Downwash-Aware Trajectory Planning for Quadrotor Swarms

James A. Preiss, Wolfgang Hönig, Nora Ayanian, and Gaurav S. Sukhatme

***Abstract*— We describe a trajectory planning pipeline for large quadrotor teams in obstacle-rich environments. We construct a sparse roadmap in the environment and use a bounded-suboptimal conflict-based graph search to generate a discrete plan. We then refine this plan into into smooth trajectories using a spatial partition and Bézier curve basis. We model downwash directly, allowing safe flight in dense formations. We show simulation results with up to 200 robots and a real-robot experiment with 32 quadrotor. To our knowledge, our approach is the first solution which can compute safe and smooth trajectories for hundreds of quadrotor in dense environments with obstacles in a few minutes.**

Fig. 1. Long exposure of 32 Crazyflie nano-quadrotors flying through an obstacle-rich environment.

I. INTRODUCTION

Trajectory planning is a fundamental problem in multi-robot systems. Consider a team of N robots in an environment defined by convex polytope $\mathcal{W}$ and containing convex obstacles $\mathcal{O}_1 \dots \mathcal{O}_{N_{obs}}$, resulting in the obstacle-free configuration space $\mathcal{F} \subset \mathbb{R}^3$ for a single robot with known shape. We are given a start position for each robot $s^i \in \mathcal{F}$ and either a set of goal positions $G \subset \mathcal{F}, |G| = N$ (*unlabeled* case) or a goal position for each robot $g^i \in G$ (*labeled* case.) We seek the following:

- The total time duration $T \in \mathbb{R}_{>0}$ until the last robot reaches its goal
- In the unlabeled case an assignment of each robot to a goal position $g^{\phi(i)} \in G$, where ϕ is a permutation of $1, \dots, N$
- For each robot r^i, a trajectory $f^i : [0, T] \to \mathcal{F}$ where $f^i(0) = s^i, \quad f^i(T) = g^{\phi(i)}$, f^i must be continuous up to the C^{th} derivative (where C is user-specified), and collisions are avoided at all times for all robot pairs.

In particular, we are interested in solving this problem for large teams of quadrotors in tight formations. To account for the downwash force generated by one quadrotor's air stream on another, we treat each robot as an axis-aligned ellipsoid of radii $0 < r_x = r_y \ll r_z$. Due to the differential flatness of quadrotor dynamics, we focus on planning smooth trajectories in 3D Euclidean space and ignore the yaw angle.

A large body of work has addressed this problem. Graph search approaches (e.g. [1]) are adept at dealing with maze-like environments and scenarios with high congestion. However, directly executing a graph plan results in a piecewise linear path, requiring the robot to fully stop at each graph vertex to maintain dynamic feasibility. Continuous approaches [2] address this issue, but they are often tightly coupled, solving one large optimization problem in which the decision variables define all robots' trajectories. These approaches are typically demonstrated only on smaller teams.

Our approach decomposes the formation change problem into three steps. *Roadmap generation* generates a roadmap using the following inputs: model of the environment, collision model for robot-obstacle interaction, collision model for robot-robot interaction, and start and goal locations. *Discrete planning* solves the goal assignment problem (generating ϕ) and computes a timed sequence of waypoints for each robot on the generated roadmap. *Continuous refinement* uses the discrete plan as a starting point to compute a set of smooth trajectories, similar to [3] but adding support for three-dimensional ellipsoidal robots, environmental obstacles, and an anytime refinement stage to further improve the plan after generating an initial set of smooth trajectories.

II. ROADMAP GENERATION

A roadmap is an undirected connected graph of the environment $\mathcal{G}_E = (\mathcal{V}_E, \mathcal{E}_E)$, where each vertex $v \in \mathcal{V}_E$ corresponds to a location in $\mathcal{F}$ and each edge $(u, v) \in \mathcal{E}_E$ denotes that there is a linear path in $\mathcal{F}$ connecting u and v. We can generate a roadmap using standard methods such as PRM* or SPARS using the model of the environment and the robot-obstacle collision model (a sphere in the case of quadrotors). Due to the increased difficulty of the graph planning problem compared to the single-robot case, it is important to generate a sparse roadmap.

Generic roadmaps might not be suitable for multi-robot planning algorithms, because they do not include constraints between robots. Thus, we annotate the roadmap with generalized edge and vertex conflicts using the ellipsoid robot-robot collision model. Those conflicts constrain the proximity to other robots.

All authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA, USA.

Email: {japreiss, whoenig, ayanian, gaurav}@usc.edu

This work was partially supported by the ONR grants N00014-16-1-2907 and N00014-14-1-0734, and the ARL grant W911NF-14-D-0005.

III. DISCRETE PLANNING STAGE

We extend the *Multi-Agent Path-Finding* (MAPF) problem to support generalized edge and vertex conflicts. We are given an undirected connected graph of the environment At each discrete timestep, a robot can either wait at its current vertex or traverse an edge. We seek paths p^i such that the following properties hold:

P1: Each robot starts at its start vertex.
P2: Each robot ends at a unique goal location.
P3: At each timestep, each robot either stays at its current position or moves along an edge.
P4: There are no robots occupying the same location at the same time (*vertex collision*).
P5: There are no robots traversing the same edge in opposite directions (*edge collision*).
P6: Robots obey downwash constraints when stationary (*generalized vertex collision*).
P7: Robots obey downwash constraints while traversing an edge (*generalized edge collision*).

We can find an optimal solution for the unlabeled problem using an ILP-based formulation and we can compute a bounded suboptimal solution to the labeled problem by extending ECBS [1]. ECBS can also be used for unlabeled problems by computing a goal assignment using a linear bottleneck assignment formulation.

IV. CONTINUOUS REFINEMENT STAGE

In the continuous refinement stage, we convert the waypoint sequences p^i from the discrete planner into smooth trajectories f^i. We begin by finding partitioning the free space $\mathcal{F}$ into *safe corridors* for each robot. The safe corridor for robot r^i is a sequence of convex polyhedra $\mathcal{P}_k^i, k \in \{1 \ldots K\}$, such that, if each r^i travels within $\mathcal{P}_k^i$ during time interval $[t_{k-1}, t_k]$, both robot-robot and robot-obstacle collision avoidance are guaranteed. These polyhedra are computed by finding ellipsoid-weighted separating hyperplanes for each robot-robot interaction and for all obstacles within a local neighborhood around each robot's trajectory.

We then plan a smooth trajectory $f^i(t)$ for each robot, contained within the robot's safe corridor. We represent these trajectories as piecewise Bézier curves with one piece per time interval $[t_k, t_{k+1}]$. In the Bézier basis, constraining the trajectory to lie inside the safe corridor can be expressed as linear inequality constraints on the curve's control points.

We select an optimal Bézier trajectory by minimizing a weighted combination of the integrated squared derivatives:

$$\mathrm{cost}(f^i) = \sum_{c=1}^{C} \gamma_c \int_0^T \left\| \frac{d^c}{dt^c} f^i(t) \right\|_2^2 dt$$

where the $\gamma_c \geq 0$ are user-chosen weights. This cost is a quadratic function of the control points, forming a quadratic program along with the linear corridor constraints. We solve one instance of this quadratic program per robot in parallel.

We further improve these trajectories with an *iterative refinement* stage. We use the smooth trajectories to define a new spatial decomposition based on sampled points along each polynomial piece, producing new safe corridors that are roughly "centered" on the smooth trajectories rather than on the discrete plan. We then repeat the same optimization method to solve for a new set of smooth trajectories. Intuitively, iterative refinement provides a chance for the smooth trajectories to move further towards a local optimum that was not feasible under the original spatial decomposition.

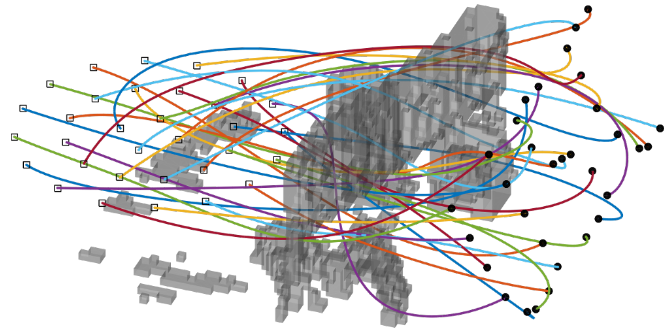

Fig. 2. Trajectory plans for "USC" formation change problem, executed on real robots. See section V for details.

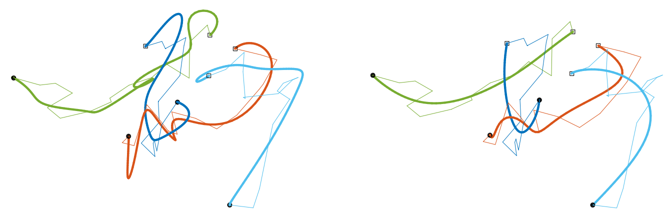

Fig. 3. Illustration of iterative refinement. Fine lines represent the discrete plans p^i; heavy curves represent the continuous trajectories f^i (left: after 1 iteration; right: after 6 iterations) for a subset of four robots.

V. EXPERIMENTS

We evaluate our method on real robots with 32 Crazyflie nano-quadrotors. The robots begin in concentric circles in the $x-y$ plane, fly through a cluttered set of obstacles, and form the letters "USC" in the air. The obstacle map was produced with a structured-light depth camera and octree-based occupancy grid mapping with resolution of $0.1\,\mathrm{m}$. The discrete roadmap produced by the SPARS algorithm contained approximately 850 vertices and 3200 edges.

The final plan is illustrated in Fig. 2. Iterative refinement was able to able to significantly improve the trajectory smoothness, as illustrated in Fig. 3. This was also confirmed in prior work using grid-based discrete planning [4], where peak acceleration was reduced from 5.2 to 1.6 m/s^2 and peak angular velocity from 2.2 to 0.38 rad/s.

REFERENCES

[1] M. Barer, G. Sharon, R. Stern, and A. Felner, "Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem," in *Symposium on Combinatorial Search (SOCS)*, 2014.

[2] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[3] S. Tang and V. Kumar, "Safe and complete trajectory generation for robot teams with higher-order dynamics," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[4] J. A. Preiss, W. Hönig, N. Ayanian, and G. S. Sukhatme, "Downwash-aware trajectory planning for large quadcopter teams," *arXiv preprint arXiv:1704.04852*, 2017.