Distributed Drone Swarm Trajectory Planner Using Topology Planning Under Communication Delay

Xingxun Liu
College of Electrical and Information Engineering
Hunan University
Changsha, China
liuxingxun@hnu.edu.cn

Zhiqiang Miao*
College of Electrical and Information Engineering
Hunan University
Changsha, China
miaozhiqiang@hnu.edu.cn

Yaonan Wang
College of Electrical and Information Engineering
Hunan University
Changsha, China
yaonan@hnu.edu.cn

2024 IEEE International Conference on Unmanned Systems (ICUS) | 979-8-3503-8418-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICUS61736.2024.10839900

***Abstract*—With the advancement of drone technology, drone swarms play a vital role in various applications. In drone swarms, path planning is crucial for ensuring both the efficiency and safety of swarm operations. However, in dynamic environments, communication delays may lead to instability and inefficiency in path planning. This paper addresses this issue by proposing a drone swarm path planning algorithm that takes communication delays into account. By analyzing the impact of communication delays on path planning and designing corresponding strategies to optimize the planning process, this algorithm aims to enhance the efficiency and safety of drone swarm operations. Through experimental validation, our computation time is reduced by 10.24%, proving the efficiency of the proposed method.**

***Index Terms*—drone swarm, path planning, communication delay, dynamic environments, topological planning**

I. Introduction

In recent years, unmanned aerial vehicles (UAVs) have garnered significant interest in the robotics community due to their mobility, agility, and flexibility. These capabilities enable UAVs operate autonomously in challenging environments that may be hazardous or inaccessible to human operators [1]. Many novel approaches have been proposed and validated in the real world. Some of these methods are centralized [2] [3] [4], while others are distributed [5] [6] [7]. The centralized approach involves a single machine planning the trajectory for each agent, whereas the decentralized approach enables each agent to plan its trajectory independently, providing robustness against potential failures of the centralized machine.

In the past, some centralised algorithms have used global MILFP [8] and SCP [9] methods, which are based on local optimization, to address UAV planning problems. However, as the number of drones increases, the issue becomes more complicated. Limited by computational power, these methods often cannot be applied effectively. Consequently, some scholars have proposed distributed motion planning methods, which offer good scalability. Tordesillas *et al*. [5] achieved motion planning for UAV clusters in dynamic dense environments by representing obstacles and UAVs as convex polyhedra and then decomposing the convex polyhedra into an optimisation problem to be solved with collision detection operations. Zhou *et al*. [7] proposed a decentralized and asynchronous system solution for collision avoidance by formulating the collision risk as a penalty in a non-linear optimisation problem, enabling drone swarms autonomous navigation in unknown, obstacle-rich scenarios using only on-board resources. Honig *et al*. [10] proposed a multi-robot trajectory planning method by first generating roadmaps in space through sampling, followed by generating feasible paths in discrete time and space using search to obtain the roadmaps and achieving smooth and continuous trajectories through continuous optimization. They also conducted test experiments using 32 quadrotors.

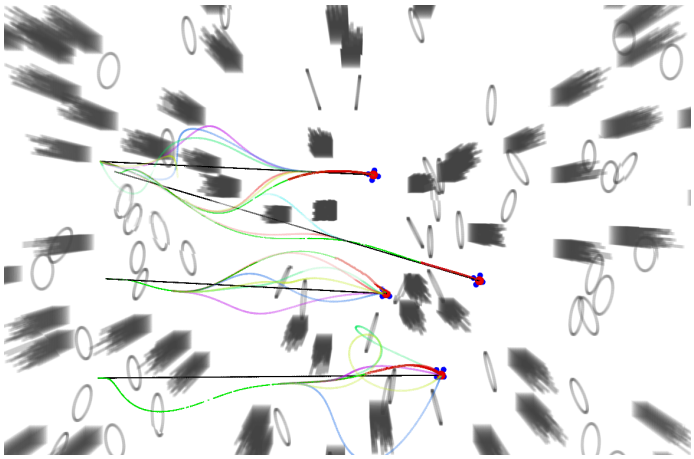

Fig. 1. Four quadrotors fly in simulation. Each color is associated with different topology trajectory.

Distributed algorithms implemented by means of optimization tend to fall into local minima, which in turn leads to the final generated trajectories not being globally optimal. Secondly, when the number of UAVs increases, it is often easy to lead to the failure of path planning because UAVs all tend to follow the path with the least cost. In order to solve this, several researchers have introduced the concept of topological planning into motion planning.

Rosmann et al. [11] introduce a method utilizing distinctive

This work was supported in part by the National Key Research and Development Program of China No. (2022YFB3903804), and the National Natural Science Foundation of China under Grant 62273138. (*Corresponding author: Zhiqiang Miao)

979-8-3503-8418-5/24/$31.00 ©2024 IEEE

topologies through Voronoi and sampling-based approaches. However, it is significantly more straightforward in 3-D contexts. To capture unique and beneficial paths, Jaillet et al. [12] develop visibility deformation roadmaps that encode richer and more pertinent information compared to typical paths for homology classes. Zhou et al. [13] introduced an efficient topological equivalence checking method for real-time topology planning.

In distributed algorithms, communication between drones is particularly important. Although the above-mentioned distributed algorithms are decentralized in the planning algorithm, agents may still require a centralized communication architecture. The current common method often idealizes communication conditions, assuming that data transmission between drones is completed immediately, without delay or loss. However, in real swarm systems, delays often occur during communication due to hardware limitations. These delays can be detrimental for multiple drones to avoid obstacles from one another [14]. Kondo *et al*. [15] introduced the idea of **DC (Delay Check)**, where each trajectory is checked for a period of time after generation to ensure that no new tracks are released during that time; if no new tracks are released, the trajectory is considered safe to execute. This method is clever, but with a large number of drones, it can often lead to situations where DC always returns false. Therefore, for large UAV swarms, this paper adopts the CDD (Communication Delay Danger) method, which checks only the agents in the CDD during DC, thus effectively enabling rapid planning of UAV swarms.

Therefore, this paper primarily focuses on effectively conducting cluster path planning to enhance robustness and security while considering communication delays.

In this study, a distributed asynchronous framework is employed to transform the UAV cluster motion planning problem into an optimization problem. The concept of topological paths is utilized to increase the planning success rate. Before a UAV executes a trajectory, other UAVs within the cluster are monitored to mitigate the effects of communication delays. For large UAV swarms, we adopt the CDD method (Communication Delay Danger), which selectively checks agents in the CDD during DC, thereby facilitating rapid planning of UAV swarms.

We summarize our contributions as follows:

1) In the front-end of UAV cluster path planning, collision possibilities is reduced by using topologically different definitions, thus reducing the number of re-planning and greatly improving the success rate of planning;
2) After the UAV completes the trajectory planning, the delay detection method is adopted, thus avoiding the impact on the UAV cluster due to the communication delay of the actual system;
3) The algorithm proposed in this paper is validated through simulation experiments.

II. Preliminaries

A. Motion Planning Preliminaries

Consider a swarm containing n drones, The UAV cluster motion planning problem can be formulated as finding N safe, dynamically constrained trajectories $f^i : [0, T] \to \mathbb{R}^3$ that satisfy the dynamics constraints in an environment $\mathcal{F} = (\mathcal{W} \setminus (\bigcup_h \mathcal{O}_h))$ bounded by $\mathcal{W}$ and containing convex obstacles $\mathcal{O}_1, \ldots, \mathcal{O}_{N_{\text{obs}}}$. where f^i denotes the trajectory of the i th UAV, and $T \in \mathbb{R} > 0$ is the total time.

B. UAV Model

The UAV studied in this paper is a quadrotor, characterized as an underactuated system. It is controlled using four inputs, corresponding to the magnitudes of the thrust generated by its four rotors, to manage its six degrees of translational and rotational freedom [16].

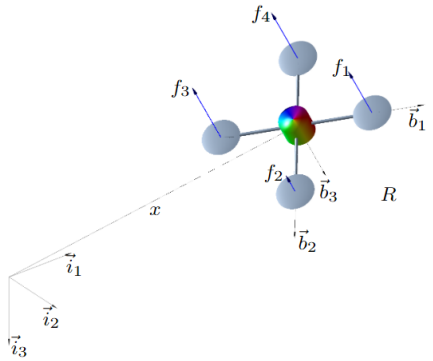

Fig. 2. where f_i represents the propeller lift.$\{\vec{i}_1, \vec{i}_2, \vec{i}_3\}$ represents inertial reference frame, $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ represents body reference frame,$R \in \text{SO}(3)$ represents the rotation matrix.

Fig. 2 depicts the dynamical model of a quadrotor. The motion of a UAV can be decomposed into translational motion of its center of mass and rotational motion about its center of mass. By applying the Newton-Euler equations, the motion model of a UAV can be formulated as follows:

$$\dot{x} = v, \tag{1}$$

$$m\dot{v} = mge_3 - fRe_3, \tag{2}$$

$$\dot{R} = R\hat{\Omega}, \tag{3}$$

$$J\dot{\Omega} + \Omega \times J\Omega = M \tag{4}$$

where $x \in \mathbb{R}^3$ represents the position, $v \in \mathbb{R}^3$ represents the velocity, $m \in \mathbb{R}$ represents the mass, $g \in \mathbb{R}$ represents the gravitational constant, $f \in \mathbb{R}$ represents the propeller lift, e_3 represents the inertial coordinate system, $\Omega \in \mathbb{R}^3$ stands for angular velocity, $J \in \mathbb{R}^{3\times3}$ stands for inertia matrix.

III. Local Planning

UAV motion planning often uses two steps: **Front-end path search** and **Back-end trajectory optimisation**. Typically, after obtaining an optimal path from the front-end search, a smooth trajectory is generated that satisfies dynamic constraints and can be directly executed by the UAV. Even with

the information on the shortest path, the trajectory obtained from the search often lacks higher-order information such as velocity and acceleration, which are crucial for accurately reflecting real-world motion dynamics. Consequently, the generated trajectory may not meet the operational requirements.

In [13], the Uniform Visibility Deformation (UVD) is defined to capture topologically distinct and beneficial trajectories, proving effective for equivalence checking. Similarly, we adopt the UVD definition to differentiate between topologically distinct paths. Fig. 3 illustrates an example where three paths belong to two different classes.

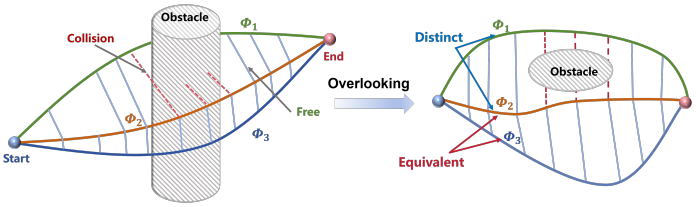

Fig. 3. An illustration of UVD. $\phi_2$is distinct to the ϕ_1, but is equivalent to the ϕ_3.

Two traces $\phi_1(s)$, $\phi_2(s)$ parameterized by $s \in [0,1]$ satisfy $\phi_1(0) = \tau_2(0)$, $\phi_1(1) = \phi_2(0)$, and for any moment s, the lines $\phi_1(s)\phi_2(s)$ of the two trajectories are conflict-free, then the two trajectories belong to the same topological path, and vice versa, they belong to paths with different topologies. After defining the UVD, the generation of topological paths can be done by sampling-based algorithms such as PRM or RRT. How to generate topological paths has been described in detail in [13], and will not be repeated in this paper.

Suppose there are n drones, each with m topological paths. We use $P_{n,m}$ to represent one of these paths. When generating topological paths, the position information of other UAVs is not taken into account, so the generated trajectories may have collisions. Fig. 4 provides an overview of this situation, where $P_{1,2}$ collide with $P_{2,4}$, $P_{1,3}$ collide with $P_{2,3}$ etc. Such trajectories consume unnecessary time in the later trajectory optimisation process, so we use an advanced detection technique to detect collisions after generating the discrete topological map. The specific algorithm is Alg. 1, which removes colliding trajectories by loops.

Similar to previous work [13], we utilize a B-spline curve to represent the UAV trajectory. A B-spline is a piecewise polynomial uniquely determined by its degree p_b, a set of $N+1$control points$\{Q_0, Q_1, \cdots, Q_N\}$ and a knot vector $\{t_0, t_1, \cdots, t_M\}$, in which $Q_i \in R^3$, $t_m \in R$ and $M = N + p_b + 1$.

For each UAV, the trajectory generation problem is a process of solving for the minimum of an objective function. The objective function is slightly different than when solving for single UAV trajectory planning, requiring the addition of an intra-cluster penalty term so that no collisions occur within the swarm. We use $\mathbf{\Phi}_i$ to denote the trajectory of each drone.To obtain the optimal trajectory, we can solve the following

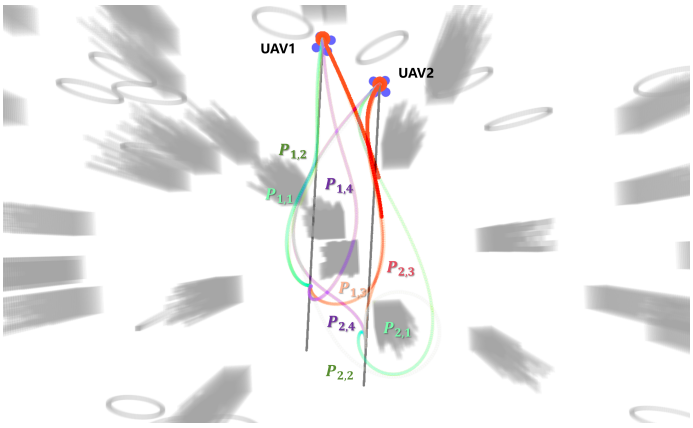

Fig. 4. There is a collision between the paths obtained by the two UAVs via topological path search.

Algorithm 1 Pure

Input: UAV number n, topo path $P_{n,m}$

```
Initialize()
for i = 1 : n − 1 do
  for j = 1 : m do
    for k = i + 1 : n do
      for l = 1 : m do
        if P_{i,j} and P_{k,l} is collision then
          P_{i,j}.pop
        end if
      end for
    end for
  end for
end for
```

optimization problem:

$$\min_{\mathcal{Q}} J_i = \lambda_s J_s + \lambda_c J_c + \lambda_d J_d + \lambda_w J_w \tag{5}$$

where J_s stands for the smoothness of the trajectory, J_c for the safety of the trajectory, and J_d for the feasibility of the trajectory,and J_w stands for the safety of the cluster. The details of how to solve for J_s,J_c and J_d will not be repeated here, more details can be found in [17]. Here we give the formula for calculating J_w :

$$J_{w,i} = \sum_k \int_{t=t_s}^{t_e} \begin{cases} d_{k,i}(t)^2 \\ 0 \end{cases} dt, \quad \begin{matrix} d_{k,i}(t) < 0 \\ d_{k,i}(t) \geq 0 \end{matrix} \tag{6}$$

$$d_{k,i}(t) = ||\mathbf{E}^{1/2} [\mathbf{\Phi}_k(t) - \mathbf{\Phi}_i(t)]|| - \mathcal{C} \tag{7}$$

where k stands for other drones, $\mathcal{C}$ stands for thresholds, and different thresholds can meet different needs, $\mathbf{E} := \mathrm{diag}(1, 1, 1/\mathrm{c}), \mathrm{c} > 1$ transforms Euclidean distance into ellipsoidal distance.

After obtaining all the gradients of the optimization variables using the above approaches, the problem is then efficiently solved with the non-linear optimization solver NLopt.[1]

[1]https://nlopt.readthedocs.io/en/latest/

IV. ELIMINATE COMMUNICATION LATENCY

Some current approaches usually assume that communication between UAVs is without delay and that trajectories generated by one UAV completing planning can be immediately accepted by other UAVs and applied in optimized trajectory generation.

However, in practice, due to the limitations of communication technology, there is often a delay in communication between UAVs, resulting in the inability to use the real-time trajectory information of other UAVs as constraint to achieve conflict cancellation in path planning. Fig. 6 illustrates an example of collisions due to communication delays.

In order to address the impact of communication delays, Kondo *et al*. [15] proposes an idea of delay detection. When a UAV generates an executable trajectory through the front-end and back-end, the system does not immediately feed it to the UAV hardware for execution but instead detects if it receives a plan issued by another UAV over a period of time (δ_{delay}, where δ_{delay} is the maximum delay time). Because of the communication delay, the received planning is not considered during path planning. Therefore, if no other UAV is detected posting a trajectory for a sustained period of time, the optimized trajectory can be considered safe and the computation can be sent to the UAV for execution. If there are other UAVs posting within δ_{delay} time, the optimized trajectory is considered to be risky and requires further processing.

A. CDD(Communication Delay Danger) Definition

This method solves the effect of communication delay very well, but its actual execution will detect whether all drones release new trajectories within δ_{delay}, so it will lead to many cases that the optimized trajectory is considered to be risky, resulting in frequent hovering of the drone. Therefore, in this paper, we first define CDD: Considering the UAV danger area in the case of communication delay, only the UAVs within the danger area will be detected during the subsequent detection. By defining the CDD, the number of UAV stops can be effectively reduced, thus reducing the overall planning time.

Algorithm 2 CCD Check

```
Function CCD CHECK(x_i)
for all UAVs do
    if d < 2x_max then
        C.add(UAV)
    end if
end for
```

First, we propose a definition of the Communication Delay Danger (CDD). When UAVs are flying in swarms, not all of them have communication delays that have a significant impact on planning. Therefore we need to determine the extent of the region of UAVs that may have an impact on planning. There exists a notion of a maximum CDD when UAVs are planning at maximum speed v_{max} as well as maximum delay time t_{max} , we define it as follows: $C = \{ x_i \mid d \leq 2x_{max} \}$ where d represents the distance between the known UAV trajectory endpoints: $d = \mathbf{\Phi}_1(t_e) - \mathbf{\Phi}_2(t_e)$. where $x_{max} = v_{max} \times t_{max}$ represents the distance that the UAVs can move under the time of maximum communication delay. Fig. 7 is a demonstration of Communication Delay Danger.

When the distance between the two drones is greater than $2x_{max}$, it can be assumed that no collision will occur even if there is a communication delay. Therefore, in the planning, for the communication delay, we only need to consider the UAVs that are within the CDD region for consideration.

B. Delay Check

In an asynchronous planning framework, UAVs can publish new trajectory information at any time, communication delays exist making it impossible for UAVs to obtain the paths published by other intelligences in the recent past, and the situation becomes more complicated when the number of UAVs increases. Therefore, additional steps need to be taken to deal with communication delays and to ensure that newly optimized paths do not conflict with paths recently published by other agents Here we use DC for the processing, and the core idea is that the generated trajectories will be continuously detected for δ_{delay} to ensure that no new trajectories are published during this time. The specific algorithm is shown in Alg. 3.

Algorithm 3 Delay Check

```
1:  while not reach goal do
2:      traj_{1_opt} = OPT()
3:      if CHECK(traj_{B_opt}) == False then
4:          Continue
5:      end if
6:      Publish traj_{B_opt}
7:      if DELAY CHECK(traj_{1_opt}) == False then
8:          Keeps executing traj_{1_comm} and go to Line 11
9:      end if
10:     traj_{B_comm} ← traj_{B_opt}
11:     Publish traj_{B_comm}
12: end while
```

V. SIMULATION

The architecture of the entire simulation platform is depicted in Fig. 5. Taking a single UAV as an example, it acquires sensor data, such as IMU readings and depth camera images, from the simulation environment. The planner utilizes this data to perform motion planning for the UAV and subsequently shares the generated trajectories with other UAVs.

We execute our implementation on a PC running Ubuntu 20.04, equipped with a 13th Gen Intel(R) Core(TM) i9-13900HX CPU and 32 GB of RAM. This CPU features 32 physical cores, significantly enhancing the runtime performance for continuous tasks. Fig. 1 shows a snapshot of the simulation process. We utilize an occupancy grid to represent the environment, with a map size of $40 \times 30 \times 2\mathrm{m}$, The resolution of the map is $0.2\mathrm{m}$. Grey squares denote randomly generated obstacles, comprising 100 rectangular obstacles and

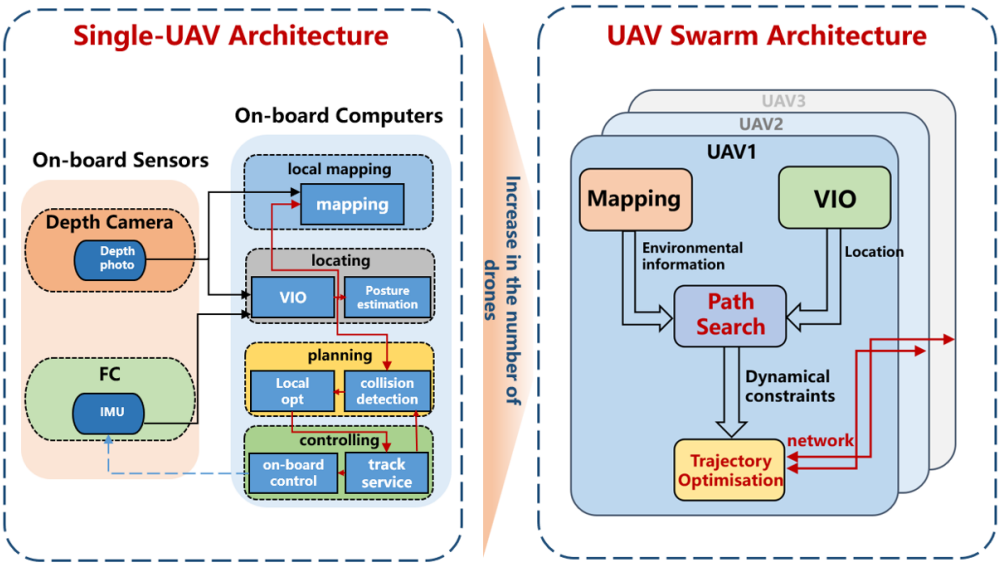

Fig. 5. System Architecture

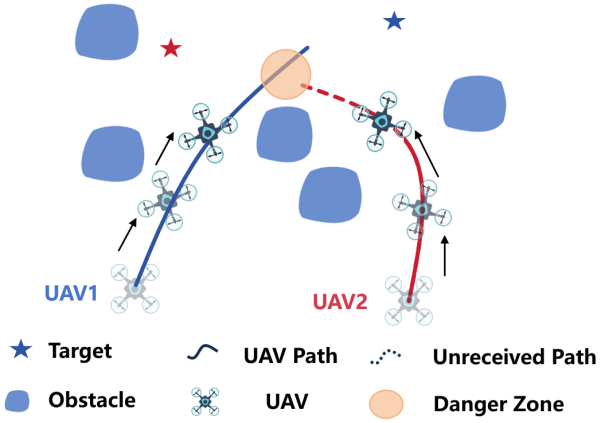

Fig. 6. For UAV 1, the second half of UAV 2's trajectory is not received temporarily due to communication delays, and therefore often appears not to be considered in the trajectory planning phase. Resulting in collision prone at the end

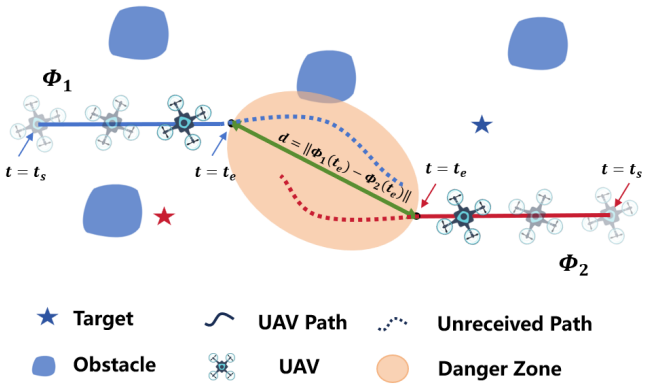

Fig. 7. Communication Delay Danger

50 circular obstacles. Fig. 8 provides a snapshot of the simulation environment. The UAV model conforms to the dynamics where v_{max} is $2m/s$ and a_{max} is $2m/s^2$. The generated trajectories are tracked using the dynamics model of the UAV through the proposed in [18].

We simulated three UAVs flying from one side of the map at a speed of 3m/s with a quadrotor radius of 0.2 m. Each UAV independently perceives the environment, and the resulting trajectories are visually distinguished by different colors. Fig. 9 illustrates the paths generated through planning.

We compared our work with method [7] and method [15], as summarized in Table I, where the starting point and goal

TABLE I
COMPARISONS OF PLANNING METHODS.

Delay Times [ms]	Method	Collision Free Rate [%]	Avg Number of Stops[s]	Avg Travel Times [s]
$\delta_{delay} = 20$	EGO-Swarm	38	0.075	4.67
	RMADER	100	0.127	10.57
	Ours	100	0.098	10.03
$\delta_{delay} = 50$	EGO-Swarm	25	0.075	4.67
	RMADER	100	0.198	11.48
	Ours	100	0.139	10.84
$\delta_{delay} = 100$	EGO-Swarm	22	0.075	4.67
	RMADER	100	0.326	12.89
	Ours	100	0.27	11.57

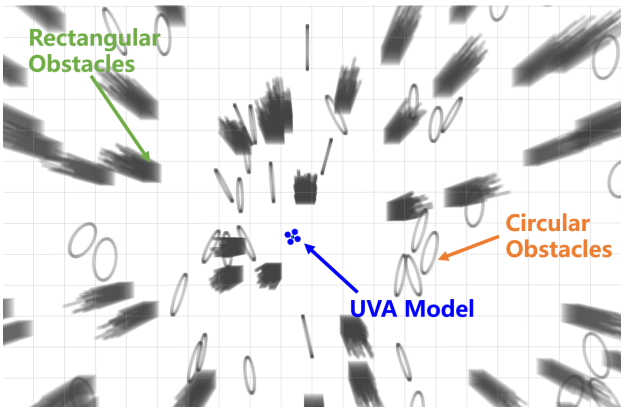

Fig. 8. Simulation environment.

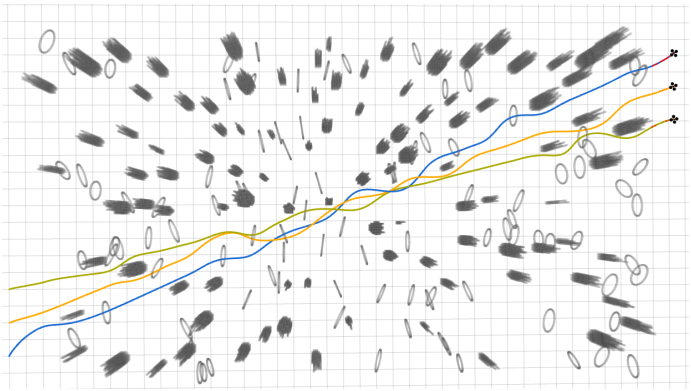

Fig. 9. Three UAVs pass through an obstacle-heavy environment.

are the same. As shown in Table I, our method balances the collision-free rate and the number of stops. Compared to [7], our method achieves a higher success rate. Furthermore, compared to [15], our method results in fewer stops and shorter travel times.

VI. CONCLUSION

In this paper, we propose a distributed planer aimed at generating feasible trajectories that consider communication delays in unknown and dense environments. By restricting conflict resolution between UAVs to the Communication Delay Domain (CDD), we ensure the discovery of viable trajectories within a limited timeframe. The CDD is defined by the maximum delay time and speed to guarantee security in cluster path planning. Simulation experiments validate the effectiveness of our approach. In future work, we plan to extend our method to more complex scenarios, such as maze-like environments and dynamic settings, and validate them in real-world applications.

REFERENCES

[1] S. -J. Chung, A. A. Paranjape, P. Dames, S. Shen and V. Kumar, "A survey on aerial swarm robotics," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 837-855, Aug. 2018.
[2] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in Proc.IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 1917–1922, 2012.
[3] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," Auton. Robots, vol. 35, no. 4, pp. 287-300, 2013.
[4] W. Wu, F. Gao, L. Wang, B. Zhou, and S. Shen, "Temporal scheduling and optimization for multi-MAV planning," Ph.D. dissertation, Hong Kong Univ. Sci. Technol., Hong Kong, 2019.
[5] J. Tordesillas and J. P. How, "MADER: trajectory planner in multiagent and dynamic environments," in IEEE Transactions on Robotics, vol. 38, no. 1, pp. 463-476, Feb. 2022.
[6] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," in IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 375-382, April 2019.
[7] X. Zhou, X. Wen, Z. Wang,, Y.Gao, H. Li, Q. Wang , F. Gao, "Swarm of micro flying robots in the wild". Science Robotics, 7(66), eabm5954, 2022.
[8] T. Schouwenaars, B. De Moor, E. Feron and J. How, "Mixed integer programming for multi-vehicle path planning," 2001 European Control Conference (ECC), Porto, Portugal, 2001, pp. 2603-2608.
[9] Y. Chen, M. Cutler and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 5954-5961.
[10] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme and N. Ayanian, "Trajectory planning for quadrotor swarms," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 856-869, Aug. 2018.
[11] C. Rösmann, F. Hoffmann and T. Bertram, "Planning of multiple robot trajectories in distinctive topologies," 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2015, pp. 1-6.
[12] J. Léonard, and T. Simeon. "Path deformation roadmaps: compact graphs with useful cycles for motion planning." The International Journal of Robotics Research, November, 1175–88, 2008.
[13] B. Zhou, F. Gao, J. Pan and S. Shen, "Robust real-time UAV replanning using guided gradient-based optimization and topological paths," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 1208-1214.
[14] J. Gielis, A. Shankar, and A. Prorok. "A critical review of communications in multi-robot systems," June, 2022.
[15] K. Kondo, R. Figueroa, J. Rached, J. Tordesillas, P. C. Lusk and J. P. How, "Robust MADER: Decentralized multiagent trajectory planner robust to communication delay in dynamic environments," in IEEE Robotics and Automation Letters, vol. 9, no. 2, pp. 1476-1483, Feb. 2024.
[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 2520-2525.
[17] B. Zhou, F. Gao, L. Wang, C. Liu and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," in IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3529-3536, Oct. 2019.
[18] T. Lee, M. Leok and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 2010, pp. 5420-5425