$$\mathcal{G}\big(\mathbf{x}(t),\mathbf{u}(t),t\big)\preceq\mathbf{0},\ \forall t\in[0,T_f], \tag{3d}$$

where the inequality constraints are formulated as

$$\mathcal{G}\big(\mathbf{x}(t),\mathbf{u}(t),t\big) = \begin{pmatrix} \|\boldsymbol{v}(t)\|_2^2 - {\mathrm{v}_{\max}}^2 \\ \arccos\left(\mathbf{e}_3^{\mathsf{T}}\mathbf{R}\mathbf{e}_3\right)-\theta_{\max} \\ \|{}^b\boldsymbol{\omega}(t)\|_2^2 - {\Omega_{\max}}^2 \\ \big(f(t)-f_m\big)^2 - f_r^2 \\ r_{tol} - f_{\mathcal{DIST}}\big(\boldsymbol{p}(t)\big)\end{pmatrix}, \tag{4a}$$

$$f_m=(f_{max}+f_{min})/2,\ f_r=(f_{max}-f_{min})/2. \tag{4b}$$

Eq.3a is the objective that indicates minimizing the control, a typical formulation for multirotor energy-minimizing planning. We also demand the trajectory duration $T_f=\sum_{i=1}^M T_i$ to be close to a predefined expected trajectory duration T_{ref}, and the weight ρ_r controls the tightness of following. The decision variables are the polynomial coefficients and the trajectory duration $\{\mathbf{c},\mathbf{T}\}$.

Eq.3b is the system dynamics constraint described in Eq.1.

Eq.3c shows the boundary equality constraints. The initial state is obtained from the trajectory planned last cycle and set fixed. The terminal state is optimizable but constrained in the reference trajectory, uniquely determined by the trajectory duration, which is set equal to the trajectory duration T_f. The optimizable trajectory duration enables following the reference trajectory adaptively.

Eq.3d are the functional process inequality constraints to limit some physical status during the flight, and are detailed in Eq.4. We limit the velocity not to exceed the maximum allowed speed $\mathrm{v}_{\max}$, limit the tilting angle against the level not to exceed the maximum allowed tilting angle $\theta_{\max}$, limit the body rate not to exceed the maximum allowed angular velocity $\Omega_{\max}$, limit the thrust to be within the minimum thrust f_{min} and the maximum thrust f_{max} that the vehicle can produce, and limit the position to be in a desired minimum distance r_{tol} away from any obstacles. $f_{\mathcal{DIST}}(\cdot)\in\mathbb{R}$ is a function that returns the minimum distance to the nearest obstacles given a position point, obtained by maintaining a distance field.

C. Equality Constraints Elimination

To follow the equality constraints imposed by the dynamics and boundary conditions, MPC-styled discrete optimizing approaches typically use penalty methods, and the resulting solutions still do not exactly satisfy those equality constraints. We here use the MINCO trajectory class [23] to naturally eliminate the equality constraints.

Based on piece-wise polynomials, MINCO provides an analytical injection $\mathbf{c}=\mathcal{M}(\mathbf{q},\mathbf{T})$ that analytically solves the following minimum control problem,

$$\min_{\mathbf{c}}\mathcal{J}_{ctrl} = \int_0^{T_f=\sum_{i=1}^M T_i}\frac{1}{2}\mathbf{u}(t)^{\mathsf{T}}\mathbf{u}(t)dt, \tag{5a}$$

$$s.t.\quad \mathbf{u}(t)=\boldsymbol{p}^{(s)}(t),\ \forall t\in[0,T_f], \tag{5b}$$

$$\boldsymbol{p}^{[s-1]}(0)=\overline{\mathbf{p}}_o,\ \boldsymbol{p}^{[s-1]}(T_f)=\overline{\mathbf{p}}_f, \tag{5c}$$

$$\boldsymbol{p}(T_i)=\boldsymbol{p}_i,\ 1\leq i<M, \tag{5d}$$

where $\mathbf{q}=(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_{M-1})$ is composed of the middle waypoints in-between polynomial pieces, and $\overline{\mathbf{p}}_o$ and $\overline{\mathbf{p}}_f$ are the initial and terminal boundary conditions that take derivatives of the position from 0 to $s-1$ order, respectively. It means that by taking $\{\mathbf{q},\mathbf{T}\}$ as the decision variables, the optimized trajectory naturally satisfies the equality constraints Eq.3b and Eq.3c. More importantly, for any differentiable function $F(\mathbf{c},\mathbf{T}) = F(\mathcal{M}(\mathbf{q},\mathbf{T}),\mathbf{T}) = H(\mathbf{q},\mathbf{T})$, if $\partial F/\partial\mathbf{c}$ and $\partial F/\partial\mathbf{T}$ are provided, the derivatives $\partial H/\partial\mathbf{q}$ and $\partial H/\partial\mathbf{T}$ can be derived with MINCO.

D. Inequality Constraints Elimination

To deal with the functional process inequality constraints in Eq.3d, the idea from the method of Teo et al. [25] is adopted. We sample and evaluate the constraint functions at discrete time stamps and integrate the violated part, constructing a penalty function with a weighted sum of the sampled evaluated points. Considering that the trajectory

duration changes during optimization, we sample the constraint functions at a fixed time interval for each piece of the polynomial trajectory, such that a high accuracy of the penalty construction can be preserved.

The evaluation of each piece of the polynomial trajectory is independent and $\mathcal{G}(\mathbf{c}_i, T_i, t),\ t \in [0, T_i]$ concerns only $\mathbf{c}_i$ and T_i. For the i^{th} piece of polynomial and the o^{th} term in Eq.3d, we denote the function value evaluated at time stamp t as

$$\mathcal{P}_o(\mathbf{c}_i, T_i, t) = \max\left[\mathcal{G}_o(\mathbf{c}_i, T_i, t),\ 0\right]. \tag{6}$$

The constructed penalty function for piece i and term o is

$$\begin{aligned}\mathcal{J}_o(\mathbf{c}_i, T_i, \delta) = \lambda_o\Big(\delta\sum_{j=0}^{n}\omega_j\mathcal{P}_o(\mathbf{c}_i, T_i, j\delta) + \frac{1}{2}(T_i \\ - n\delta)\cdot\left[\mathcal{P}_o(\mathbf{c}_i, T_i, n\delta) + \mathcal{P}_o(\mathbf{c}_i, T_i, T_i)\right]\Big),\ n = \left\lfloor\frac{T_i}{\delta}\right\rfloor,\end{aligned} \tag{7}$$

where λ_o is the penalty factor of the o^{th} term, δ the sampling time interval, $n = \lfloor T_i/\delta \rfloor \in \mathbb{Z}_+$ the number of sampling points, $(\omega_0, \omega_1, \cdots, \omega_{j-1}, \omega_j) = (1/2, 1, \cdots, 1, 1/2)$ the integral parameter of the trapezoidal rule.

Though n may change as T_i changes, the penalty function constructed with a fixed time interval is continuous with T_i, which is important for optimization.

E. Unconstrained Formulation & Derivative Deducing

So far, by adopting the MINCO trajectory class to eliminate the equality constraints, and by using penalty methods to eliminate the inequality constraints, we transform Eq.3 to an unconstrained optimization problem as follows,

$$\min_{\{\mathbf{q},\mathbf{T}\}}\left\{\mathcal{J}_{ctrl} + \rho_r(T_f - T_{ref})^2 + \sum_{o=1}^{5}\sum_{i=1}^{M}\mathcal{J}_o(\mathbf{c}_i, T_i, \delta)\right\}. \tag{8}$$

We now apply the chain rule to deduce derivatives of each component in Eq.8 with respect to $\{\mathbf{c}, \mathbf{T}\}$, and the derivatives with respect to the decision variables $\{\mathbf{q}, \mathbf{T}\}$ is provided by MINCO.

1) For the control effort cost,

$$\begin{aligned}\mathcal{J}_{ctrl} &= \sum_{i=0}^{M}\int_0^{T_i}\left(\frac{1}{2}\mathbf{u}_i(t)^{\mathsf{T}}\mathbf{u}_i(t)\right)dt \\ &= \frac{1}{2}\sum_{i=1}^{M}\mathbf{c}_i^{\mathsf{T}}\mathbf{Q}_{ctrl}(T_i)\mathbf{c}_i,\end{aligned} \tag{9}$$

$$\frac{\partial\mathcal{J}_{ctrl}}{\partial\mathbf{c}_i} = \mathbf{Q}_{ctrl}(T_i)\mathbf{c}_i, \quad \frac{\partial\mathcal{J}_{ctrl}}{\partial T_i} = \frac{1}{2}\mathbf{c}_i^{\mathsf{T}}\dot{\mathbf{Q}}_{ctrl}(T_i)\mathbf{c}_i.$$

2) For the duration related cost $\mathcal{J}_T = \rho_r(T_f - T_{ref})^2$,

$$\frac{\partial\mathcal{J}_T}{\partial T_i} = 2\rho_r(T_f - T_{ref}). \tag{10}$$

3) For each term of the functional costs $\mathcal{J}_o$, we only need to compute the derivatives for sample points that the penalty is positive, and the derivatives are

$$\begin{aligned}\frac{\partial\mathcal{J}_o}{\partial\mathbf{c}_i} =& \lambda_o\Big(\delta\sum_{j=0}^{n}\omega_j\frac{\partial\mathcal{G}_o(\mathbf{c}_i, T_i, j\delta)}{\partial\mathbf{c}_i} + \frac{1}{2}(T_i \\ & - n\delta)\Big[\frac{\partial\mathcal{G}_o(\mathbf{c}_i, T_i, n\delta)}{\partial\mathbf{c}_i} + \frac{\partial\mathcal{G}_o(\mathbf{c}_i, T_i, T_i)}{\partial\mathbf{c}_i}\Big]\Big), \\ \frac{\partial\mathcal{J}_o}{\partial T_i} =& \lambda_o\Big(\frac{1}{2}\left[\mathcal{G}_o(\mathbf{c}_i, T_i, n\delta) + \mathcal{G}_o(\mathbf{c}_i, T_i, T_i)\right]\Big),\end{aligned} \tag{11}$$

where $\partial\mathcal{G}_o(\mathbf{c}_i, T_i, t)/\partial\mathbf{c}_i$ differs for each term in Eq.(4). We give the derivatives of each term regarding $\{\mathbf{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}, f\}$ as below, while their derivatives relating to $\{\mathbf{x}, \mathbf{u}\}$ is obtained by automatic differentiation[1] according to Eq.(2), and finally $\partial\mathcal{G}_o(\mathbf{c}_i, T_i, t)/\partial\mathbf{c}_i$ can be calculated by the chain rule.

1) For the first term about the velocity,

$$\frac{\partial\mathcal{G}_1(\mathbf{c}_i, T_i, t)}{\partial\boldsymbol{v}(\mathbf{c}_i, t)} = 2\boldsymbol{v}(\mathbf{c}_i, t), \tag{12}$$

2) For the second term about the tilting angle, we use quaternion $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^{\mathsf{T}}$ obtained through Hopf Fibration [26] to denote the rotation, and $\mathbf{R} = \mathcal{R}_{quat}(\boldsymbol{q})$ can then be calculated as described in [27].

$$\frac{\partial\mathcal{G}_2(\mathbf{c}_i, T_i, t)}{\partial\boldsymbol{q}(\mathbf{c}_i, T_i, t)} = \frac{-2}{\sqrt{1 - \left(\mathbf{e}_3^{\mathsf{T}}\mathbf{R}\mathbf{e}_3\right)^2}}\cdot(q_0, -q_1, -q_2, q_3)^{\mathsf{T}}, \tag{13}$$

3) For the third term relating to body rate,

$$\frac{\partial\mathcal{G}_3(\mathbf{c}_i, T_i, t)}{\partial\boldsymbol{\omega}(\mathbf{c}_i, T_i, t)} = 2\boldsymbol{\omega}(\mathbf{c}_i, T_i, t), \tag{14}$$

4) For the fourth term relating to thrust,

$$\frac{\partial\mathcal{G}_4(\mathbf{c}_i, T_i, t)}{\partial f(\mathbf{c}_i, T_i, t)} = 2\left(f(\mathbf{c}_i, T_i, t) - f_m\right), \tag{15}$$

5) For the fifth term about collision avoidance,

$$\frac{\partial\mathcal{G}_5(\mathbf{c}_i, T_i, t)}{\partial\mathbf{p}(\mathbf{c}_i, T_i, t)} = -\frac{\partial f_{\mathcal{DIST}}}{\partial\mathbf{p}(\mathbf{c}_i, t)}, \tag{16}$$

which is obtained by trilinear interpolation in the distance field. We adopt L-BFGS[2] [28] to solve the unconstrained optimization problem Eq.8.

V. Experiment Results

A. Implementation Details

To validate the proposed method, we adopt the high-speed trajectory following scenarios simultaneously avoiding obstacles. As shown in Fig.1, we conduct a simulated navigation test in a $100m \times 100m \times 4m$ large underground garage scene. The quadrotor is equipped with a 32-line lidar with 90-degree vertical FOV and 360-degree horizontal FOV. The current laser point cloud observation information and odometry information are integrated to update the local occupancy grid map in real-time. The local mapping range is 32m × 32m × 8m, with a resolution of 0.25m, centered

[1]http://tapenade.inria.fr:8080/tapenade/index.jsp

[2]https://github.com/ZJU-FAST-Lab/LBFGS-Lite

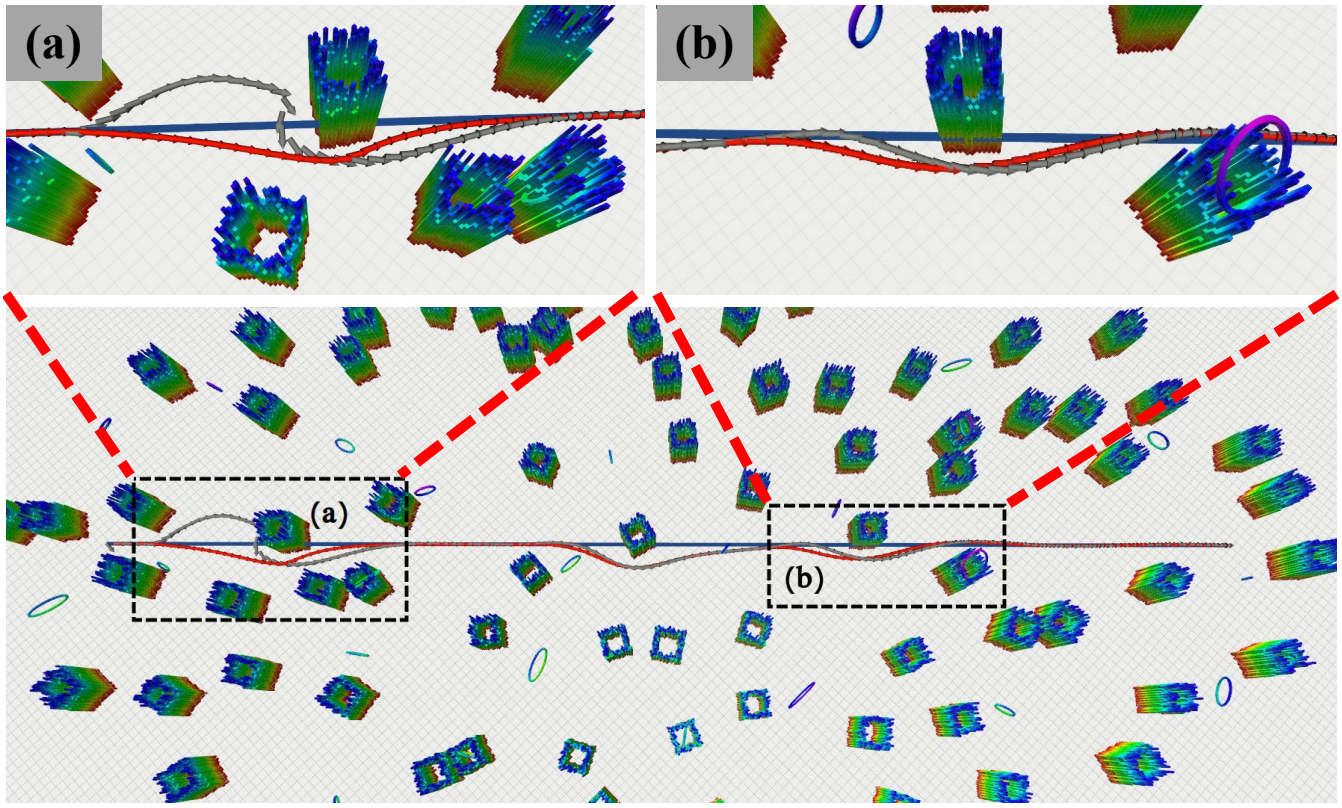


Fig. 5: Ablation experiment 1 results. The blue trajectory is the global reference trajectory. The route made of red arrows is generated by our method which considers the topological consistency. The gray arrows do not consider it.

on the quadrotor. This local occupancy grid map is then expanded by 2 resolution units in the horizontal direction and then converted into a gradient distance field map. The local map update frequency is 10Hz. All computations are performed with an Intel Core i7-10700 CPU @ 2.90GHZ. The planning parameters are shown in the Table.I and the convergence condition of the quasi-Newton method is set as the cost change rate is less than $1e-4$.

TABLE I: Parameters of Planning

v_{max}	θ_{max}	Ω_{max}	f_{min}	f_{max}	τ	$\mathcal{T}_{ref}$	m
$8[m/s]$	35[°]	$\pi/2$[rad/s]	$15[N]$	$60[N]$	$1[s]$	$2[s]$	$3[kg]$
λ_1	λ_2	λ_3	λ_4	λ_5	ρ_r	r_{tol}	d_{tol}
$1e3$	$1e4$	$1e4$	$1e2$	$1e6$	$1e11$	$1e3$	$0.65[m]$

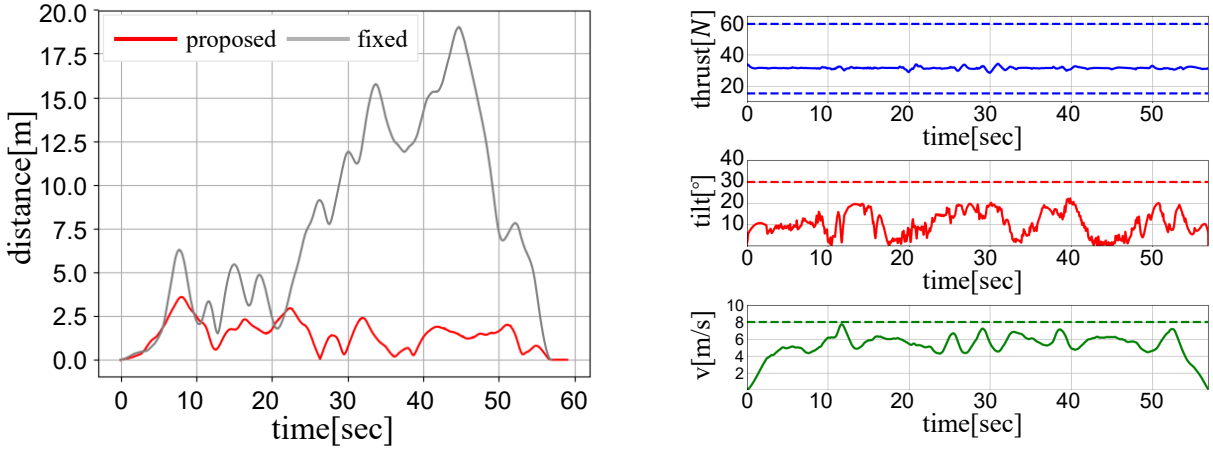
The replan starting state is selected from the last planning result. As for the ending state, we first select the state on the global reference trajectory at time T_{ref} from the current time. If the position of this state point is within an obstacle, then we slide backward until a non-colliding state point is found. If the previous replanned trajectory is not τ-topological consistent with the global reference trajectory, it means that obstacle avoidance is needed at this time and we use τ-topological consistency growing tree to get the initial trajectory. The initial trajectory solution time is set as $15ms$, then this initial trajectory is given to the flexible trajectory optimization method to flexibly optimize the trajectory termination state to satisfy all constraints.

B. Simulation Results

To validate the effectiveness of our method, we conduct two ablation experiments about topological consistency and flexible trajectory optimization respectively.

1) Ablation experiment 1: Under the same parameters setting as above, as shown in the Fig.5, we conduct the ablation experiment with the method without considering the replanning topological consistency. From Fig.5a we can see that the gray route initially chooses the left side to avoid the obstacle, and then changes to the right side, resulting in a twisted and more risky route. In contrast, the red route considers replanning topological consistency, and the final trajectory is smoother and safer.

2) Ablation experiment 2: We first verify the performance improvement of the proposed flexible replanning trajectory optimization method on tracking tasks. As shown in Fig.1, our proposed flexible trajectory generation method has better tracking results. The tracking difference is shown in Fig.6a, flexible trajectory generation method has a smaller tracking difference due to the elimination of delays taken into account during replanning. Compared with the fixed version, the flexible version takes slightly longer to calculate, but it also maintains a replanning frequency of 10HZ. Fig.6b shows the real-time thrust, tilt angle, and velocity curve of the quadrotor during the tracking process. All indicators are within the set constraints in the entire process.



(a) Tracking difference.

(b) Thrust, tilt angle, and velocity.

Fig. 6: Ablation experiment 2 results.

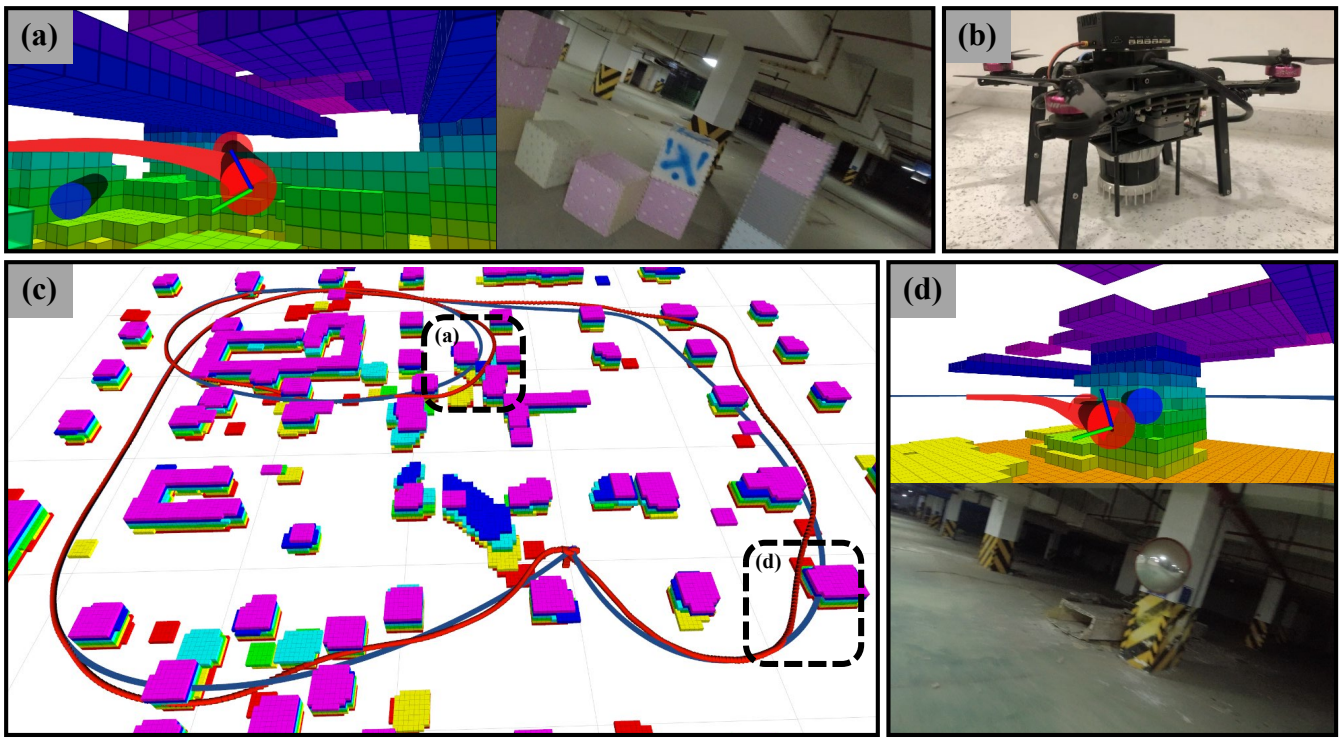


Fig. 7: Real world experiment results. (a)(d)The first perspective of the corresponding position in (c). (b)Hardware structure. (c)The blue trajectory is the global reference trajectory. The red one is the traversed trajectory.

C. Real-World Experiments

Real-world experiments are present in an underground garage in high-speed reference trajectory following scenarios simultaneously avoiding obstacles. We use Fast-LIO [29] for localization. All algorithms are computed onboard by a computer with an Intel Core i7-8550U CPU. As shown in Fig.7, the blue reference trajectory is blocked by obstacles, our method generates a smooth and safe trajectory to follow the reference as well as ensure safety.

VI. Conclusion & Future Work

In this work, we propose a framework to solve the problems of following a time-indexed trajectory as well as avoiding obstacles. To solve the conflicts between tracking the reference precisely, flying safely, and finishing the mission timely, we design a topology-preserving path search method to get a collision-free initial trajectory and design an optimization-based flexible trajectory generation method to satisfy dynamic constraints and rejoin the trajectory timely. We design ablation experiments to verify the role of τ-topological consistency in reducing trajectory risk. Simulation benchmarks and real-world applications validate the effectiveness of our algorithms. In future work, we will modify the τ-topological consistency definition to adapt to dynamic obstacles to achieve safe trajectory generation in dynamic scenarios.

REFERENCES

[1] E. J. Rodríguez-Seda, "Decentralized trajectory tracking with collision avoidance control for teams of unmanned vehicles with constant speed," in *American Control Conference*, 2014, pp. 1216–1223.

[2] E. J. Rodríguez-Seda and J. J. Dawkins, "Decentralized cooperative collision avoidance control for unmanned rotorcraft with bounded acceleration," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 11, pp. 2445–2454, 2018.

[3] S. Fukuda, Y. Satoh, and O. Sakata, "Trajectory-tracking control considering obstacle avoidance by using control barrier function," in *International Automatic Control Conference*, 2020, pp. 1–6.

[4] C. Peng, X. Liu, and J. Ma, "Design of safe optimal guidance with obstacle avoidance using control barrier function-based actorcritic reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 11, pp. 6861–6873, 2023.

[5] J. Liu, M. Zhao, and L. Qiao, "Adaptive barrier lyapunov function-based obstacle avoidance control for an autonomous underwater vehicle with multiple static and moving obstacles," *Ocean Engineering*, vol. 243, p. 110303, 2022.

[6] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5332–5339.

[7] B. Şenbaşlar, W. Hönig, and N. Ayanian, "Rlss: Real-time multi-robot trajectory replanning using linear spatial separations," *arXiv preprint arXiv:2103.07588*, 2021.

[8] X. Liu, M. Zhang, and E. Rogers, "Trajectory tracking control for autonomous underwater vehicles based on fuzzy re-planning of a local desired trajectory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 657–11 667, 2019.

[9] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[10] J. Ji, X. Zhou, C. Xu, and F. Gao, "Cmpcc: Corridor-based model predictive contouring control for aggressive drone flight," in *International Symposium on Experimental Robotics*. Springer, 2021, pp. 37–46.

[11] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Safe trajectory tracking in uncertain environments," *IEEE Transactions on Automatic Control*, vol. 68, no. 7, pp. 4204–4217, 2023.

[12] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, pp. 273–290, 2012.

[13] L. Jaillet and T. Siméon, "Path deformation roadmaps: Compact graphs with useful cycles for motion planning," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1175–1188, 2008.

[14] A. H. Quispe, T. Kunz, and M. Stilman, "Generation of diverse paths in 3d environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5994–5999.

[15] D. Yi, M. A. Goodrich, and K. D. Seppi, "Homotopy-aware rrt*: Toward human-robot topological path-planning," in *11th ACM/IEEE International Conference on Human-Robot Interaction*, 2016, pp. 279–286.

[16] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.

[17] B. Zhou, F. Gao, J. Pan, and S. Shen, "Robust real-time uav replanning using guided gradient-based optimization and topological paths," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 1208–1214.

[18] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5054–5061.

[19] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[20] 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.

[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] Z. Wang, C. Xu, and F. Gao, "Robust trajectory planning for spatial-temporal multi-drone coordination in large scenes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 12 182–12 188.

[23] 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.

[24] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2872–2879.

[25] K. L. Teo, V. Rehbock, and L. S. Jennings, "A new computational algorithm for functional inequality constrained optimization problems," *Automatica*, vol. 29, no. 3, pp. 789–792, 1993.

[26] M. Watterson and V. Kumar, "Control of quadrotors using the hopf fibration on so(3)," in *The International Symposium of Robotics Research*. Springer, 2020, pp. 199–215.

[27] J. Vince, *Quaternions for Computer Graphics*. Springer, 2011.

[28] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[29] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.