

Optimal virtual tube planning and control for swarm robotics

The International Journal of
Robotics Research
2024, Vol. 43(5) 602–627
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/02783649231210012
journals.sagepub.com/home/ijrr



Pengda Mao , Rao Fu and Quan Quan

Abstract

This paper presents a novel method for efficiently solving a trajectory planning problem for swarm robotics in cluttered environments. Recent research has demonstrated high success rates in real-time local trajectory planning for swarm robotics in cluttered environments, but optimizing trajectories for each robot is still computationally expensive, with a computational complexity from $O(k(n_t, \varepsilon)n_t^2)$ to $O(k(n_t, \varepsilon)n_t^3)$ where n_t is the number of parameters in the parameterized trajectory, ε is precision, and $k(n_t, \varepsilon)$ is the number of iterations with respect to n_t and ε . Furthermore, the swarm is difficult to move as a group. To address this issue, we define and then construct the optimal virtual tube, which includes infinite optimal trajectories. Under certain conditions, any optimal trajectory in the optimal virtual tube can be expressed as a convex combination of a finite number of optimal trajectories, with a computational complexity of $O(n_t)$. Afterward, a hierarchical approach including a planning method of the optimal virtual tube with minimizing energy and distributed model predictive control is proposed. In simulations and experiments, the proposed approach is validated and its effectiveness over other methods is demonstrated through comparison.

Keywords

swarm robotics, trajectory planning, virtual tubes, optimization

Received 6 April 2023; Revised 8 September 2023; Accepted 1 October 2023

Senior Editor: Dongjun Lee

Associate Editor: Lorenzo Sabattini

1. Introduction

Swarm robotics has potential applications in various real-world scenarios, such as air traffic control, land and sea search and rescue, and target detection. A certain amount of research has been dedicated to the smooth and safe navigation of swarm robotics passing through obstacle-dense environments.

Passing through an obstacle-dense environment safely and smoothly as a group is still a challenging task for swarm robotics. Fast and smooth movement of a swarm can be achieved through distributed trajectory planning, but such method may also cause the swarm to split apart. On the other hand, centralized formation path planning combined with multi-robot formation control can achieve swarm passing-through in formation, but it lacks flexibility and maneuverability. In this work, an optimal virtual tube planning method inspired by the application of tubes in transporting objects in real life is proposed for a robotic swarm to centrally plan an optimal virtual tube that provides a safe area without obstacles and forward commands in the environment; then the distributed model predictive control is used to track the commands, thereby achieving fast and smooth passing of the swarm as a group in obstacle-dense environments.

We employ the path planning algorithm to plan a safe area without obstacles for a swarm to form a group, which effectively simplifies the obstacle avoidance problem by replacing obstacle constraints with the tube boundaries. Hence, the robots in the swarm only need to consider inter-robot collision avoidance and tube boundary constraints. Furthermore, the fact that optimal trajectories do not intersect with each other in the tube substantially reduces the chance that collision avoidance control is triggered. Based on optimization theories, we propose a method with a computational complexity of $O(n_t)$, where n_t is the number of parameters in trajectory to generate infinite optimal trajectories with minimizing energy. This method has a significant computational advantage over existing methods with the computational complexity from $O(k(n_t, \varepsilon)n_t^2)$ to $O(k(n_t, \varepsilon)n_t^3)$ where ε is precision and

School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

Corresponding author:

Quan Quan, School of Automation Science and Electrical Engineering, Beihang University, XueYuan Road No.37, Beijing 100191, China.

Email: qq_buaa@buaa.edu.cn

$k(n_t, \varepsilon)$ is the number of iterations needed to solve the optimization problem, making it possible to generate optimal trajectories for large-scale robot swarms in a timely and centered fashion. The optimal virtual tube planning method is validated effectively by using the distributed model predictive control method in simulations and experiments. Furthermore, the superiority of our method is demonstrated through comparison with other methods.

1.1. Related work

In this section, we provide an overview of the related studies on topics including trajectory planning, infinite homotopic trajectory generation, and virtual tube control and planning.

1.1.1. Trajectory planning. Trajectory planning is an efficient method for robots to pass through a complex environment. Considering the kinematics of the robot and obstacles, an optimization problem is formulated to plan a smooth and safe trajectory for each robot. By leveraging the differential flatness, the optimal trajectory could be tracked by some robots. The methods of trajectory parameterization mainly include k -th order polynomials and B-spline. (i) To ensure passing through fixed waypoints, a trajectory can be parameterized as k -th order polynomials ([Mellinger and Kumar \(2011\)](#)). However, trajectories described as polynomials are prone to collisions with obstacles due to finite discrete avoidance constraints. To address this issue, various methods have been proposed for generating collision-free trajectories. For instance, a collision-free trajectory has been generated using mixed integer methods in [Deits and Tedrake \(2015\)](#). An approach proposed by [Richter et al. \(2016\)](#) refines the trajectory iteratively by minimizing a cost function and adding additional waypoints between two ends of a particular trajectory segment that passes through obstacles to avoid collisions. However, these methods are computationally intensive for real-time planning. To overcome these challenges, some methods have been proposed that use simple constraints to design a quadratic programming (QP) problem, which can significantly reduce the computation time. For example, safe flight corridor (SFC), a collection of convex connected polyhedra that models free space in a map, is used as a constraint for obstacle avoidance to generate a collision-free trajectory in real-time ([Liu et al. \(2017\); Gao et al. \(2020\)](#)). (ii) The trajectory parameterized by B-spline ([Usenko et al. \(2017\)](#)) is proposed to eliminate constraints of intermediate conditions on any derivative of the trajectory. An essential advantage of using B-splines is the convex hull property, which imposes bounds on both the trajectory and its derivatives. This property allows for strict bounding of the position, velocity, and acceleration of a robot ([Ding et al. \(2019\)](#)). The drawback of B-spline, however, is that the B-spline curve does not pass through the control points that represent waypoints. To repel the trajectory from obstacles, a gradient-based method is applied ([Zhou et al. \(2020\)](#)) such that a penalty function is introduced. Although this method

has been successful, it is experience dependent and lacks rigorous theoretical proof. It has been shown that if the control points satisfy certain conditions, the trajectory will strictly avoid obstacles ([Zhou et al. \(2019\)](#)).

To plan swarm trajectories, existing methods are mostly extension of single trajectory planning methods. However, in swarm robotics, other agents are regarded as dynamic obstacles, which increase the complexity of the problem. Several methods have been proposed to solve this avoidance problem, such as RSFC (Relative Safe Flight Corridor) ([Park et al. \(2021\)](#)), distributed predictive control ([Soria et al. \(2022\)](#)), HOOP (Hold Or take Optimal Plan) ([Tang et al. \(2018\)](#)), and EGO (ESDF-free Gradient-based lOcal) Swarm ([Zhou et al. \(2021\)](#)), all of which employ increasingly complex constraints. Safe and smooth trajectories need to be generated by optimization problems, which require significant computational resources with an increasing scale of the swarm. Furthermore, the constraint of gathering the swarm as a group is also needed to increase the computation cost. Therefore, the method efficiency for swarm robotics is a crucial point needed to be considered, especially with the limited onboard computing resources of robots.

1.1.2. Infinite homotopic trajectories generation. The objective of trajectory planning for a robot passing through an environment is to plan a trajectory from a start position to a goal position for each robot. Ideally, trajectory planning would yield a unique solution to the optimization problem, but [Orthey et al. \(2019\)](#) demonstrate that local minima in the optimization problem result in an infinite set of homotopic collision-free trajectories. To address this challenge, a generative model of collision-free trajectory is trained in [Osa \(2022\)](#) to represent an infinite set of homotopic solutions to the optimization problem. Motivated by the methods of finding homotopic trajectories for a robot, we proposed an *optimal virtual tube* for swarm robotics, which contains an infinite number of homotopic optimal trajectories in free space. Our focus is on finding infinite trajectories for a swarm, rather than for a single robot, although generating infinite optimal trajectories through infinite optimization problems is not feasible due to computational complexity. To mitigate this issue, we first propose and define the optimal virtual tube, including infinite optimal trajectories. Then, a theorem is proposed so that any optimal trajectory in the virtual tube could be represented by a convex combination of these finite number of optimal trajectories with a computational complexity of $O(n_t)$ if trajectories are linearly parameterized and satisfy the conditions in lemmas. Therefore, combining the advantages of formation control ([Alonso-Mora et al. \(2017\)](#)), the complexity of the optimal virtual tube planning is independent of the scale of the swarm.¹

1.1.3. Virtual tube control and planning. The “virtual tube” provides safety boundary and direction of motion for a swarm. This concept appeared in AIRBUS’ SkyRoad project, is a free space for unmanned aerial vehicles to fly over cities ([Airbus \(2019\)](#)). These virtual tubes establish a

safe flight area in which drone flights will not interfere with ground or air traffic. Motivated by this, in our previous work ([Quan et al. \(2022\)](#)), a *straight virtual tube* was proposed for swarm robotics. There are no obstacles inside the virtual tube, so the area inside can be regarded as a safety zone. Restricting the robot to fly in the safety zone can simplify the control, that is, only requiring to guarantee no collision with other robots and the no collision among robots and the boundary of the virtual tube, rather than the collision avoidance with complex obstacles in environment. The virtual tube ensures the safety of robots and cities. Then, to guide the swarm to pass through corridors, doors and windows, and surround surveillance targets without diverging ([Gao et al. \(2022\)](#)), we have also generalized the definition of virtual tubes and proposed the *curved virtual tube* ([Quan et al. \(2023b\)](#)). This concept of the curve virtual tube is similar to the lane for autonomous road vehicles in [Rasekhpor et al. \(2016\)](#), [Luo et al. \(2018\)](#) and corridor for a multi-UAV system in [Nagrare et al. \(2022\)](#).

As a summary, the earlier part of this subsection is dedicated to the introduction of two problems, namely, the *virtual tube planning problem* and *virtual tube passing-through control problem*. However, the above-mentioned methods all apply to passing-through control problems. For the virtual tube planning problem, in previous work ([Mao and Quan \(2022\)](#)), a generator curve is first obtained by trajectory planning based on several discrete waypoints, which are generated using search-based methods. Then the virtual tube is generated by expanding the generator curve while avoiding obstacles.

1.2. Motivations and contributions

Although there are many studies on the virtual tube passing-through control problem, a suitable design of the forward commands for the swarm is still lacking. And the previous virtual tube planning work also has many limitations. The motivations of the optimal virtual tube are demonstrated as follows.

First, the kinematics of the robot are not considered in design of the forward commands for the swarm in the virtual tube passing-through control problem. The desired speeds for robots are manually designed and the directions of the forward velocity commands of the robots parallel to the direction of the generator curve, which may cause the swarm to block performing turning and other maneuvers.

Secondly, the previous virtual tube planning method ([Mao and Quan \(2022\)](#)) poses limitations to the control algorithm and is not scalable in high-dimension spaces. The regular virtual tube is only suitable for some specific methods such as APF (Artificial Potential Field), flocking algorithm, and CBF (Control Barrier Function). And it is difficult to be extended into high-dimension spaces.

Thirdly, trajectories of swarm robotics in existing research are not optimal, namely, any robot in the virtual tube is not assigned an optimal trajectory. On the other hand, it costs much computation with the number of robots increasing if the trajectory optimization method is adopted.

To overcome these limitations, this paper first generalizes the definition of the virtual tube without a *generator curve* so that it is convenient to be applied in a high-dimension space, and the optimal virtual tube which includes infinite optimal trajectories is proposed for trajectory planning of swarm robotics. An optimal virtual tube planning method is proposed to generate the infinite optimal trajectories in the optimal virtual tube with a small computational complexity which makes the virtual tube applicable to a wider range of control methods. Then the optimal virtual tube planning method combined with model predictive control is validated in simulations and experiments, as shown in [Figure 1](#). The main contributions are as follows.

- Compared with our previous work ([Mao and Quan \(2022\)](#)), the definition of the virtual tube is extended into a high-dimension space. Further, the optimal virtual tube is defined for trajectory planning of swarm robotics, whose properties and advantages are analyzed in this paper. There are infinite optimal trajectories within the optimal virtual tube. Each robot is assigned to an optimal trajectory without intersecting with other trajectories.
- Based on the definitions and theorems, we show that a linear virtual tube with convex hull terminals is a type of optimal virtual tubes which is easy to implement in practice. The infinite optimal trajectories are generated by the convex combination of finite optimal trajectories. Therefore, the computational complexity is only $O(n_t)$, in contrast with solving existing optimization problems with computational complexity of $O(k(n_t, \varepsilon)n_t^2)$ to $O(k(n_t, \varepsilon)n_t^3)$.
- A hierarchical approach including an optimal virtual tube planning method and model predictive control is proposed. The corresponding application on drone swarm is demonstrated in simulations and experiments. Furthermore, comparisons with other methods are implemented in simulations to demonstrate the superior performance of our approach.

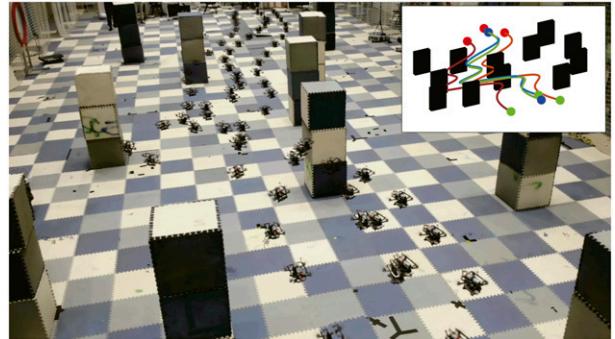


Figure 1. By optimal virtual tube method, an overlay image of four drones passing through the cluttered environment: The trajectory of each drone is labeled by a different color in the upper right figure. Compared to the traditional method, our method solves only three optimization problems for the three optimal trajectories of drones, such as the red, blue, and orange lines. The fourth optimal trajectory of the drone (the green line) is generated by an affine combination.

1.3. Paper organization

This paper is organized as follows. The preliminaries about topology and convex optimization, and problem formulation are described in Section 2. Section 3 is devoted to demonstrating the definitions such as virtual tube, optimal virtual tube, and linear virtual tube, and distinguishes the proposed work from our previous work. Then, a specific virtual tube, named linear virtual tube with convex hull terminals, is defined and proved to be an optimal virtual tube based on a proposed theorem in Section 4. In Section 5, a planning method to obtain an optimal virtual tube is proposed. Model predictive control is proposed for the swarm to track trajectories in an optimal virtual tube as a validation of the tube planning method in Section 6. Experimental results and comparisons in simulations are presented to illustrate the superior performance of our approach in Section 7. Finally, Section 8 contains the conclusion and future work.

2. Preliminaries and problem formulation

2.1. Topology

The topological theory is complex and profound. For the purpose of comprehensibility, we will roughly and intuitively introduce some basic concepts that need to be used in the following sections.

2.1.1. Interior. The *interior* of the subset \mathcal{T} of a topological space, denoted by $\text{int } \mathcal{T}$, is the largest open subset contained in \mathcal{T} , expressed as

$$\text{int}(\mathcal{T}) = \{\mathbf{x} \in \mathcal{T} | B(\mathbf{x}, r) \cap \mathcal{T} \subseteq \mathcal{T} \text{ for some } r > 0\},$$

where $B(\mathbf{x}, r) = \{\mathbf{y} | \|\mathbf{y} - \mathbf{x}\| \leq r\}$ represents the ball of radius r and center \mathbf{x} in the any norm $\|\cdot\|$.

2.1.2. Diffeomorphism. Intuitively, a manifold (Tu (2011)) is a generalization of curves and surfaces to higher dimensions. Given two manifolds M and N , a differentiable map $f: M \rightarrow N$ is called a *diffeomorphism* if it is a bijection and its inverse is differentiable as well.

2.1.3. Simple connectivity (Shick (2007)). Intuitively, a space is *simply-connected* if every loop in the space can be shrunk to a point within the space.

2.2. Convex optimization (Boyd and Vandenberghe (2004))

2.2.1. Convex hull. For a set $\mathcal{C} = \{\mathbf{x}_i\}, i = 1, 2, \dots, q$, the *convex hull* of the set \mathcal{C} , denoted $\text{conv } \mathcal{C}$, is the set of all convex combinations of points in \mathcal{C} :

$$\text{conv } \mathcal{C} = \left\{ \sum_{i=1}^q \theta_i \mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{C}, \theta_i \geq 0, \sum_{i=1}^q \theta_i = 1 \right\}.$$

The $\text{int}(\text{conv } \mathcal{C})$ and $\partial(\text{conv } \mathcal{C})$ denote the *interior* of $\text{conv } \mathcal{C}$ and the *boundary* of the $\text{conv } \mathcal{C}$, respectively. An example is shown in Figure 2.

2.2.2. An optimality criterion for differentiable f_0 . Suppose that a convex function f_0 is differentiable, so that for all $\mathbf{x}, \mathbf{y} \in \text{dom} f_0$ (domain of the function f_0),

$$f_0(\mathbf{y}) \geq f_0(\mathbf{x}) + \nabla f_0(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}). \quad (1)$$

Let \mathcal{X}_f denote the feasible set. Then \mathbf{x} is optimal if and only if $\mathbf{x} \in \mathcal{X}_f$ and

$$\nabla f_0(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \geq 0, \text{ for all } \mathbf{y} \in \mathcal{X}_f. \quad (2)$$

2.3. Problem formulation

Let \mathcal{X}_c be the configuration space and denote the obstacle-free space as $\mathcal{X}_{\text{free}}$. The trajectory between two given points $\mathbf{q}_0, \mathbf{q}_m \in \mathcal{X}_{\text{free}}$ is defined by a map $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t)$ where $t \in [0, 1]$, $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), 0) = \mathbf{q}_0$, $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), 1) = \mathbf{q}_m$.

Definition 1. A trajectory $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t)$ is *optimal* with respect to cost g if

$$\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t) \in \arg \min_{\mathbf{h} \in \mathcal{H}} g(\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t))$$

where $t \in [0, 1]$ and \mathcal{H} is a candidate space, g could be the energy cost, length of trajectory or other concerned. Especially when the cost g is convex, the optimal trajectory $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t)$ is unique, namely, $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t) = \arg \min_{\mathbf{h} \in \mathcal{H}} g(\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t))$.

Given a start point \mathbf{q}_0 and a goal point \mathbf{q}_m , the trajectory planning problem for a single robot is to find an optimal trajectory $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t) \in \mathcal{X}_{\text{free}}$ from \mathbf{q}_0 to \mathbf{q}_m . Consequently, given start points $\{\mathbf{q}_{0,i}\}$ for robots respectively in a start area \mathcal{C}_0 and a goal area \mathcal{C}_1 , the trajectory planning problem for swarm robotics in an obstacle-dense environment is to assign goal points $\mathbf{q}_{m,i}$ for all robots and then find optimal homotopic trajectories with minimal energy $\mathbf{h}^*((\mathbf{q}_{0,i}, \mathbf{q}_{m,i}), t)$ from all start points $\mathbf{q}_{0,i} \in \mathcal{C}_0$ to the goal

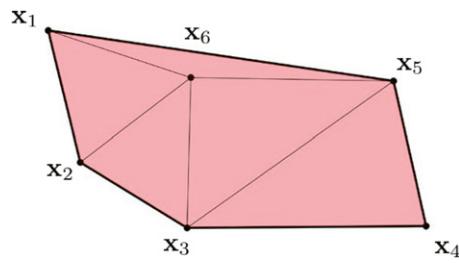


Figure 2. Convex hull example. For a set $\mathcal{C} = \{\mathbf{x}_i\} (i = 1, 2, \dots, 6)$, the pink area is the $\text{conv } \mathcal{C}$. The $\partial(\text{conv } \mathcal{C})$ and $\text{int}(\text{conv } \mathcal{C})$ denote the bold black lines and the pink area without the bold black lines, respectively.

points $\mathbf{q}_{m,i} \in \mathcal{C}_1$. And then, a control method is to be adopted to make the robot swarm move fast and smoothly as a group.

However, as the number of robots in a swarm increases, the computational cost of generating optimal trajectories also increases. Therefore, we use the optimal virtual tube to constrain the safety space where trajectories are located in and realize the generation of optimal trajectories with a low computational complexity of $O(n_t)$ in the following sections.

3. Optimal virtual tube

A virtual tube is constructed for swarm robotics, which could be used for swarm robotics flow. A generalization of a virtual tube is defined as a set in space, and some specific types of the virtual tube including the optimal virtual tube are introduced for trajectory planning of swarm robotics.

Tubes are often used for fluid transportation in three-dimension Euclidean space, which is denoted by \mathbb{R}^3 . However, it is not sufficient to describe a motion of a robot in three-dimension Euclidean space. For example, a set of all affine motions, including rotations and translations in \mathbb{R}^3 , is a six-dimension manifold. But this six-dimension manifold is not \mathbb{R}^6 . Since the virtual tube aims at swarm robotics, it is extended to be defined as a set in n -dimension space.

Intuitively, a virtual tube is constructed by defining maps between two bounded convex sets in space. The convex set is simply connected so that there are no “holes” in the set. The mathematical definition of the virtual tube is formulated in the following.

Definition 2. A virtual tube \mathcal{T} , as shown in Figure 3, is a set in n -dimension space represented by a 4-tuple $(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$ where

- $\mathcal{C}_0, \mathcal{C}_1$, called *terminals*, are disjoint bounded convex subsets in n -dimension space.
- \mathbf{f} is a *diffeomorphism*: $\mathcal{C}_0 \rightarrow \mathcal{C}_1$, so that there is a set of order pairs $\mathcal{P} = \{(\mathbf{q}_0, \mathbf{q}_m) | \mathbf{q}_0 \in \mathcal{C}_0, \mathbf{q}_m = \mathbf{f}(\mathbf{q}_0) \in \mathcal{C}_1\}$.
- \mathbf{h} is a smooth² map: $\mathcal{P} \times \mathcal{I} \rightarrow \mathcal{T}$ where $\mathcal{I} = [0, 1]$, such that $\mathcal{T} = \{\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t) | (\mathbf{q}_0, \mathbf{q}_m) \in \mathcal{P}, t \in \mathcal{I}\}$, $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), 0) = \mathbf{q}_0$, $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), 1) = \mathbf{q}_m$. The function $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t)$ is called a *trajectory* for an order pair $(\mathbf{q}_0, \mathbf{q}_m)$.

And, a cross-section \mathcal{C}_t of a virtual tube at $t \in \mathcal{I}$ is expressed as

$$\mathcal{C}_t = \{\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t) | (\mathbf{q}_0, \mathbf{q}_m) \in \mathcal{P}\}. \quad (3)$$

The surface of the virtual tube is the boundary of \mathcal{T} , defined as $\partial\mathcal{T}$.

In the following, some properties are directly obtained based on Definition 1.

Proposition 1. All cross-sections of a virtual tube are simply connected.

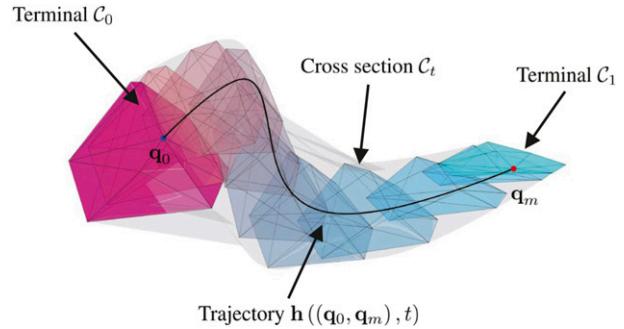


Figure 3. Virtual tube example. The purple and blue polyhedrons are terminals. The shaded polyhedrons are cross-sections \mathcal{C}_t . The black curve is a trajectory that is from \mathbf{q}_0 in terminal \mathcal{C}_0 to \mathbf{q}_m in terminal \mathcal{C}_1 . The gray area is the virtual tube.

Proof. The convex sets $\mathcal{C}_0, \mathcal{C}_1$ are simply connected. Thus, the order pair \mathcal{P} is simply connected. For a cross-section \mathcal{C}_{t_0} , there is a smooth map \mathbf{h} from $\mathcal{P} \times t_0$ to \mathcal{C}_{t_0} . Therefore, the cross-section is simply connected.

Proposition 2. All cross-sections of a virtual tube are continuous. That is, for every point in the cross-section, there exists at least one trajectory through it.

Proof. The domain $\mathcal{P} \times \mathcal{I}$ is continuous. And the map \mathbf{h} is continuous. So all cross-sections are continuous.

Remark 1. It should be noted that there is no explicit definition of a generator curve in Definition 2, compared with the definition of the virtual tube with a generator curve in our previous work (Mao and Quan (2022)). In the following, we will show that the previous definition is a special case of Definition 2. The virtual tube with a generator curve is defined as

$$\mathcal{T}(s, \theta, \rho) = \gamma(s) + \rho \lambda(s, \theta) (\mathbf{n}(s) \cos \theta + \mathbf{b}(s) \sin \theta),$$

where $s \in \mathcal{D}_s = [s_0, s_f] \subset \mathbb{R}$, $\rho \in \mathcal{D}_\rho = [\rho_{\min}(s), 1]$, $\theta \in \mathcal{D}_\theta \subset \mathbb{R}$, $\rho_{\min}(s)$ is a real-value function with respect to s , the second order differentiable curve $\gamma(s) \in \mathbb{R}^n$ is called generator curve, $\mathbf{n}(s) = \ddot{\gamma}(s) / \| \dot{\gamma}(s) \| - \dot{\gamma}(s) \cdot \ddot{\gamma}(s) / \| \dot{\gamma}(s) \|^3 \dot{\gamma}(s)$ and $\mathbf{b}(s) = \dot{\gamma}(s) / \| \dot{\gamma}(s) \| \times \mathbf{n}(s) \in \mathbb{R}^n$ are principle normal and binormal vectors of generator curve γ at the point $\gamma(s)$, respectively, and $\lambda(s, \theta) \in \mathbb{R}$ called radius is continuous with respect to s . Let terminals

$$\mathcal{C}_0 = \{\mathbf{q}_0 | \mathbf{q}_0 = \mathbf{f}_s(\theta, \rho) = \mathcal{T}(s_0, \theta, \rho), \theta \in \mathcal{D}_\theta, \rho \in \mathcal{D}_\rho\},$$

$$\mathcal{C}_1 = \{\mathbf{q}_m | \mathbf{q}_m = \mathbf{f}_f(\theta, \rho) = \mathcal{T}(s_f, \theta, \rho), \theta \in \mathcal{D}_\theta, \rho \in \mathcal{D}_\rho\}.$$

Thus, denote $\mathbf{f} = \mathbf{f}_f \circ \mathbf{f}_s^{-1}$ which is a diffeomorphism, and then, the set of order pairs \mathcal{P} is generated by \mathbf{f} . Let the trajectory be

$$\mathbf{h} \left((\mathcal{T}(s_0, \theta, \rho), \mathcal{T}(s_f, \theta, \rho)), \frac{s - s_0}{s_f - s_0} \right) = \mathcal{T}(s, \theta, \rho)$$

which is a smooth map. Therefore, this virtual tube with a generator curve is a special case of the virtual tube in

Definition 2. Here is a simple example. As shown in **Figure 4**, the generator curve $\gamma(s) = [s \ 0 \ 0]^T$, radius $\lambda(s, \theta) = 1$, normal unit vector $\mathbf{n}(s) = [0 \ 1 \ 0]^T$, bi-normal unit vector $\mathbf{b}(s) = [0 \ 0 \ 1]^T$, $s \in [0, 1]$, $\theta \in [0, 2\pi]$, $\rho \in [0, 1]$. Obviously, \mathcal{C}_0 and \mathcal{C}_1 are disjoint bounded convex subsets, and are called terminals. For any $\mathbf{q}_0 = [\rho \cos \theta \ \rho \sin \theta \ 0]^T \in \mathcal{C}_0$, the corresponding point in terminal \mathcal{C}_1 is $\mathbf{q}_m = \mathbf{f}(\mathbf{q}_0) = [1 \ \rho \cos \theta \ \rho \sin \theta]^T \in \mathcal{C}_1$, the set of order pairs \mathcal{P} is constructed. And, \mathbf{f} is a diffeomorphism, \mathbf{h} is a smooth map. Thus, this tube with a generator curve could be defined by *Definition 2*.

Some definitions which will be used are proposed in the following.

Definition 3. A virtual tube $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$ is *linear* if it satisfies following properties:

- The diffeomorphism \mathbf{f} is a *linear* map for any $\mathbf{q}_{0,k} \in \mathcal{C}_0$ and $\theta_k \in \mathbb{R}$, namely

$$\mathbf{f}\left(\sum_{k=1}^q \theta_k \mathbf{q}_{0,k}\right) = \sum_{k=1}^q \theta_k \mathbf{f}(\mathbf{q}_{0,k}).$$

- The smooth map \mathbf{h} is a *linear* map for any $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k}) \in \mathcal{P}$ and $\eta_k \in \mathbb{R}$, namely

$$\begin{aligned} \mathbf{h}\left(\sum_{k=1}^q \eta_k (\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t\right) \\ = \sum_{k=1}^q \eta_k \mathbf{h}((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t). \end{aligned}$$

Definition 4. A terminal \mathcal{C}_0 (or \mathcal{C}_1) is a *convex hull* of the finite set $\{\mathbf{q}_{0,k}\}$ if it can be expressed as

$$\mathcal{C}_0 = \left\{ \mathbf{q}_0 \middle| \mathbf{q}_0 = \sum_{k=1}^q \theta_k \mathbf{q}_{0,k}, \sum_{k=1}^q \theta_k = 1, \theta_k \geq 0 \right\}. \quad (4)$$

According to *Definition 1*, a trajectory \mathbf{h}^* for an order pair $(\mathbf{q}_0, \mathbf{q}_m)$ is optimal with respect to a cost function defined as $g(\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t))$. Consequently, the definition of an optimal virtual tube is proposed in the following.

Definition 5. A virtual tube \mathcal{T} is *optimal* with respect to a cost g if every trajectory in the tube is *optimal* with respect to a cost g , namely, $\mathcal{T} = (\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$.

As shown in **Figure 3**, if all trajectories in the tube are optimal, the tube is an optimal virtual tube.

4. A type of optimal virtual tubes: Linear virtual tube with convex hull terminals

There are infinite trajectories in the virtual tube, so it is impossible to solve optimization problems for every optimal trajectory, respectively. The first natural thought is to perform interpolation to reduce calculation burden significantly. There

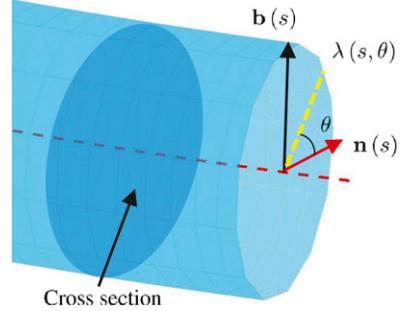


Figure 4. Virtual tube with a generator curve in \mathbb{R}^3 , where the red dotted line is the generator curve; the dark blue plane is a cross-section and the light blue surface is the virtual tube surface.

are two lemmas about conditions for optimality with proofs shown in [Appendix A](#), which are useful for constructing a type of optimal virtual tubes.

Lemma 1. Constrained with hyperplanes: Suppose that a convex optimization problem has the following type

$$\begin{array}{ll} \min & f_0(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \end{array} \quad (5)$$

where $f_0(\mathbf{x})$ is a convex function with respect to $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_n]^T$, $\mathbf{A} \in \mathbb{R}^{p \times (n+1)}$ ($p < n+1$), $\mathbf{b} \in \mathbb{R}^p$ and $\nabla f_0(\mathbf{x})$ is linear with respect to \mathbf{x} . The optimal solution \mathbf{x}_k is obtained when $\mathbf{b} = \mathbf{b}_k$, $k = 1, 2, \dots, q$. Denote $\mathcal{B} = \{\mathbf{b}_k\}$, $\mathcal{X} = \{\mathbf{x}_k\}$, $\mathbf{b}(\theta) = \sum_{k=1}^q \theta_k \mathbf{b}_k \in \text{conv } \mathcal{B}$. Then, $\mathbf{x}(\theta) = \sum_{k=1}^q \theta_k \mathbf{x}_k \in \text{conv } \mathcal{X}$ is feasible and optimal.

Lemma 2. Standard form: Suppose that a convex optimization problem has the following type

$$\begin{array}{ll} \min & f_0(\mathbf{x}) \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & f_i(\mathbf{x}) \leq 0, i = 1, \dots, n_c \end{array} \quad (6)$$

where $f_i(\mathbf{x}) (i = 0, 1, \dots, n_c)$ are convex functions with respect to $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_n]^T$, $\mathbf{A} \in \mathbb{R}^{p \times (n+1)}$ ($p < n+1$), $\mathbf{b} \in \mathbb{R}^p$ and $\nabla f_0(\mathbf{x})$ is linear with respect to \mathbf{x} . The optimal solution \mathbf{x}_k is obtained when $\mathbf{b} = \mathbf{b}_k$, $k = 1, 2, \dots, q$. Denote $\mathcal{B} = \{\mathbf{b}_k\}$, $\mathcal{X} = \{\mathbf{x}_k\}$, $\mathbf{b}(\theta) = \sum_{k=1}^q \theta_k \mathbf{b}_k \in \text{conv } \mathcal{B}$. Then, $\mathbf{x}(\theta) = \sum_{k=1}^q \theta_k \mathbf{x}_k \in \text{conv } \mathcal{X}$ is feasible and optimal.

Definition 6. A virtual tube $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$ is called a *linear virtual tube with convex hull terminals* if it satisfies:

- The terminal \mathcal{C}_0 is the convex hull of the finite set $\{\mathbf{q}_{0,k}\}$, and the terminal \mathcal{C}_1 is the convex hull of the finite set $\{\mathbf{q}_{m,k}\}$, $k = 1, 2, \dots, q$. The set $\{\mathbf{q}_{0,k}\}$ and the set $\{\mathbf{q}_{m,k}\}$ have the same number of elements.
- For any $\mathbf{q}_{0,k}$ in the set $\{\mathbf{q}_{0,k}\}$, $\mathbf{q}_{m,k} = \mathbf{f}(\mathbf{q}_{0,k})$.
- The virtual tube $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$ is *linear*

A linear virtual tube with convex hull terminals is a special case of virtual tubes, and the convex hull of the finite

set is a polyhedron (Boyd and Vandenberghe (2004)), as shown in Figure 5. An example of a linear virtual tube with convex hull terminals is shown in Figure 3.

Theorem 1. A linear virtual tube with convex hull terminals $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$ is optimal with respect to cost f_0 if (i) its trajectory for each order pair $(\mathbf{q}_0, \mathbf{q}_m)$ can be linearly parameterized as $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t) = \mathbf{C}(t)\mathbf{x}$, where $\mathbf{C}(t)$ is a matrix only respect to t , and \mathbf{x} is a parameter vector, (ii) the trajectories $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ for order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})(k = 1, 2, \dots, q)$ are optimal with respect to cost f_0 and satisfy conditions in (6) where $\mathbf{b} = \mathbf{b}_k$, $\mathbf{x} = \mathbf{x}_k$ for each trajectory $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$, (iii) the linear equality constraint is $\mathbf{Ax} = \sum_{k=1}^q \theta_k \mathbf{b}_k$ for each order pair $(\mathbf{q}_0, \mathbf{q}_m)$.

Proof. We show that, in Step 1, any trajectory $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t)$ in the linear virtual tube with convex hull terminals can be expressed as a convex combination of trajectories $\mathbf{h}((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$. Then, in Step 2, based on Lemma 2, we will show that any trajectory $\mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t)$ is optimal if the trajectories $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ for order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})(k = 1, 2, \dots, q)$ are optimal. Therefore, this virtual tube is optimal.

Step 1. Suppose that the terminal \mathcal{C}_0 is the convex hull of the finite set $\{\mathbf{q}_{0,k}\}$, such as

$$\mathcal{C}_0 = \left\{ \mathbf{q}_0 \middle| \mathbf{q}_0 = \sum_{k=1}^q \theta_k \mathbf{q}_{0,k}, \sum_{k=1}^q \theta_k = 1, \theta_k \geq 0 \right\},$$

where q is the number of the elements in set $\{\mathbf{q}_{0,k}\}$. Then q order pairs $\{(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})\}$ are constructed by

$$\mathbf{q}_{m,k} = \mathbf{f}(\mathbf{q}_{0,k}).$$

Thus, the set of order pairs \mathcal{P} is constructed by

$$\begin{aligned} \mathcal{P} &= \{(\mathbf{q}_0, \mathbf{f}(\mathbf{q}_0))\} \\ &= \left\{ \left(\sum_{k=1}^q \theta_k \mathbf{q}_{0,k}, \mathbf{f} \left(\sum_{k=1}^q \theta_k \mathbf{q}_{0,k} \right) \right) \right\} \\ &= \left\{ \left(\sum_{k=1}^q \theta_k \mathbf{q}_{0,k}, \sum_{k=1}^q \theta_k \mathbf{f}(\mathbf{q}_{0,k}) \right) \right\} \\ &= \left\{ \left(\sum_{k=1}^q \theta_k \mathbf{q}_{0,k}, \sum_{k=1}^q \theta_k \mathbf{q}_{m,k} \right) \right\} \\ &= \left\{ \sum_{k=1}^q \theta_k (\mathbf{q}_{0,k}, \mathbf{q}_{m,k}) \right\}. \end{aligned}$$

Any trajectory for the order pair $(\mathbf{q}_0, \mathbf{q}_m)$ in \mathcal{P} is expressed as

$$\begin{aligned} \mathbf{h}((\mathbf{q}_0, \mathbf{q}_m), t) &= \mathbf{h} \left(\sum_{k=1}^q \theta_k (\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t \right) \\ &= \sum_{k=1}^q \theta_k \mathbf{h}((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t). \end{aligned} \quad (7)$$

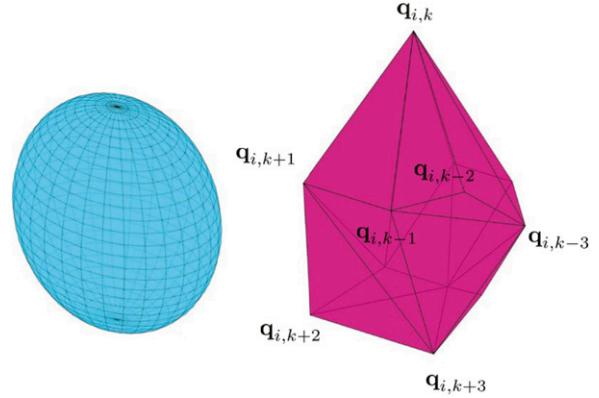


Figure 5. The blue sphere is a terminal which is a convex set. The purple polyhedron is a convex hull terminal. And they could also indicate the cross-section \mathcal{C}_t at t .

Step 2. We will show that any trajectory in the linear virtual tube with convex hull terminals is optimal, namely, this virtual tube is optimal.

Through solving optimization problem in (6) where $\mathbf{b} = \mathbf{b}_k$, the q optimal trajectories for order pairs $\{(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})\}$ are generated, which are linearly parameterized as

$$\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t) = \mathbf{C}(t)\mathbf{x}_k, k = 1, \dots, q, \quad (8)$$

where $\mathbf{C}(t)$ is a matrix with t , \mathbf{x}_k is the optimal solution. Because the linear constraint for each order pair $(\mathbf{q}_0, \mathbf{q}_m)$ is $\mathbf{Ax} = \sum_{k=1}^q \theta_k \mathbf{b}_k$, according to Lemma 2, $\mathbf{x} = \sum_{k=1}^q \theta_k \mathbf{x}_k$ is the optimal solution. Therefore, any trajectory for order pair $(\mathbf{q}_0, \mathbf{q}_m) \in \mathcal{P}$ is optimal which is expressed as

$$\begin{aligned} \mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t) &= \sum_{k=1}^q \theta_k \mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t) \\ &= \sum_{k=1}^q \theta_k \mathbf{C}(t)\mathbf{x}_k = \mathbf{C}(t) \sum_{k=1}^q \theta_k \mathbf{x}_k = \mathbf{C}(t)\mathbf{x}. \end{aligned}$$

Thus, this linear virtual tube with convex hull terminals $(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$ is optimal.

An example of the optimal linear virtual tube with convex hull terminals is given as the following.

Example 1. In two-dimension Euclidean space, suppose that there exists a linear virtual tube with convex hull terminals $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h})$. The terminals $\mathcal{C}_0, \mathcal{C}_1$ are expressed as the convex hull of the the set $\{\mathbf{q}_{0,1}, \mathbf{q}_{0,2}\}$ and $\{\mathbf{q}_{m,1}, \mathbf{q}_{m,2}\}$, namely, $\mathcal{C}_0 = \mathbf{q}_0(\theta) = (1 - \theta)\mathbf{q}_{0,1} + \theta\mathbf{q}_{0,2}$, $\mathcal{C}_1 = \mathbf{q}_m(\theta) = (1 - \theta)\mathbf{q}_{m,1} + \theta\mathbf{q}_{m,2}$, $\theta \in [0, 1]$. The smooth map \mathbf{f} is defined as $\mathbf{q}_{m,k} = \mathbf{f}(\mathbf{q}_{0,k})$, $k = 1, 2$. Thus, a set of order pairs is expressed as $\mathcal{P} = \{(\mathbf{q}_0(\theta), \mathbf{q}_m(\theta))\}$. Let $\mathbf{h}_0^*(t), \mathbf{h}_1^*(t)$ denote the optimal trajectories for order pairs $(\mathbf{q}_{0,1}, \mathbf{q}_{m,1})$ and $(\mathbf{q}_{0,2}, \mathbf{q}_{m,2})$, respectively, which are generated by solving optimization problems satisfying Lemma 2. And the linear equality constraint for $(\mathbf{q}_0(\theta), \mathbf{q}_m(\theta))$ satisfies condition (iii) in Theorem 1. Thus, the optimal trajectory

of any order pair $(\mathbf{q}_0(\theta), \mathbf{q}_m(\theta)) \in \mathcal{P}$ is expressed as $\mathbf{h}_\theta^*(t) = (1 - \theta)\mathbf{h}_0^*(t) + \theta\mathbf{h}_1^*(t)$. Since every trajectory in the virtual tube is optimal, $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$ is the optimal virtual tube, as shown in Figure 6.

5. A planning method to obtain an optimal virtual tube

Considering a robot swarm composed of M agents labeled by $j \in \mathcal{M} = \{1, \dots, M\}$, a start area which is the convex hull of set $\{\mathbf{q}_{0,k}\}$ is labeled by \mathcal{C}_0 , and a goal area which is the convex hull of set $\{\mathbf{q}_{m,k}\}$ is labeled by \mathcal{C}_1 . Suppose that the start position $\mathbf{q}_0(\theta_j) \in \mathcal{C}_0$ of robot j is given. An obstacle-dense environment is between the start area and the goal area. The objective is to construct an optimal virtual tube and make robots pass through the obstacle-dense environment in the optimal virtual tube.

5.1. Outline

For an optimal virtual tube $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$, the set of order pairs \mathcal{P} is constructed by \mathbf{f} first, after inputting terminals \mathcal{C}_0 and \mathcal{C}_1 . Then the paths for order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are found. Next, the optimal trajectories $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ for order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are generated by solving optimization problems. Finally, an optimal linear virtual tube with convex hull terminals $\mathcal{T}(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$ is constructed, with an infinite number of optimal trajectories. The whole process of optimal virtual tube planning is shown in Figure 7.

5.2. Constructing order pairs

Suppose that the start area and goal area called terminal \mathcal{C}_0 and \mathcal{C}_1 are the convex hull of $\{\mathbf{q}_{0,k}\}$ and $\{\mathbf{q}_{m,k}\}$, respectively, with the same number of extreme points (Adasch et al. (2006)), as shown in Step ① of Figure 7.

Since the order pairs in \mathcal{P} are constructed by a map \mathbf{f} , a selection strategy of order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$, namely, a suitable map \mathbf{f} , is simply described in the following. Suppose the number of extreme points of terminals is q . First, select the extreme points (or so called vertices) of the terminal \mathcal{C}_0 as the set of $\{\mathbf{q}_{0,k}\}(k = 1, 2, \dots, q)$. Then, the order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are constructed from a map \mathbf{f} , where $\mathbf{q}_{m,k} = \mathbf{f}(\mathbf{q}_{0,k})$ are also the vertices of terminal \mathcal{C}_1 and the sorting principle of $\mathbf{q}_{m,k}(k = 1, 2, \dots, q)$ is specified by a classic assignment problem (AP) (Pentico (2007)) in which the objective function minimizes both the total distance and the variance of distance between $\mathbf{q}_{0,k}$ and $\mathbf{q}_{m,k}$, as shown in Step ② of Figure 7. Then, the set of order pairs \mathcal{P} is constructed from the map \mathbf{f} .

Any start position $\mathbf{q}_0(\theta_j)$ in terminal \mathcal{C}_0 is expressed as

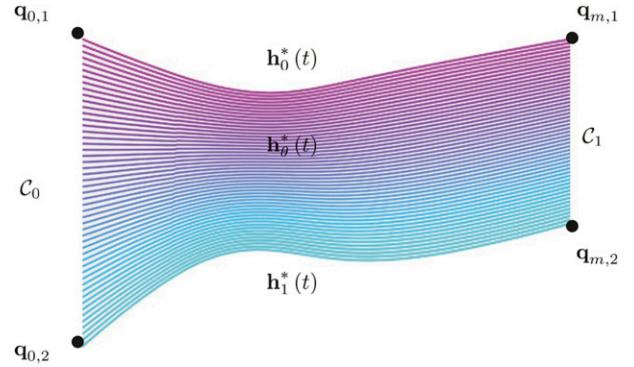


Figure 6. An optimal virtual tube example. Different colors represent different optimal trajectories. Curves $\mathbf{h}_0^*(t)$ and $\mathbf{h}_1^*(t)$ are optimal trajectories generated by solving optimization problems. And $\mathbf{h}_\theta^*(t)$ are optimal trajectories generated by interpolation.

$$\mathbf{q}_0(\theta_j) = \sum_{k=1}^q \theta_{kj} \mathbf{q}_{0,k}, \quad \sum_{k=1}^q \theta_{kj} = 1, \quad \theta_{kj} \geq 0, \quad j \in \mathbb{Z}^+, \quad (9)$$

where q is a finite number of elements in $\{\mathbf{q}_{0,k}\}$. Compared with the infinite number of start positions in terminal \mathcal{C}_0 , the number of elements in $\{\mathbf{q}_{0,k}\}$ is small enough. Let \mathbf{f} be the linear map between \mathcal{C}_0 and \mathcal{C}_1 so that for any goal position $\mathbf{q}_m(\theta_j) \in \mathcal{C}_1$,

$$\begin{aligned} \mathbf{q}_m(\theta_j) &= \mathbf{f}(\mathbf{q}_0(\theta_j)) = \mathbf{f}\left(\sum_{k=1}^q \theta_{kj} \mathbf{q}_{0,k}\right) = \sum_{k=1}^q \theta_{kj} \mathbf{f}(\mathbf{q}_{0,k}) \\ &= \sum_{k=1}^q \theta_{kj} \mathbf{q}_{m,k}. \end{aligned}$$

The set of order pairs \mathcal{P} , thus, has been constructed, such as $\{(\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j))\}$.

5.3. Finding paths

The paths for order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})(k = 1, \dots, q)$ are needed to be found after constructing order pairs. And a normalized parameterization of the paths is designed.

There are many path-finding methods used for finding waypoints in the obstacle environment such as RRT, RRT* (Karaman and Frazzoli (2011)), and A*. For each order pair $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})(k = 1, \dots, q)$, we select RRT* to find $m + 1$ waypoints $\{\mathbf{q}_{i,k}\}(i = 0, \dots, m)$ between $\mathbf{q}_{0,k}$ and $\mathbf{q}_{m,k}$.

To parameterize the waypoints $\{\mathbf{q}_{i,k}\}$, parameters $u_{i,k}$ are assigned to corresponding waypoint $\mathbf{q}_{i,k}$. Thus, a partition of u_k is first designed, called knots $\{u_{i,k}\}$. Then public knots $\{u_i\}$ and public normalized knots $\{t_k\}$ are generated by domain transformation. The chord length parameterization, a suitable method in most engineering applications (Piegl and Tiller (2000)), is used to generate knots $\{u_{i,k}\}$ expressed as

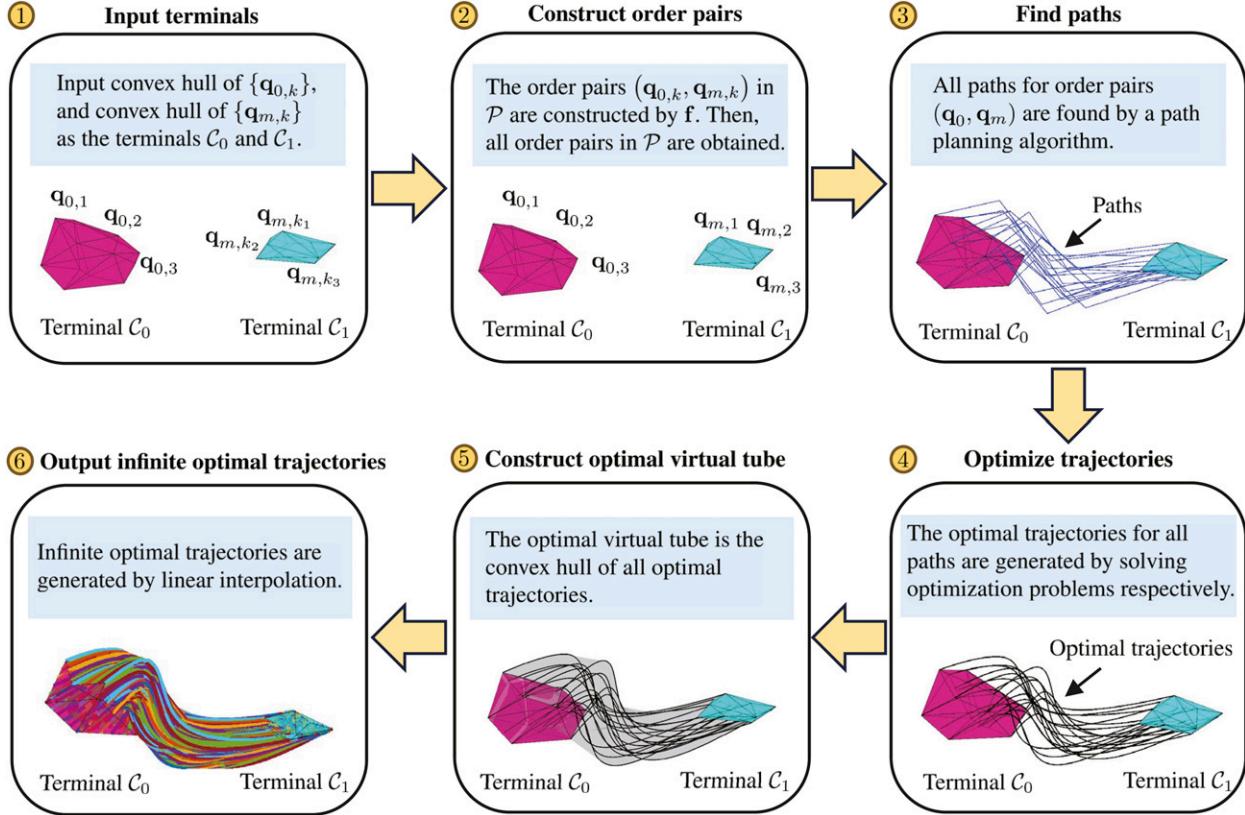


Figure 7. Optimal virtual tube planning process.

$$u_{i,k} = \begin{cases} 0, & i = 0, \\ u_{i-1,k} + \|\mathbf{q}_{i,k} - \mathbf{q}_{i-1,k}\|, & i = 1, 2, \dots, m. \end{cases} \quad (10)$$

It should be noted that the knots $\{u_{i,k}\}$ for each pair $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are generally not the same, because of variation in distances $\|\mathbf{q}_{i,k} - \mathbf{q}_{i-1,k}\|$, $k = 1, \dots, q$. Motivated by the parameterization of tensor product surface (Strothotte and Schlechtweg (2002)), all parameterizations of paths are expected to be the same for convenience. The public knots $\{u_i\}$ are the arithmetic mean of all knots, which are expressed as

$$u_i = \frac{\sum_{k=1}^q u_{i,k}}{q}. \quad (11)$$

To get the normalized parameterization for correspondence with *Definition 2*, the normalized knots $\{t_i\}$ are obtained by

$$t_i = \frac{u_i}{u_m}, i = 0, 1, \dots, m. \quad (12)$$

Connecting adjacent waypoints with straight lines, the RRT* path planner finds q collision-free paths for q order pairs $\{(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})\}$ respectively, as shown in Step ③ of Figure 7.

Remark 2. The numbers of the waypoints $\{\mathbf{q}_{i,k}\}$ of different order pairs are assumed as the same. Otherwise, different numbers of the waypoints $\{\mathbf{q}_{i,k}\}$ lead to different lengths of parameter vectors \mathbf{x} in *Theorem 1*, that would not satisfy *Lemmas 1,2*. Meanwhile, through shrinking the sample space around the first found path to eliminate any potential detours or alternate routes, all paths could be homotopic so that there is no obstacle between paths.

Remark 3. The scale of the swarm would be limited in an extremely cluttered environment. The swarm is not allowed to split apart to move around obstacles because of the homotopic paths. Thus, in an extremely cluttered environment, all paths need to pass through a narrow gap whose size is much smaller than that of the swarm, which would cause blockage of the movement of the swarm. To overcome this weak point, large swarms could be divided into several subgroups satisfying the minimum size of gaps to plan virtual tubes.

5.4. Optimizing trajectories

The original paths in the above subsection have some sharp corners, as shown in Step ③ of Figure 7, which are not feasible considering the dynamics of the robots. It is necessary to improve the smoothness of the original paths, as shown in Step ④ of Figure 7.

5.4.1. Trajectory representation. There are many ways to represent trajectories linearly, such as B-spline and polynomials. It is convenient to use n -th order piecewise polynomial to represent the trajectory \mathbf{h} , such as

$$\mathbf{h}(t) = \begin{cases} \sum_{i=0}^n \mathbf{a}_{i,1} t^i & t_0 < t < t_1 \\ \sum_{i=0}^n \mathbf{a}_{i,2} t^i & t_1 < t < t_2 \\ \vdots & \vdots \\ \sum_{i=0}^n \mathbf{a}_{i,m} t^i & t_{m-1} < t < t_m \end{cases} \quad (13)$$

where $\mathbf{a}_{i,j} = [a_{1,i,j} \ a_{2,i,j} \ \dots \ a_{d,i,j}]^T \in \mathbb{R}^d, j = 1, 2, 3, \dots, m$. For any $t \in [t_{k-1}, t_k]$, the trajectory $\mathbf{h}(t)$ could be represented by matrix

$$\mathbf{h}_j(t) = \mathbf{C}(t)\mathbf{x}_j \quad (14)$$

where $\mathbf{C}(t) = [\mathbf{I}_d \ t\mathbf{I}_d \ t^2\mathbf{I}_d \ \dots \ t^n\mathbf{I}_d]$,

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{a}_{0,j} \\ \mathbf{a}_{1,j} \\ \mathbf{a}_{2,j} \\ \vdots \\ \mathbf{a}_{n,j} \end{bmatrix}.$$

The representation of trajectory (14) is convenient to be used in quadratic programming (QP) in the following.

5.4.2. Constraints. The constraints, including the terminal conditions and intermediate conditions, are constructed as linear equations for smoothness. In terms of obstacle avoidance, adding corridor constraints could avoid collisions.

The terminal conditions are expressed as

$$\begin{aligned} \mathbf{h}_1(t_0) &= \mathbf{C}(t_0)\mathbf{x}_1 = \mathbf{q}_{0,k}, \\ \mathbf{h}_m(t_m) &= \mathbf{C}(t_m)\mathbf{x}_m = \mathbf{q}_{m,k}, \\ \frac{d^p \mathbf{h}_1(t)}{dt^p} \Big|_{t=t_0} &= \mathbf{q}_{0,k}^{(p)}, \\ \frac{d^p \mathbf{h}_m(t)}{dt^p} \Big|_{t=t_m} &= \mathbf{q}_{m,k}^{(p)}, \end{aligned} \quad (15)$$

where $\mathbf{q}_{0,k}^{(p)}$ and $\mathbf{q}_{m,k}^{(p)}$ are the p -order conditions in the start and goal points, terminals of the trajectory are constrained in start and goal points. And the intermediate conditions are expressed as

$$\frac{d^p \mathbf{h}_i(t)}{dt^p} \Big|_{t=t_i} = \frac{d^p \mathbf{h}_{i+1}(t)}{dt^p} \Big|_{t=t_i}, \quad (16)$$

where $p = 0, 1, 2, \dots, k_r, i = 1, 2, \dots, m - 1$,

$$\frac{d^p \mathbf{h}_i(t)}{dt^p} \Big|_{t=t_i} = \frac{d^p \mathbf{C}(t)}{dt^p} \Big|_{t=t_i} \mathbf{x}_i,$$

$$\frac{d^p \mathbf{h}_{i+1}(t)}{dt^p} \Big|_{t=t_i} = \frac{d^p \mathbf{C}(t)}{dt^p} \Big|_{t=t_i} \mathbf{x}_{i+1}.$$

Specifically, when $p = 0$, the terminals of each segment are waypoints:

$$\mathbf{h}_i(t_i) = \mathbf{h}_{i+1}(t_i) = \mathbf{q}_{i,k}. \quad (17)$$

For convenience, we denote

$$\mathbf{C}_t^{(p)} = \frac{d^p \mathbf{C}(t)}{dt^p}$$

which could transform (16) into

$$[\mathbf{C}_{t_i}^{(p)} \ -\mathbf{C}_{t_i}^{(p)}] \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_{i+1} \end{bmatrix} = \mathbf{0}. \quad (18)$$

Based on (15), (17), and (18), the linear equality constraints are derived.

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}_k = \mathbf{b}_k, \quad (19)$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{bmatrix}, \mathbf{b}_1 = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \mathbf{b}_2 = \begin{bmatrix} \mathbf{q}_{0,k} \\ \mathbf{q}_{1,k} \\ \vdots \\ \mathbf{q}_{m-1,k} \\ \mathbf{q}_{m,k} \end{bmatrix},$$

$$\mathbf{b}_3 = [\mathbf{q}_{0,k}^{(p)} \ \dots \ \mathbf{q}_{0,k}^{(1)} \ \mathbf{q}_{m,k}^{(p)} \ \dots \ \mathbf{q}_{m,k}^{(1)}]^T,$$

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{C}_{t_1}^{(p)} & -\mathbf{C}_{t_1}^{(p)} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{t_{m-1}}^{(p)} & -\mathbf{C}_{t_{m-1}}^{(p)} \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{C}_{t_0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{t_1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{t_{m-1}} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{t_m} \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} \mathbf{C}_{t_0}^{(p)} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_{t_0}^{(1)} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_{t_m}^{(p)} \\ \vdots & \vdots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_{t_m}^{(1)} \end{bmatrix}.$$

To avoid collision between trajectories and obstacles, corridor constraints (Mellinger and Kumar (2011)) are added by constraining n_j intermediate points in the segment from $\mathbf{q}_{i,k}$ to $\mathbf{q}_{i+1,k}$ ($i = 0, 1, 2, \dots, m - 1$).

For an intermediate point of trajectory $\mathbf{h}(s_j)$ in the segment from $\mathbf{q}_{i,k}$ to $\mathbf{q}_{i+1,k}$, the perpendicular distance vector between $\mathbf{h}(s_j)$ and the segment is expressed as

$$\mathbf{d}_i(s_j) = (\mathbf{h}(s_j) - \mathbf{q}_{i,k}) - ((\mathbf{h}(s_j) - \mathbf{q}_{i,k})^T \mathbf{t}_i) \mathbf{t}_i,$$

where \mathbf{t}_i is the unit vector along segment from $\mathbf{q}_{i,k}$ to $\mathbf{q}_{i+1,k}$. Thus, the convex constraints are expressed as

$$f_i(\mathbf{x}) = \|\mathbf{d}_i(s_j)\|_\infty - \delta_i \leq 0, \quad (20)$$

where $s_j = t_i + j(t_{i+1} - t_i)/(1 + n_c)$, $j = 1, \dots, n_c$, δ_i is the corridor width, $\mathbf{d}_i(s_j)$ is the perpendicular distance vector.

5.4.3. Cost function. The optimization problem to minimize energy cost could be formulated as a QP by regarding \mathbf{x} in (19) as variable vectors (Mellinger and Kumar (2011)). The cost function is expressed as

$$\begin{aligned} E_s &= \int_{t_0}^{t_m} \left\| \frac{d^{k_r} \mathbf{h}(t)}{dt^{k_r}} \right\|^2 dt \\ &= \sum_{i=1}^m \mathbf{x}_i^T \int_{t_{i-1}}^{t_i} \mathbf{C}_t^{(k_r)T} \mathbf{C}_t^{(k_r)} dt \mathbf{x}_i \\ &= \mathbf{x}^T \mathbf{H} \mathbf{x}. \end{aligned} \quad (21)$$

5.4.4. Optimization problem. The goal of the optimization problem is to minimize the energy cost meanwhile satisfying constraints. Thus, combining (19), (20), and (21), this problem could be expressed as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T \mathbf{H} \mathbf{x} \\ \text{s. t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & f_i(\mathbf{x}) \leq 0, i = 1, \dots, n_c, \end{aligned} \quad (22)$$

which is in the form concerned with Lemma 2. Therefore, the optimal trajectories $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ are derived by solving q optimization problems for q order pairs $\{(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})\}$.

5.5. Constructing optimal virtual tube

The waypoints for any order pair $(\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j))$ could be expressed as

$$\mathbf{q}_i(\theta_j) = \sum_{k=1}^q \theta_{kj} \mathbf{q}_{i,k}, i = 0, 1, 2, \dots, m.$$

And the normalized knots $\{t_i\}$ are used to parameterize the waypoints. Thus, the linear equality constraints are expressed as

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \mathbf{x} = \sum_{k=1}^q \theta_{kj} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}_k. \quad (23)$$

According to *Theorem 1*, the linear virtual tube with convex hull terminals $(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$ is optimal, as shown in Step ⑤ of [Figure 7](#). In other words, for any start point $\mathbf{q}_0(\theta_j)$, a goal point $\mathbf{q}_m(\theta_j)$ is assigned, and then the optimal trajectory $\mathbf{h}^*((\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j)), t)$ is expressed as

$$\mathbf{h}^*((\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j)), t) = \sum_{k=1}^q \theta_{kj} \mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t), \quad (24)$$

where $\sum_{k=1}^q \theta_{kj} = 1, \theta_{kj} \geq 0$.

For all robots at each t' , the desired positions are $\{\mathbf{h}^*((\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j)), t')\} (j \in \mathcal{M})$ which are within the convex hull of $\{\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t')\} (k = 1, 2, \dots, q)$. In other words, the swarm would approach a convex hull at each t' .

Remark 4. The optimal virtual tube has similar effects to the formation constraint. The swarm approaches a desired convex hull or called graph \mathcal{C}_t at each time t . However, the distance constraint of adjacent robots is not considered in tube planning as it is in the process of controller design, which leaves more flexibility for the formation of the swarm, compared with the formation constraint.

5.6. Complexity analysis

In this subsection, we analyze the time complexity of the proposed method. Suppose the q optimal trajectories $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ have been generated before. And according to (13), the number of parameters in trajectory $\mathbf{h}^*((\mathbf{q}_{0,k}, \mathbf{q}_{m,k}), t)$ is $n_t = (n+1)md$ where n is the orders of trajectory, m is the number of the segments and d is the number of dimension. The operation of the convex combination in (24) only has some basic operations including addition and multiplication on the n_t -dimension vectors. Therefore, time complexity of our method is $O(n_t)$.

As for solving optimization problem to generate the optimal trajectory, the time complexity is both related to the number of iterations and the time complexity of each iteration.³ Furthermore, the number of iterations is related to the precision ε and the dimension n_t , which could be expressed as $k(n_t, \varepsilon)$. Therefore, the time

complexity of solving optimization problem is normally from $O(k(n_t, \varepsilon)n_t^2)$ to $O(k(n_t, \varepsilon)n_t^3)$ (Wright (1997)). For example, the best-known interior-point method algorithm require $O(k(n_t, \varepsilon)) = O(\sqrt{n_t} \log(1/\varepsilon))$ iterations and the complexity of each iteration is roughly $O(n_t^3)$. Thus, the complexity of the interior-point method is $O(n_t^{3.5} \log(1/\varepsilon))$. Compared with solving optimization problem, our method effectively reduces the computations by removing the iteration and matrix operations.

6. Model predictive control for tracking trajectories in an optimal virtual tube

The optimal virtual tube is applicable to a wide range of control methods. Specifically, the optimal virtual tube can provide the desired trajectories, which makes the swarm exhibit an effect similar to the formation control by using the optimal control methods such as Model Predictive Control (MPC). Or it can also provide desired velocity command to a swarm based on the control method such as Artificial Potential Field (APF) and Control Barrier Function (CBF) for smoother control.

To easily understand and verify the characteristics of the optimal virtual tube, an application in control for the optimal virtual tube is provided. A hierarchical approach is employed to enable swarm robotics to move within the optimal virtual tube, comprising two layers: *optimal virtual tube planning* and *trajectory tracking* for a robot swarm. First, the optimal virtual tube planning is proposed in the above section, which generates optimal trajectories for the swarm. Subsequently, each robot tracks its own trajectory while avoiding collisions with other robots. This section first introduces a robot model and then designs an MPC controller to realize trajectory tracking and conflict avoidance.

6.1. Robot model

The dynamic of a holonomic robot is described as a mass point model:

$$\begin{aligned}\dot{\mathbf{p}}_i &= \mathbf{v}_i, \\ \dot{\mathbf{v}}_i &= \mathbf{u}_i,\end{aligned}\quad (25)$$

where $\mathbf{p}_i \in \mathbb{R}^2(\mathbb{R}^3)$ and $\mathbf{v}_i \in \mathbb{R}^2(\mathbb{R}^3)$ are the position and velocity of the i th robot, $\mathbf{u}_i \in \mathbb{R}^2(\mathbb{R}^3)$ is the acceleration input to the i th robot, $i = 1, 2, \dots, M$.

This system (25) could be expressed as

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (26)$$

where $\mathbf{x} = [\mathbf{p}_i \ \mathbf{v}_i]^T$, $\mathbf{u} = \mathbf{u}_i$. And the discretized form is

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \quad (27)$$

where \mathbf{x}_k and \mathbf{u}_k are state and input at the time k ,

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{I} & T_s \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{B}_k = \begin{bmatrix} \mathbf{0} \\ T_s \mathbf{I} \end{bmatrix}.$$

6.2. Dynamic obstacles

As depicted in Figure 8(a), for a robot i in the swarm, other robots j is regarded as the dynamic obstacles which are represented by an ellipse of area $\mathcal{O}_{j,k}$ at time k . The ellipse of area $\mathcal{O}_{j,k}$ at time k is denoted by

$$\mathcal{O}_{j,k} = \{\mathbf{p} \mid \|E(\mathbf{p} - \mathbf{p}_{j,k})\| \leq 1\}, k = 0, 1, \dots, N, \quad (28)$$

where $\mathbf{p}_{j,k}$ is the position of the center mass of robot j at time k , $E = \text{diag}(a_j, b_j, c_j)$ is an invertible scaling matrix to bound the boundary of the ellipse. The details of designing E can be found in Quan et al. (2023a). If the robot i is regarded as a mass point, the obstacle space \mathcal{O}_k of robot i at time k is expressed as

$$\mathcal{O}_k = \bigcup_{j \in \mathcal{M}, j \neq i} \mathcal{O}_{j,k}, \quad (29)$$

where $\mathcal{O}_{ij,k} = \{\mathbf{p} \mid \mathbf{p} \in \mathcal{O}_{i,k} \oplus \mathcal{O}_{j,k}\}$, \oplus is the Minkowski sum. However, when the robot i does not collide with other robots, the avoidance constraints

$$\mathbf{p}_{i,k} \cap \mathcal{O}_k = \emptyset \quad (30)$$

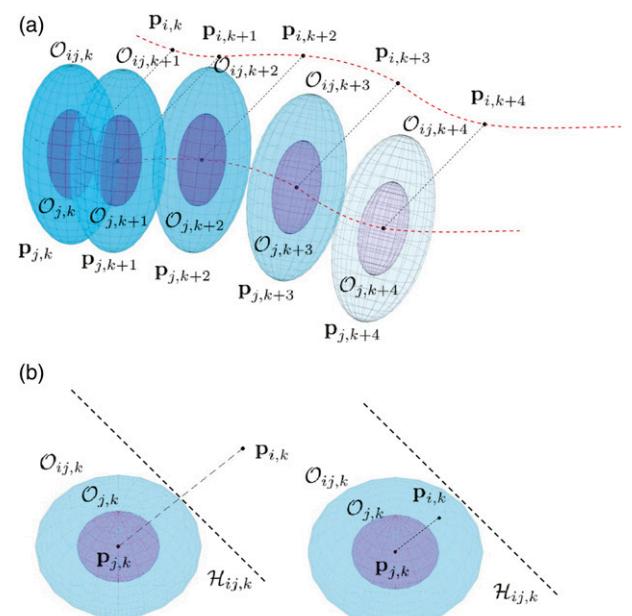


Figure 8. Constructing constraints for avoiding collisions. The purple ellipses $\mathcal{O}_{j,k+n}$ ($n = 0, 1, \dots, 4$) are boundaries of robot j at time $k+n$. The blue ellipses $\mathcal{O}_{i,k+n}$ ($n = 0, 1, \dots, 4$) are obstacle space for robot i at time $k+n$. (a) The obstacle constraints for robot i at time $k+n$. (b) Two situations: collision and avoidance. The dotted lines are tangent planes $H_{ij,k}$ of points which are intersections of surfaces of blue ellipses and connections of points $\mathbf{p}_{j,k}$, $\mathbf{p}_{i,k}$.

are not convex constraints. Thus, half spaces $\mathcal{H}_{j,k}$, as shown in Figure 8(b), are used to construct affine inequality constraints at each time k . The constraints (30) are transformed into

$$\mathcal{O} = \{\mathbf{x}_k | \mathbf{H}_k \mathbf{x}_k \leq h_k + s_k, k = 0, 1, \dots, N\}, \quad (31)$$

where \mathbf{x}_k are states of robot i , \mathbf{H}_k , h_k are parameters of half space constraints, s_k is a positive relaxation variable to avoid infeasibility problem in practice.

6.3. Optimization problem

The distance to desired state $\mathbf{x}_{d,k} = [\mathbf{p}_{d,k} \ \mathbf{v}_{d,k}]^T$ and desired input $\mathbf{u}_{d,k}$ at time k generated by desired trajectory $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t(s))$ ⁴ is minimized so that the robot could track the desired trajectory. Thus, the model predictive control problem containing robot model and other constraints in $N + 1$ steps is formulated as

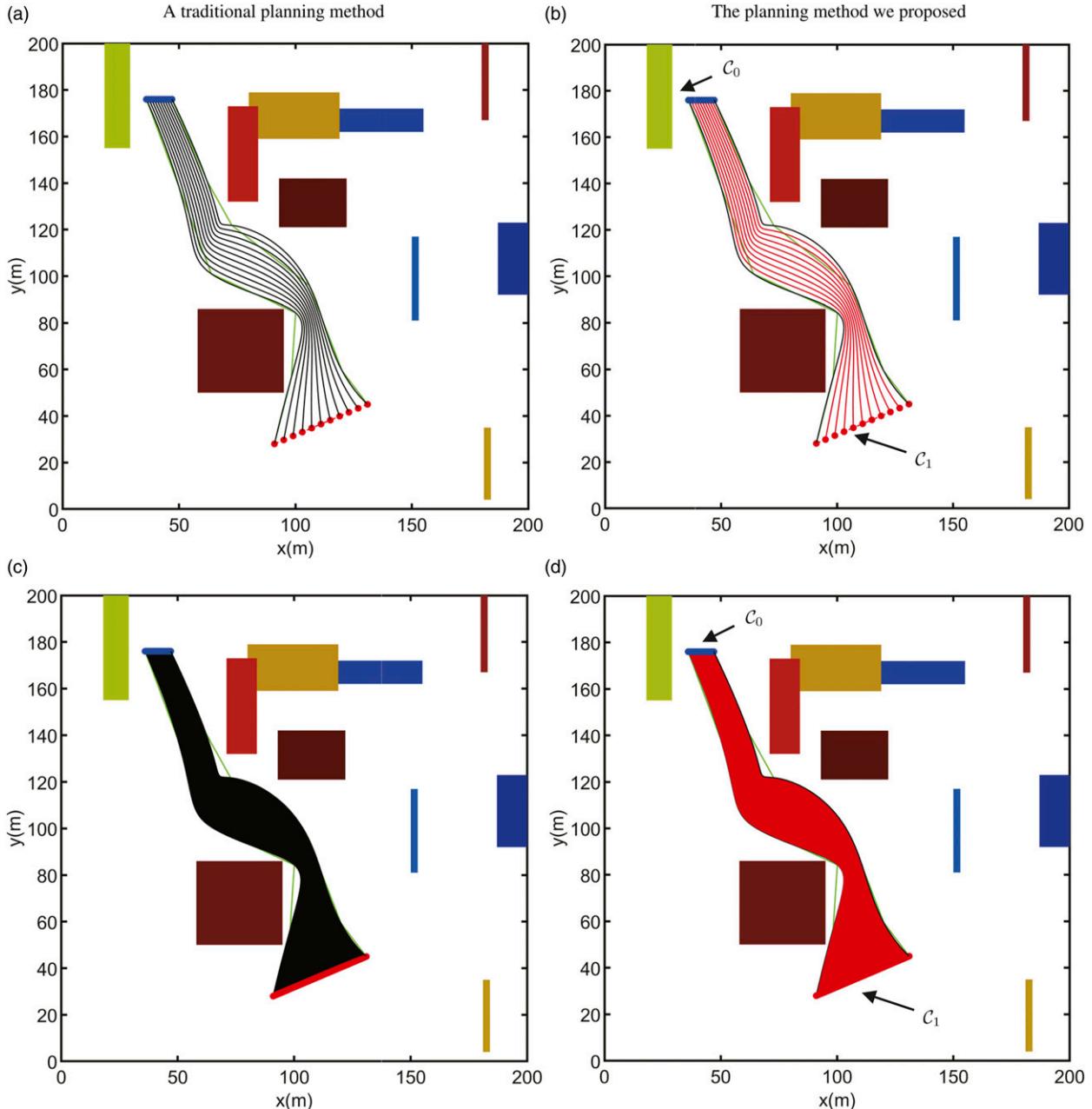


Figure 9. Comparison between the traditional method and the optimal virtual tube planning. (a) Trajectories planned by 11 different optimization problems with the traditional method. (b) Trajectories are planned only by two optimization problems with optimal virtual tube planning. (c) Trajectories planned by 101 different optimization problems with the traditional method. (d) Trajectories are planned only by two optimization problems with optimal virtual tube planning. The colorful rectangles are obstacles. The blue points and the red points are start points and goal points. The optimal trajectories planned by optimization problems are depicted in black, while the optimal trajectories obtained by interpolation are shown in red. The magenta dotted lines denote the terminals.

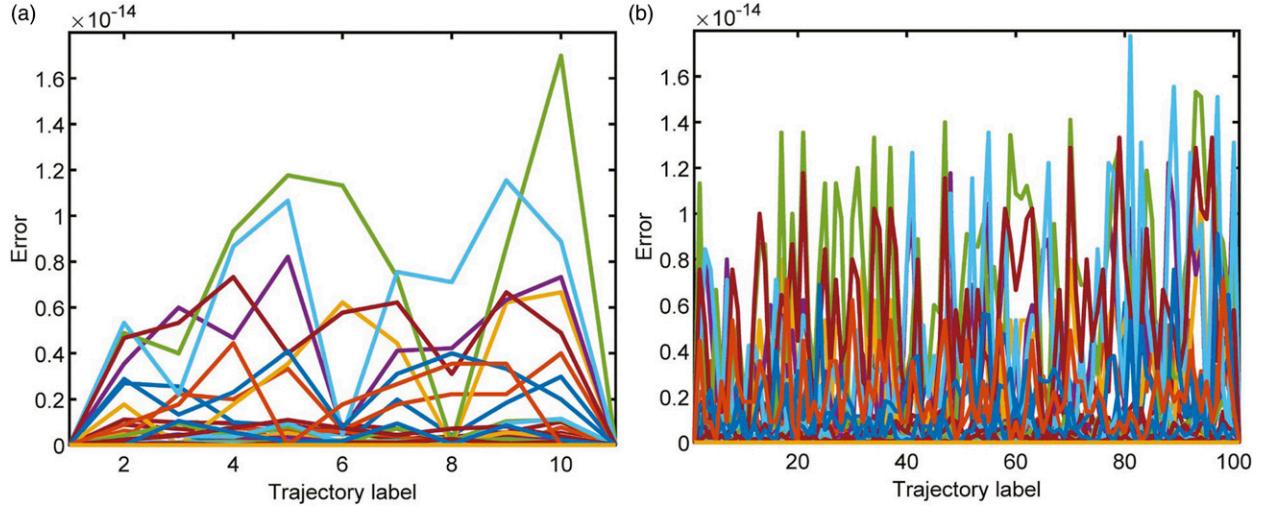


Figure 10. Errors between variable vectors by interpolation and optimization program are illustrated using colorful curves representing the error in each parameter of the variable vectors. (a) The errors of 11 trajectories. (b) The errors of 101 trajectories.

$$\begin{aligned} \min_{\mathbf{x}_k, \mathbf{u}_k} \quad & \tilde{\mathbf{x}}_N^T \mathbf{Q}_{x,N} \tilde{\mathbf{x}}_N + \sum_{k=0}^N q_{s,k} s_k^2 + \sum_{k=0}^{N-1} \tilde{\mathbf{x}}_k^T \mathbf{Q}_{x,k} \tilde{\mathbf{x}}_k + \\ \text{s.t.} \quad & \tilde{\mathbf{u}}_k^T \mathbf{Q}_{u,k} \tilde{\mathbf{u}}_k \\ & \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \\ & \mathbf{x}_k \in \mathcal{X}_t \cap \mathcal{O}, \\ & \mathbf{u}_k \in \mathcal{U}, \\ & k = 0, 1, \dots, N, \end{aligned} \quad (32)$$

where $\tilde{\mathbf{x}}_k = \mathbf{x}_{d,k} - \mathbf{x}_k$, $\tilde{\mathbf{u}}_k = \mathbf{u}_{d,k} - \mathbf{u}_k$, $\mathbf{Q}_{x,k}$ and $\mathbf{Q}_{u,k}$ are positive definite coefficient matrixs, \mathcal{X}_t is the feasible set of \mathbf{x}_k , \mathcal{U} is the feasible set of \mathbf{u}_k . The feasible set of \mathbf{v}_k in \mathcal{X}_t satisfies the dynamic constraints of robots. And the feasible set of \mathbf{p}_k in \mathcal{X}_t is the interior of the virtual tube $\text{int}(\mathcal{T})$ for the robot within the virtual tube, and $\{\mathbf{p}_k | \|\mathbf{p}_k - \mathbf{p}_{d,k}\| \leq \varepsilon_c \in \mathbb{R}^+\}$ for the robot on the boundary of the virtual tube $\partial\mathcal{T}$ where ε_c is a feasible constant.

7. Simulations and experiments

This section presents simulations and experiments of the drone swarm in simple scenarios to demonstrate the effectiveness of the proposed optimal virtual tube planning method and the MPC controller. Comparisons with existing methods including multi-robots formation control (Alonso-Mora et al. (2017)) and predictive control of aerial swarm (Soria et al. (2021)) are made in complex scenarios. The optimal virtual tube planning and trajectory tracking in simulations are implemented using MATLAB code, and executed on a PC with Intel Core i7-7700 @ 2.8GHz CPU and 16G RAM. The safety distance between drones is 1m. A video of simulations and experiments is available on <https://youtu.be/9pT5SiCsZis>.

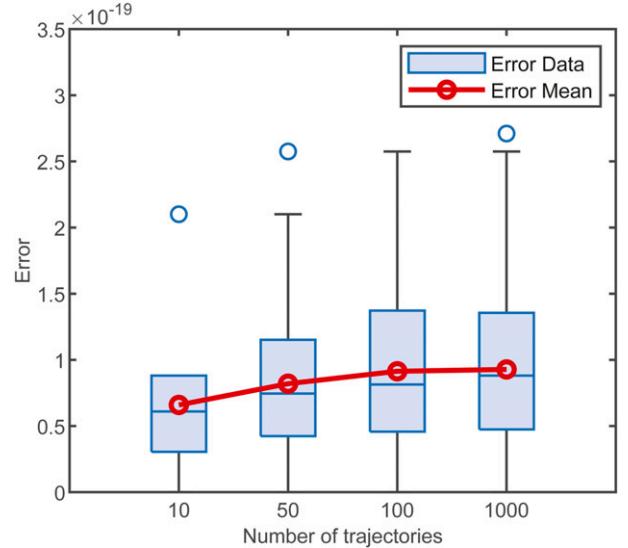


Figure 11. Error distribution of a parameter in variable vector.

7.1. Simulations in simple scenarios

The results of the optimal virtual tube planning and trajectory tracking are obtained. To demonstrate the effectiveness of the proposed method, comparisons between the optimal virtual tube planning and a traditional planning method are made in 2-D and 3-D space. The simulation results demonstrate that the proposed method with a computational complexity $O(n_t)$ produces the same results as the traditional planning method with a computational complexity from $O(k(n_t, \varepsilon)n_t^2)$ to $O(k(n_t, \varepsilon)n_t^3)$ while requiring less computational effort.

7.1.1. A 2-D optimal virtual tube. The optimal virtual tube planning is first validated in a 2-D map. Two methods were set with identical terminals C_0 and C_1 , as illustrated in Figure 9 with dotted lines. Additionally, 11 start positions

and 11 goal positions are also the same. For the traditional planning method, 11 separate optimization problems are required to be solved, as depicted in [Figure 9\(a\)](#). In contrast, for the proposed method, only two trajectories need to be planned: black curves in [Figure 9\(b\)](#). The remaining 9 trajectories, depicted by red curves in [Figure 9\(b\)](#), are generated by interpolation of the two planned trajectories. Similarly, 101 trajectories are generated in the same way, requiring the resolution of only two optimization problems for two trajectories. Comparing the traditional method with the proposed method, the errors were found to be negligible, with magnitudes less than 1.8×10^{-14} m, as shown in

[Figure 10](#). Thus, this outcome validates *Theorem 1*. To demonstrate that the error is independent of the number of interpolations, the variable vector is analyzed and the distribution of one parameter in the variable vector is shown with respect to the number of trajectories obtained by interpolation in [Figure 11](#). The results show that the error for this parameter remains stable as the number of interpolations increases from 10 to 1000.

In [Figure 9\(b\)](#), the terminal \mathcal{C}_0 is represented by a line with a length of 10m. To validate the controller, 11 drones are tightly placed in the terminal \mathcal{C}_0 , as shown in [Figure 12](#). Then, 11 optimal trajectories are assigned to each drone. In

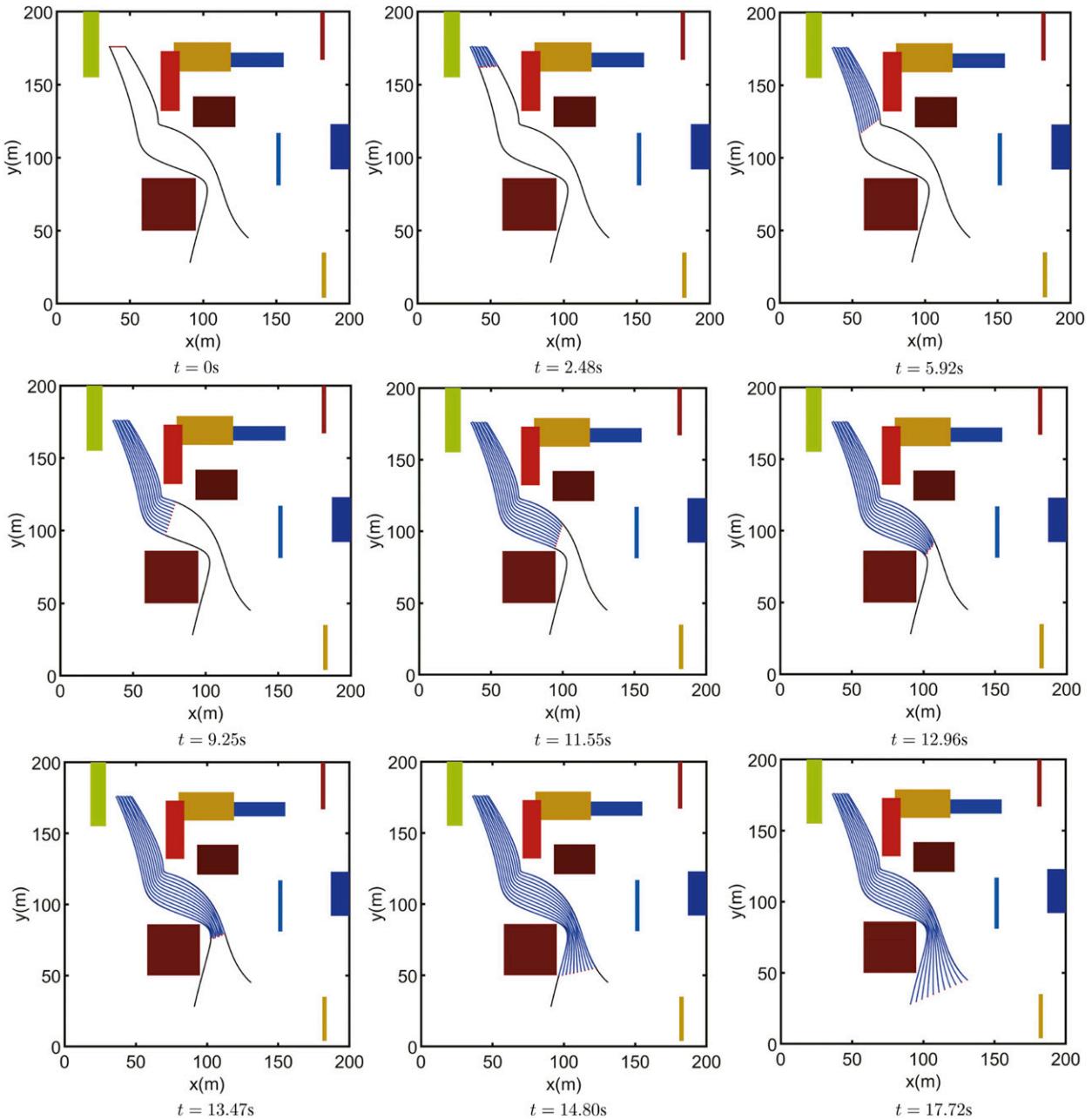


Figure 12. Simulation results of drone swarm trajectory tracking. The 11 dark blue lines are trajectories for 11 drones, while the red circles indicate drones.

the result of the simulation depicted in [Figure 12](#), the drone swarm safely and smoothly arrives at terminal C_1 . When the drones are far apart, they follow their own trajectories. However, when the swarm passes through a narrow cross-section of the virtual tube in [Figure 12](#), some drones need to avoid nearby drones, which may cause them to deviate from their trajectories ($t = 12.96\text{s}$) but return to planned trajectories after passing through the narrow space ($t = 14.80\text{s}$). The minimum distance among drones in the swarm with respect to the time shown in [Figure 13](#) is always greater than the safety distance 1m , which indicates that there is no collision among drones. To demonstrate the accuracy of drone tracking, errors between the desired and true trajectory of all drones are depicted in [Figure 14](#). Therefore, the simulation demonstrates that every drone can track the trajectory with high accuracy while avoiding other drones.

7.1.2. A 3-D optimal virtual tube. For a $240\text{ m} \times 240\text{ m} \times 120\text{ m}$ map, illustrated in [Figure 15\(a\)](#), the start terminal C_0 which is the convex hull of the set $\{\mathbf{q}_{0,k}\}$ is represented as a green tetrahedron, while the goal terminal C_1 which is the convex hull of the set $\{\mathbf{q}_{m,k}\}$ is represented as a red tetrahedron. First, the order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are constructed by assigning the red points to green points, respectively. Thus, any start point $\mathbf{q}_0(\theta_j)$ in terminal C_0 is expressed as (9), and the map \mathbf{f} maps this point to the goal point C_0 . Then, a set of order pairs \mathcal{P} is constructed as $\{(\mathbf{q}_0(\theta_j), \mathbf{q}_m(\theta_j))\}$. Next, the paths of the order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are generated by a tree-based motion planner called RRT*, as shown in [Figure 15\(b\)](#). The optimal trajectories \mathbf{h}^* of the order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are then obtained by solving optimization problems, as depicted in [Figure 15\(c\)](#). Finally, the optimal virtual tube is generated based on *Theorem 1*, which is illustrated in [Figure 15\(d\)](#).

The drone swarm comprises 84 drones, and their start and goal points are generated by equispaced interpolation. Subsequently, the 84 optimal trajectories for all drones are generated by interpolation between four optimal trajectories. The

error between the trajectories by interpolation and by solving 84 optimization problems, as shown in [Figure 16](#), is negligible. Then, drones could track their trajectories correspondingly.

The terminals C_0 and C_1 in [Figure 15\(a\)](#) are tetrahedra. The drone swarm with 84 drones is placed in the terminal C_0 . There are only 4 trajectories being planned and other 80 trajectories are generated by interpolation. In the simulation result, as shown in [Figure 17](#), the drone swarm arrives at terminal C_1 safely and smoothly. The minimum distance among drones in the swarm with respect to the time shown in [Figure 18](#) is always higher than the safety distance, which demonstrates no collision among drones. Errors between the desired and actual trajectory of drone swarm are depicted in [Figure 19](#).

7.2. Comparisons in complex scenarios

In this subsection, we assess the performance of the proposed method in terms of navigation efficiency in a complex scenario. We first present several performance metrics and then compare different methods in complex scenarios.

7.2.1. Performance metrics. We use the following performance metrics to compare our proposed method with other methods.

- Average time: The average of the arrival times of robots in the swarm to the goal area. If any robot in the swarm does not reach the goal area within a certain time limit, the average time is set to be infinite.
- Arrival rate: The ratio of the number of robots reaching the goal area to the total number of robots in a certain time limit.
- Average speed: The average speed of the robots reaching the goal area in a certain time limit. The other robots not reaching the goal area are not considered.

We run each method 30 times for each scenario to report the mean for each metric.

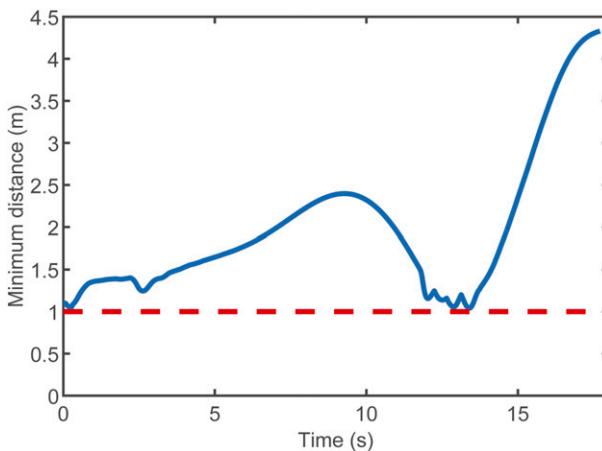


Figure 13. Minimum distance among drones. The blue curve is the minimum distance with respect to time. The red dotted line is the safety distance among drones.

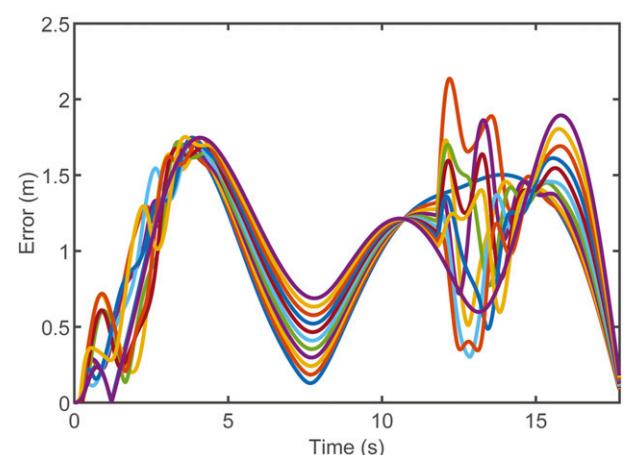


Figure 14. Comparison of actual and desired trajectories for the drone swarm.

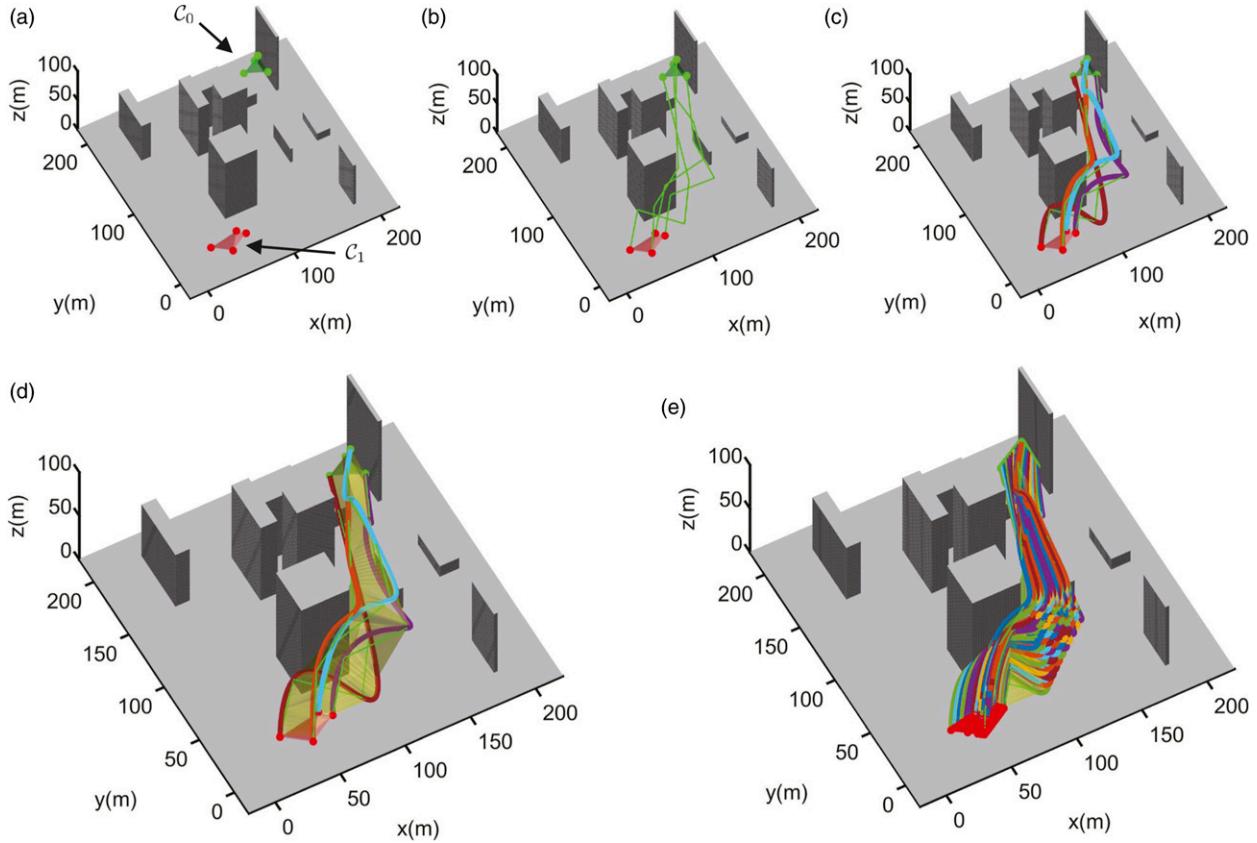


Figure 15. An optimal virtual tube planning process. (a) Constructing order pairs \mathcal{P} . The red points represent the start points, and the green points represent the goal points. The red trapezoidal pyramid denotes the start area, and the green trapezoidal pyramid denotes the goal area. (b) Finding paths. A path-finding algorithm is used to find paths, shown as green lines. (c) Optimizing trajectories. Optimal trajectories are generated by solving optimization problems. The colorful lines represent different trajectories for different pairs. (d) Constructing optimal virtual tube. An optimal virtual tube is generated based on the optimal trajectories, shown as a yellow area. (e) Generating infinite optimal paths. Other optimal trajectories, denoted by colorful lines, are automatically generated by interpolation.

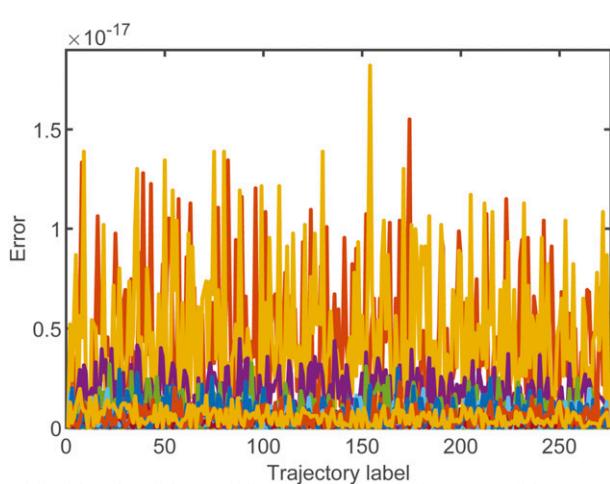


Figure 16. Errors between variable vectors of trajectories by interpolation and optimization problem are illustrated using colorful lines representing the error in each parameter of the variable vectors.

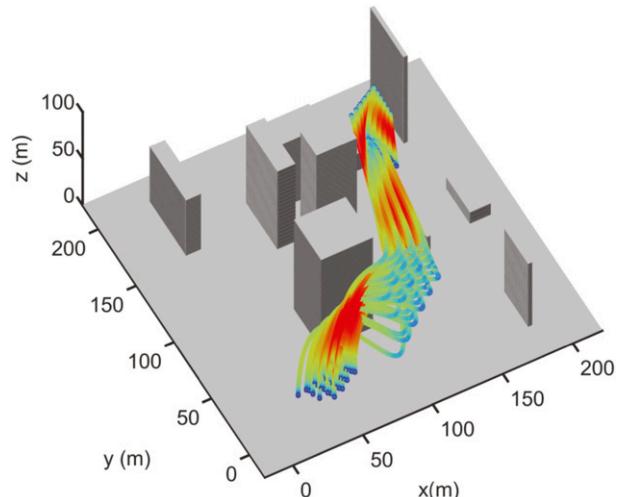


Figure 17. Simulation results of drone swarm following in 3-D. The colorful curves are different trajectories. The colors of trajectories correspond to varied speeds where the red represents the high speed and the blue represents the low speed.

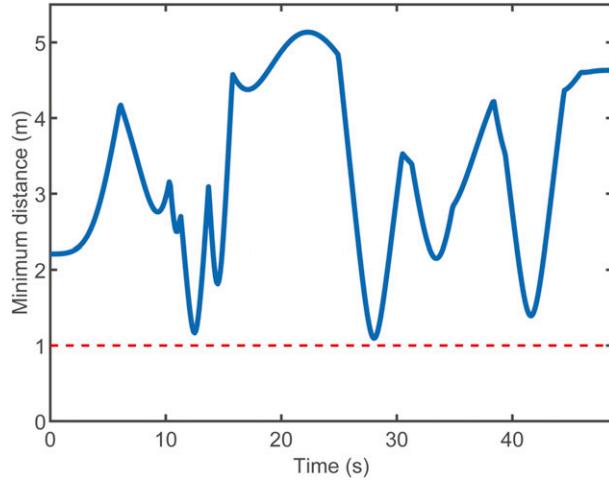


Figure 18. Minimum distance among drones in 3-D. The red dotted line is the 1m safety distance.

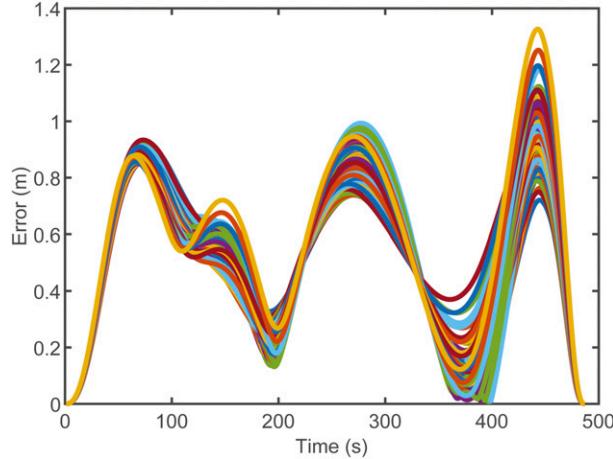


Figure 19. Error between desired and actual trajectory for drone swarm.

7.2.2. Complex scenarios. In the complex scenarios, as shown in Figure 20, the obstacles are randomly placed and the start area and goal area are designed as the tetrahedra. We test the navigation performance of the swarm in scenarios with 10, 20, 30, 40, and 50 random obstacles each. And the numbers of drones are set as the same.

Let the radius of drones be 0.5 m and the maximum speed of drones be 5 m/s. The results of the comparisons among the three methods are summarized in Table 1. Three typical scenarios with 10, 30, and 50 random obstacles are shown in Figure 21. (i) In an optimal virtual tube, each drone is assigned to an optimal trajectory considering the kinematics of the drone. Compared with other methods, the desired optimal trajectories make drone swarm fly more smoothly and fast as a group as shown in Figure 21(b). Constraints in the obstacle avoidance process are efficiently simplified as the result of replacing the normal constraints with tube boundary constraints. Moreover, there is no intersection between the optimal trajectories for all drones in the optimal virtual tube so that the

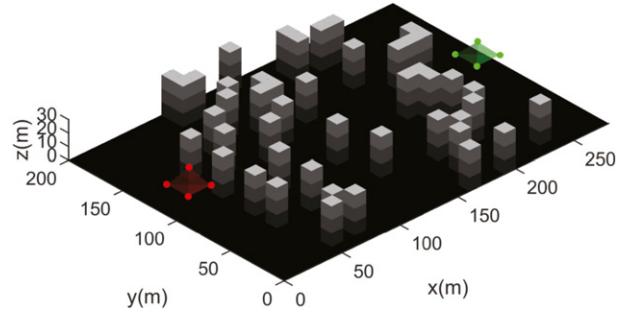


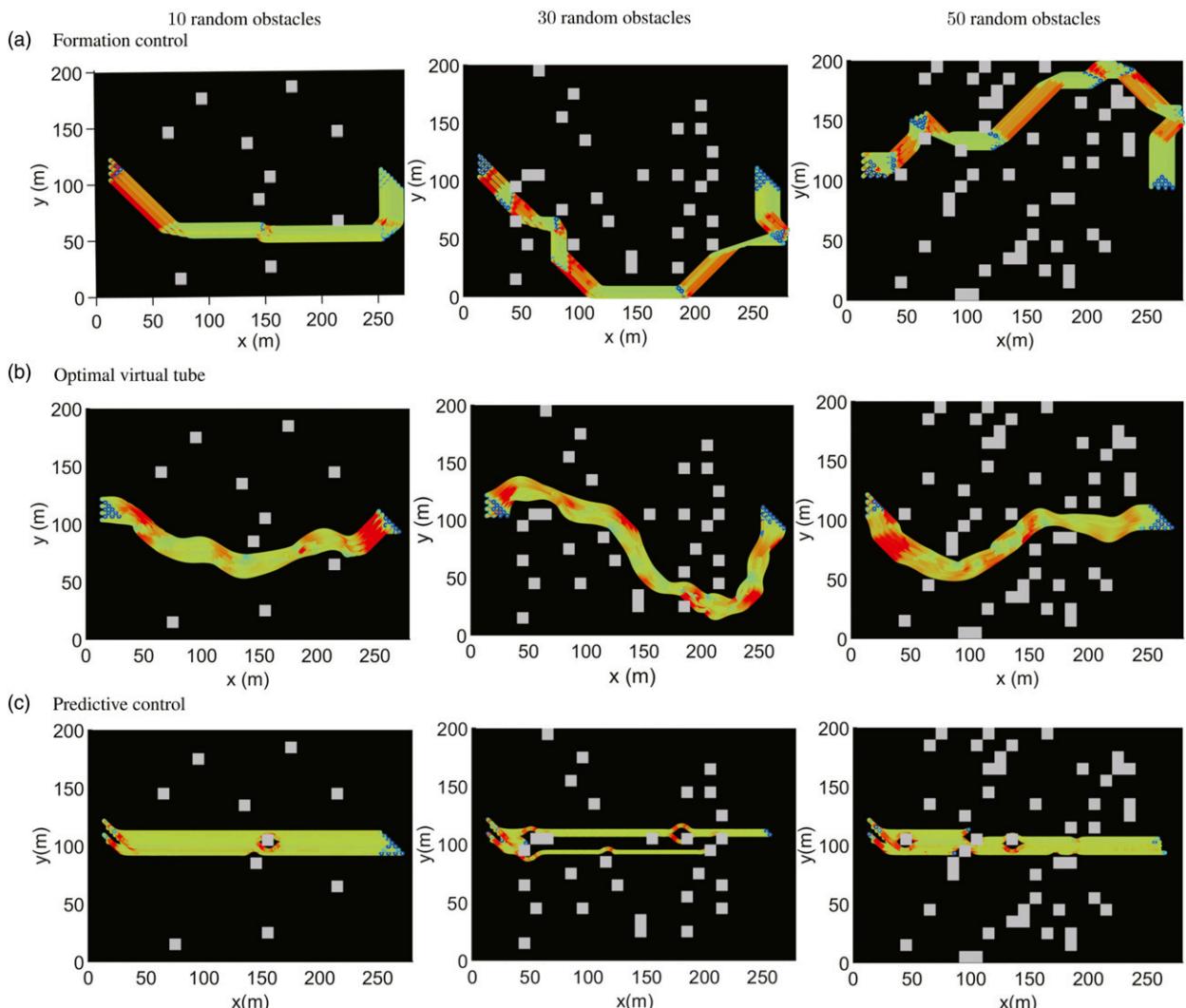
Figure 20. Complex scenario with random obstacles. The start area and goal area are denoted by the red and green tetrahedra.

collision avoidance among drones is reduced. Therefore, the optimal virtual tube planning method is superior to other methods in terms of average time and average speed in obstacle-dense environments. Meanwhile, the optimal virtual tube has the same arrival rate as the formation control due to containing goal path planning algorithms in both methods to avoid falling into a deadlock. Compared with the formation control, the optimal virtual tube achieves fast passing through obstacle-dense environments by relaxing hard formation constraints. The swarm will try to maintain the desired formation during the flight but will deform in a small gap between obstacles. (ii) As shown in Figure 21(a), the multi-robot formation control guarantees that all drones in the swarm arrive at the goal area while maintaining formation. The global path planner used in formation control finds several intersecting convex polytopes to remain the swarm in the free space so that all drones could reach the goal area, namely, 100% arrival rate can be achieved. However, the formation constraint also requires a minimum volume of the convex polytope while maintaining the formation, resulting in a longer planned path, compared with other methods. The average time, thus, is longer than that of our method. (iii) As for the predictive control of the aerial swarm, it has the minimum arrival time in obstacle-sparse environments. However, it should be noted that the average time is set to be infinity in obstacle-dense environments due to at least one drone does not reach the goal area in a certain time limit. Without the goal paths, the predictive control considers the goal point as the direction of migration that has a better performance in obstacles-sparse environments. However, in obstacles-dense environments especially with non-convex obstacles, the direction of obstacle avoidance may be opposite to the direction of migration so that the drone is easily blocked resulting in the lowest arrival rate, as shown in Figure 21(c). And the several non-regular obstacle constraints negatively affect the average speed.

7.2.3. Summaries. In this subsection, we have validated the superiority of our method in several scenarios with random obstacles by comparing with the state-of-art methods including multi-robot formation control and predictive control of aerial swarm. The metrics of our method including arrival rate, average time, and average speed are superior to other methods in obstacle-dense environments. The swarm could

Table 1. Performance Metrics for Different Methods in Complex Scenarios With a Varied Number of Obstacles.

The number of obstacles	Metrics	Methods		
		Formation control	Optimal virtual tube	Predictive control
10	Average time (s)	67.8	58.6	53.1
	Arrival rate (%)	100	100	100
	Average speed (m/s)	4.4556	4.6656	4.516
20	Average time (s)	67.5	59.3	51.5
	Arrival rate (%)	100	100	100
	Average speed (m/s)	4.5032	4.5992	4.5871
30	Average time (s)	84.6	76.0	∞
	Arrival rate (%)	100	100	35
	Average speed (m/s)	4.4792	4.6083	1.9498
40	Average time (s)	63.2	57.0	∞
	Arrival rate (%)	100	100	75
	Average speed (m/s)	4.5623	4.8729	3.4681
50	Average time (s)	83.3	65.3	∞
	Arrival rate (%)	100	100	75
	Average speed (m/s)	4.6201	4.8541	4.2050

**Figure 21.** Trajectories of swarm robotics for three methods in environments with different numbers of obstacles. The colors of trajectories correspond to varied speeds where the red represents the high speed and the blue represents the low speed.

pass through obstacle-dense environments as a group fast and smoothly.

7.3. Experiments

A $30\text{ m} \times 10\text{ m} \times 8\text{ m}$ space for real flight is depicted in Figure 22. Known obstacles are randomly positioned on the ground within this area, and a VICON motion capture system is utilized for drone localization. The drone depicted in Figure 23, which has a wheelbase of 350 mm, is equipped with a Jetson NX for onboard computation and utilizes the CUAV V5 nano Autopilot as its flight controller.

7.3.1. Task settings. The designated flight area is partitioned into three distinct regions, namely, the starting area, obstacle area, and goal area. The primary objective of this task for a swarm of four drones is to initiate their flight from the starting area, navigate through the obstacle area without any collision, and ultimately reach the goal area.

7.3.2. Optimal virtual tube planning. Firstly, the initial positions $\mathbf{q}_{0,k}$ of four drones in the start area must be specified, as depicted in Figure 24. It is observed that Drone 1, Drone 2, and Drone 3 are assigned to $\mathbf{q}_{0,1}$, $\mathbf{q}_{0,2}$, $\mathbf{q}_{0,3}$, respectively, whose convex hull is the terminal \mathcal{C}_0 in the start

area. And the position $\mathbf{q}_{0,4}$ of Drone 4 is the affine combination of $\{\mathbf{q}_{0,k}\}$ within the terminal \mathcal{C}_0 , which is expressed as

$$\mathbf{q}_{0,4} = \theta_{1,4}\mathbf{q}_{0,1} + \theta_{2,4}\mathbf{q}_{0,2} + \theta_{3,4}\mathbf{q}_{0,3}, \sum_{k=1}^3 \theta_{k,4} = 1. \quad (33)$$

Similarly, the goal positions $\mathbf{q}_{m,k}$ of drones are assigned. Specially, the goal position $\mathbf{q}_{m,4}$ is expressed as

$$\mathbf{q}_{m,4} = \theta_{1,4}\mathbf{q}_{m,1} + \theta_{2,4}\mathbf{q}_{m,2} + \theta_{3,4}\mathbf{q}_{m,3}, \sum_{k=1}^3 \theta_{k,4} = 1. \quad (34)$$

Thus, the order pairs $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})$ are obtained.

Subsequently, path-finding is used to generate paths for the three drones, as depicted in Figure 25(a). For each order pair $(\mathbf{q}_{0,k}, \mathbf{q}_{m,k})(k = 1, 2, 3)$, there are 8 intermediate configurations, with $m = 7$ denoting the final configuration. To parameterize the trajectories, normalized knots $\{t_i\}$ are automatically generated. Three optimization problems are then solved to obtain the optimal trajectories shown in Figure 25(b), which are formulated as follows:

$$\begin{aligned} \mathbf{h}^*((\mathbf{q}_{0,1}, \mathbf{q}_{7,1}), t) &= \mathbf{C}(t)\mathbf{x}_1, \\ \mathbf{h}^*((\mathbf{q}_{0,2}, \mathbf{q}_{7,2}), t) &= \mathbf{C}(t)\mathbf{x}_2, \\ \mathbf{h}^*((\mathbf{q}_{0,3}, \mathbf{q}_{7,3}), t) &= \mathbf{C}(t)\mathbf{x}_3. \end{aligned} \quad (35)$$

Hence, utilizing *Theorem 1*, the optimal virtual tube $(\mathcal{C}_0, \mathcal{C}_1, \mathbf{f}, \mathbf{h}^*)$ is constructed. Since the position $\mathbf{q}_{0,4}$ is located in the terminal \mathcal{C}_0 , the optimal trajectory $\mathbf{h}^*((\mathbf{q}_{0,4}, \mathbf{q}_{7,4}), t)$ for Drone 4 can be generated by



Figure 22. Space for real flight.



Figure 23. Drone used for real flight.

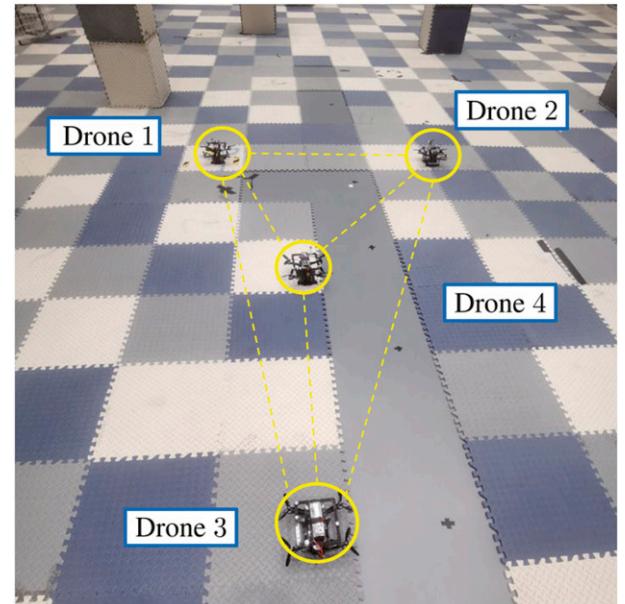


Figure 24. Drone positions at the start area.

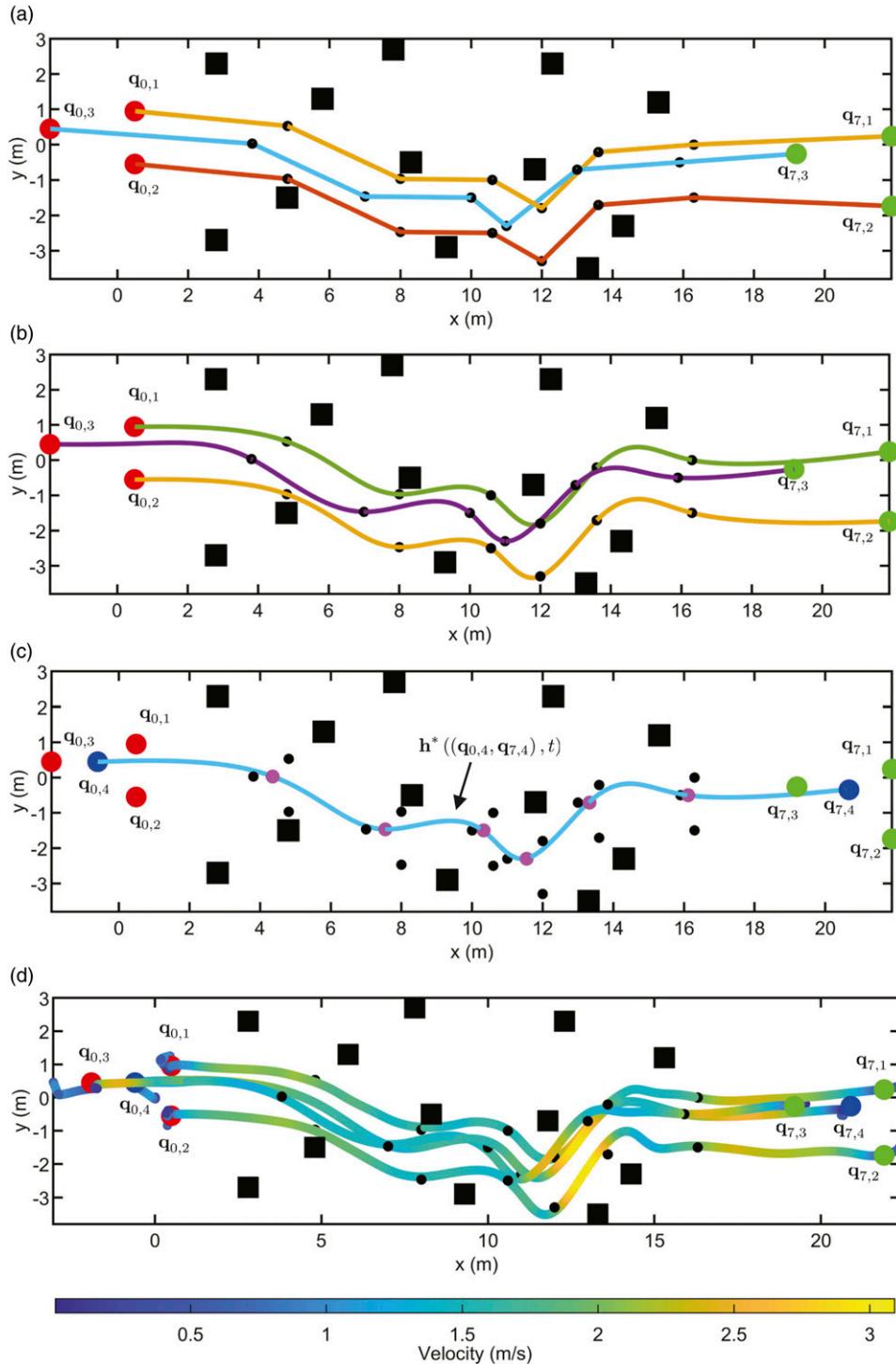


Figure 25. Process of optimal virtual tube planning and application. The squares are obstacles in reality. The red points and green points are start positions and goal positions, respectively. (a) The paths for three order pairs. (b) The optimal trajectories for three order pairs. (c) The optimal trajectory of the drone 4 is generated by interpolation. (d) The true trajectories of drone swarm.

$$\mathbf{h}^*((\mathbf{q}_{0,4}, \mathbf{q}_{7,4}), t) = \mathbf{C}(t) \sum_{k=1}^3 \theta_{k,4} \mathbf{x}_k, \quad (36)$$

as illustrated in Figure 25(c). Upon obtaining the four trajectories for the drones, each drone can track its own trajectory while avoiding the others, as depicted in Figure 25(d). As

depicted in Figure 26, the 4th optimal trajectory of Drone 4 is generated by the convex combination of three optimal trajectories. Subsequently, the drone swarm successfully tracks trajectories to pass through the environment. The velocity of randomly selected Drone 2 with respect to time is shown in Figure 27, which indicates that the drone swarm can fast and smoothly pass through the optimal virtual tube.

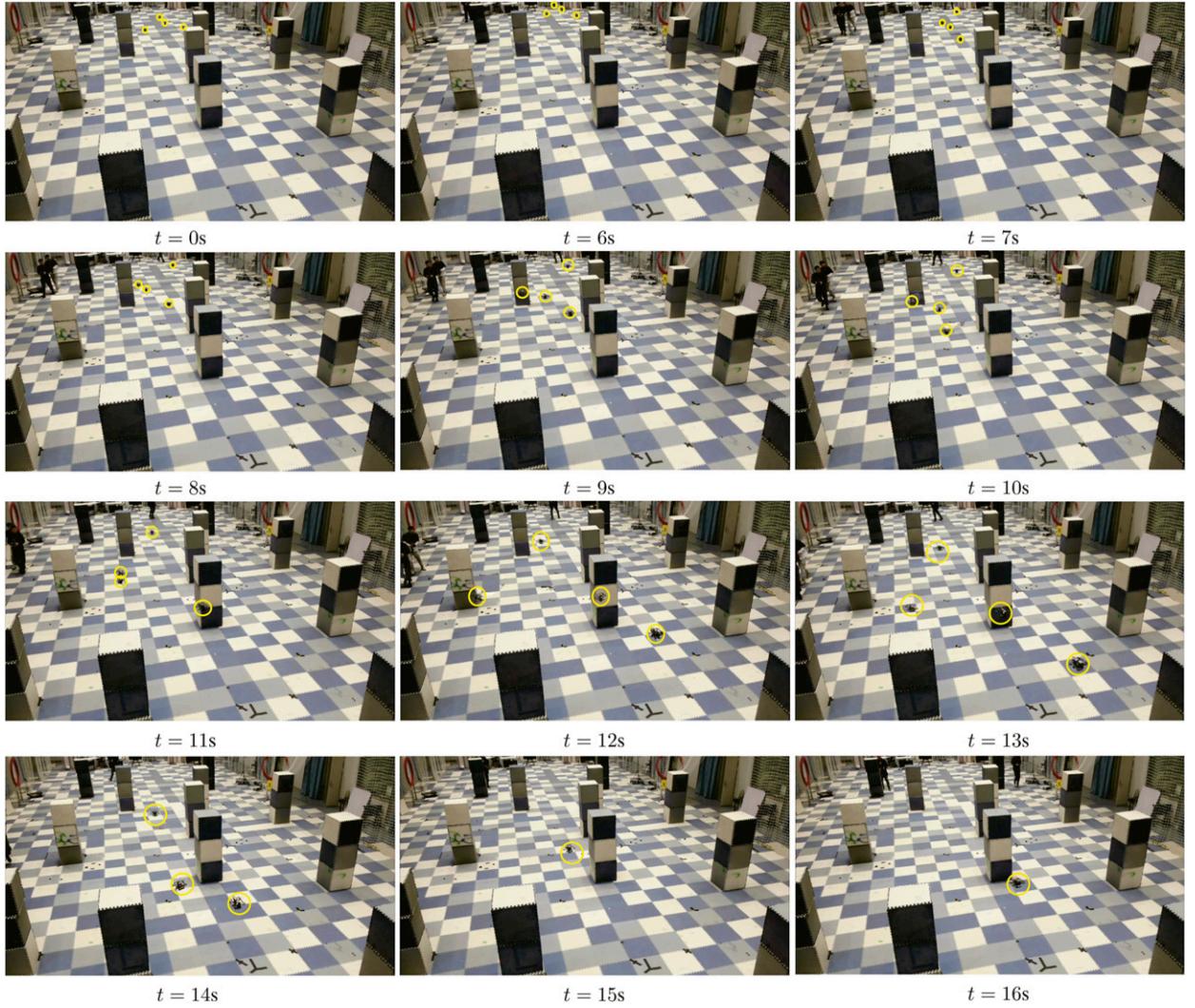


Figure 26. Real flight of drone swarm.

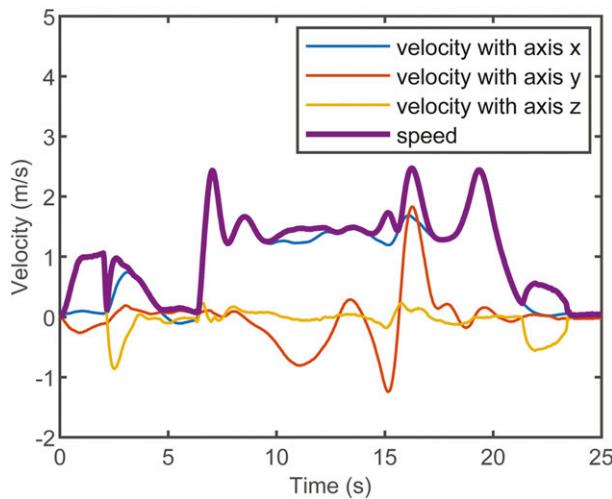


Figure 27. Velocity of drone 2.

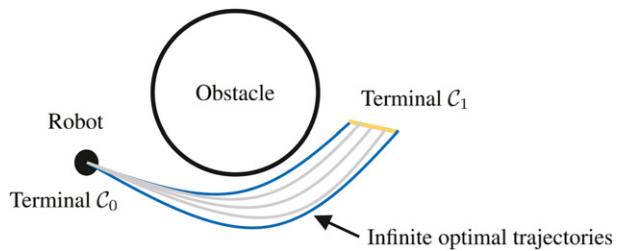


Figure 28. An example for a single robot to generate infinite optimal trajectories. The terminal \mathcal{C}_0 is shrunk to be a point where the robot is, and the terminal \mathcal{C}_1 is denoted by the yellow line.

7.4. Summary

The hierarchical approach including the optimal virtual tube planning method and model predictive control is validated in simulations and experiments. First, the optimal virtual tube planning method in 2-D and 3-D space is simulated to

verify the correctness of *Theorem 1* by comparing it with the traditional methods. And the MPC controller is implemented in the swarm to achieve inter-drone collision avoidance while the trajectory tracking is achieved. Simulation results demonstrate that the proposed method of optimal virtual tube planning is applicable to large-scale swarm movements. The number of optimization problems is independent of the number of drones when the start positions of drones are within the terminal \mathcal{C}_0 . Then, comparisons with other methods show that our approach implements the fast and safe movement of the swarm as a group in obstacle-dense environments. In real flight, the approach is validated by the swarm with 4 drones passing through an obstacle-dense environment fast and smoothly.

8. Conclusion and future work

This paper extends the definition of the virtual tube, proposes optimal virtual tubes, and then analyzes their properties. One type of virtual tubes, namely, the linear virtual tube with convex hull terminals, is used for constructing an optimal virtual tube. Then, a planning method for this linear virtual tube with convex hull terminals is proposed and its effectiveness is demonstrated in simulations and experiments. The results in 2-D and 3-D show that the error between results obtained through our planning method and optimization is negligible. Therefore, our method can efficiently reduce the computation burden and generate optimal trajectories by convex combinations. An application of the optimal virtual tube, which uses an MPC controller for a drone swarm to track the trajectories in simulations and experiments, shows that the swarm could pass through obstacle-dense environments fast and safely as a group. In addition, as shown in Figure 28, when the start terminal of the virtual tube is shrunk to a point, it can also be applied to single-robot trajectory planning, that is, to plan infinite optimal trajectories for a single robot to reach the target area.

In summary, there are several advantages of our method. (i) The computational burden involved in trajectory planning for large swarm robotics is significantly reduced by our approach, making it practical for real-world applications. (ii) Swarm robotics is enabled by our method to navigate through obstacle-dense environments fast and safely as a cohesive group with desired formations. (iii) Enhanced scalability for swarm robotics is provided by our method, enabling the management of larger swarms while maintaining efficient navigation. On the other hand, there are also some limitations of our method. (i) The large swarm robotics may be blocked in extremely cluttered environments.⁵ Due to the property of simply connected cross-sections, all trajectories in the virtual tube are homotopic, preventing the swarm from splitting apart to move around obstacles. This may cause difficulties in extremely cluttered environments. When the space of the gap that the virtual tube passes through is smaller than the physical volume of the swarm, collision avoidance procedure is necessary for the swarm to pass through this gap. However, this may block

the swarm. (ii) The swarm is not allowed dividing into subgroups for a single virtual tube. Multiple virtual tubes become necessary when the swarm is partitioned into distinct subgroups for various goal areas or within extremely cluttered environments. Nonetheless, it is worth noting that managing multiple virtual tubes may entail greater complexity and potentially higher costs compared to individual planning for each robot in extremely cluttered environments.

To overcome these limitations and make the swarm autonomous, dependable, and affordable (ADA) (Cai et al. (2021)), further studies are included in the following. (i) Path planning for the virtual tube will be designed to reduce the computation time and limit the minimum volume of the cross-section according to the physical volume of the swarm. (ii) The design of the map f for ordered pairs needs to be studied further, encompassing not only the convex hull terminals but also the general terminals. This map f will aid in the prevention of path crossings and the efficient conservation of energy, among other potential objectives. In cases where paths for different order pairs intersect, it can be difficult to distinguish between the boundary and interior of the virtual tube, where the boundary is used to constrain the swarm within the virtual tube. (iii) Furthermore, a method of constructing the virtual tube network or multiple virtual tubes will be explored to allow the swarm to split apart to move around obstacles.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China under Grant 61973015.

ORCID iDs

Pengda Mao  <https://orcid.org/0009-0003-3442-0492>

Rao Fu  <https://orcid.org/0000-0002-6000-4906>

Quan Quan  <https://orcid.org/0000-0001-8216-8998>

Notes

1. The scale of a swarm means the number of robots in the swarm.
2. A real-valued function is said to be smooth if its derivatives of all orders exist and are continuous.
3. There is another popular approach to analyze the complexity of optimization. The black-box model is used for the oracle complexity, which regards queries to the object function as oracles and considers the number of queries (Bubeck (2015)). However, it is not suitable for comparison with the convex combination operation.
4. In practice, the normalized knots $\{t_i\}$ are not suitable for a robot to track trajectory. To address this issue, the $t(s) = v_R/u_ms$ is

- used in the trajectory $\mathbf{h}^*((\mathbf{q}_0, \mathbf{q}_m), t(s))$, where v_R is a coefficient related with robots, $u_m \in \{u_i\}$.
5. The extremely cluttered environment signifies that the gaps or spaces between obstacles are considerably narrower than the collective volume occupied by the swarm, compared with the obstacle-dense environment. For example, the environment where gaps between obstacles can only hold a single robot to pass through is the extremely cluttered environment.

References

- Adasch N, Ernst B and Keim D (2006) *Topological Vector Spaces: The Theory without Convexity Conditions*. Berlin, Germany: Springer
- Airbus (2019) *Airbus Skyways: The Future of the Parcel Delivery in Smart Cities*. Blagnac, France: Airbus. <https://www.embention.com/projects/airbus-parcel-delivery/>
- Alonso-Mora J, Baker S and Rus D (2017) Multi-robot formation control and object transport in dynamic environments via constrained optimization. *The International Journal of Robotics Research* 36(9): 1000–1021.
- Boyd SP and Vandenberghe L (2004) *Convex Optimization*. Cambridge, MA: Cambridge University Press.
- Bubeck S (2015) Convex optimization: algorithms and complexity. *Foundations and Trends® in Machine Learning* 8(3–4): 231–357.
- Cai KY, Trivedi KS and Yin B (2021) S-Ada: software as an autonomous, dependable and affordable system. 2021 51st Annual IEEE/IFIP international conference on dependable systems and networks-supplemental volume (DSN-S), Taipei, Taiwan, 21–24 June 2021.
- Deits R and Tedrake R (2015) Efficient mixed-integer planning for UAVs in cluttered environments. 2015 ieee international conference on robotics and automation (ICRA), Washington, DC, USA, 26–30 May 2015.
- Ding W, Gao W, Wang K, et al. (2019) An efficient B-Spline-Based kinodynamic replanning framework for quadrotors. *IEEE Transactions on Robotics* 35: 1287–1306.
- Gao F, Wang L, Zhou B, et al. (2020) Teach-repeat-replan: a complete and robust system for aggressive flight in complex environments. *IEEE Transactions on Robotics* 36(5): 1526–1545.
- Gao Y, Bai C, Zhang L, et al. (2022) Multi-UAV cooperative target encirclement within an annular virtual tube. *Aerospace Science and Technology* 128: 107800.
- Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.
- Liu S, Watterson M, Mohta K, et al. (2017) Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. *IEEE Robotics and Automation Letters* 2: 1688–1695.
- Luo Y, Cai P, Bera A, et al. (2018) Porca: modeling and planning for autonomous driving among many pedestrians. *IEEE Robotics and Automation Letters* 3: 3418–3425.
- Mao P and Quan Q (2022) Making robotics swarm flow more smoothly: a regular virtual tube. IEEE/RSJ international conference on intelligent robots and systems (IROS), Detroit, MI, USA, 23–27 October 2022.
- Mellinger D and Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. 2011 IEEE international conference on robotics and automation, Shanghai, China, 9–13 May 2011.
- Nagrare SR, Tony LA, Ratnoo A, et al. (2022) Multi-lane UAV traffic management with path and intersection planning. AIAA Scitech 2022 Forum, San Diego, CA, USA, 3–7 January 2022.
- Orthey A, Frész B and Toussaint M (2019) Motion planning explorer: visualizing local minima using A local-minima tree. *IEEE Robotics and Automation Letters* 5(2): 346–353.
- Osa T (2022) Motion planning by learning the solution manifold in trajectory optimization. *The International Journal of Robotics Research* 41(3): 281–311.
- Park J, Lee SM, Shim SW, et al. (2021) The influence of restrictive pulmonary dysfunction on osteoporotic thoracic vertebral fractures. *Asian spine journal* 15: 659–663.
- Pentico DW (2007) Assignment problems: a golden anniversary survey. *European Journal of Operational Research* 176(2): 774–793.
- Piegł LA and Tiller W (2000) Surface approximation to scanned data. *The Visual Computer* 16(7): 386–395.
- Quan Q, Fu R, Li M, et al. (2022) Practical distributed control for VTOL UAVs to pass a virtual tube. *IEEE Transactions on Intelligent Vehicles* 7(2): 342–353.
- Quan Q, Fu R and Cai KY (2023a) How far two UAVs should be subject to communication uncertainties. *IEEE Transactions on Intelligent Transportation Systems* 24(1): 429–445.
- Quan Q, Gao Y and Bai C (2023b) Distributed control for a robotic swarm to pass through a curve virtual tube. *Robotics and Autonomous Systems* 162: 104368.
- Rasekhipour Y, Khajepour A, Chen SK, et al. (2016) A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems* 18: 1255–1267.
- Richter C, Bry A and Roy N (2016) Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. *Robotics Research: The 16th International Symposium ISRR*. Berlin, Germany: Springer.
- Shick PL (2007) *Topology: Point-set and Geometric*. London, UK: Wiley-Interscience.
- Soria E, Schiano F and Floreano D (2021) Predictive control of aerial swarms in cluttered environments. *Nature Machine Intelligence* 3(6): 545–554.
- Soria E, Schiano F and Floreano D (2022) Distributed predictive drone swarms in cluttered environments. *IEEE Robotics and Automation Letters* 7(1): 73–80.
- Strothotte T and Schlechtweg S (2002) 7 - geometric models and their exploitation in NPR. *Non-Photorealistic Computer Graphics, the Morgan Kaufmann Series in Computer Graphics*. Burlington, MA: Morgan Kaufmann, pp. 203–245.
- Tang S, Thomas J and Kumar V (2018) Hold or take optimal plan (hoop): a quadratic programming approach to multi-robot trajectory generation. *The International Journal of Robotics Research* 37(9): 1062–1084.

- Tu LW (2011) *An Introduction to Manifolds*. Berlin, Germany: Springer.
- Usenko V, von Stumberg L, Pangercic A, et al. (2017) Real-time trajectory replanning for MAVs using uniform B-splines and A 3D circular buffer 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 215–222.
- Wright SJ (1997) *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics.
- Zhou B, Gao F, Wang L, et al. (2019) Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters* 4(4): 3529–3536.
- Zhou B, Gao F, Pan J, et al. (2020) Robust real-time UAV re-planning using guided gradient-based optimization and topological paths. 2020 IEEE international conference on robotics and automation (ICRA), Paris, France, 31 May 2020.
- Zhou X, Zhu J, Zhou H, et al. (2021) EGO-swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments. 2021 IEEE international conference on robotics and automation (ICRA), Xi'an, China, 5 June 2021.

Appendix

A Proofs of Lemma 1 and Lemma 2

A.1 Proof of Lemma 1.

Proof. In Step 1, it is proved that $\mathbf{x}(\boldsymbol{\theta})$ is feasible. Then the optimality of $\mathbf{x}(\boldsymbol{\theta})$ is shown in Step 2.

Step 1. According to an optimality criterion (Boyd and Vandenberghe (2004)) for differentiable f_0 . Let \mathcal{X}_θ , \mathcal{X}_k denote feasible sets:

$$\mathcal{X}_\theta = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}(\boldsymbol{\theta})\}, \mathcal{X}_k = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}_k\}.$$

Then for all $\mathbf{x} \in \mathcal{X}_k$,

$$\nabla f_0(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) \geq 0, k = 1, 2, \dots, q. \quad (37)$$

Since \mathbf{x}_k is feasible, each solution \mathbf{x} in equation (37) can be expressed as the combination of a special solution and a solution in zero space, such as $\mathbf{x} = \mathbf{x}_k + \mathbf{v}, \mathbf{v} \in N(\mathbf{A})$. Therefore, this optimal condition (37) is transformed into

$$\nabla f_0(\mathbf{x}_k)^T \mathbf{v} \geq 0, \text{ for all } \mathbf{v} \in N(\mathbf{A}). \quad (38)$$

When $\mathbf{b} = \mathbf{b}(\boldsymbol{\theta})$, we substitute $\mathbf{x}(\boldsymbol{\theta})$ into (38) to obtain

$$\mathbf{A}\mathbf{x}(\boldsymbol{\theta}) = \sum_{k=1}^q \theta_k \mathbf{A}\mathbf{x}_k = \sum_{k=1}^q \theta_k \mathbf{b}_k = \mathbf{b}(\boldsymbol{\theta}).$$

Therefore, $\mathbf{x}(\boldsymbol{\theta})$ is feasible.

Step 2. For all $\mathbf{x} \in \mathcal{X}_\theta$, it has

$$\nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T(\mathbf{x} - \mathbf{x}(\boldsymbol{\theta})) = \sum_{k=1}^q \theta_k \nabla f_0(\mathbf{x}_k)^T \mathbf{v}. \quad (39)$$

Combining equation (39) with (38) derives

$$\nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T(\mathbf{x} - \mathbf{x}(\boldsymbol{\theta})) \geq 0, \text{ for all } \mathbf{x} \in \mathcal{X}_\theta. \quad (40)$$

Therefore, $\mathbf{x}(\boldsymbol{\theta})$ is optimal.

A.2 Proof of Lemma 2.

Proof. In Step 1, it is shown that $\mathbf{x}(\boldsymbol{\theta})$ is feasible. Then the optimality of $\mathbf{x}(\boldsymbol{\theta})$ is shown in Step 2.

Step 1. According to an optimality criterion (Boyd and Vandenberghe (2004)) for differentiable f_0 . Let \mathcal{X}_θ , \mathcal{X}_k denote feasible sets,

$$\mathcal{X}_\theta = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}(\boldsymbol{\theta}), f_i(\mathbf{x}) \leq 0, i = 1, \dots, n_c\},$$

$$\mathcal{X}_k = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}_k, f_i(\mathbf{x}) \leq 0, i = 1, \dots, n_c\}.$$

Let \mathcal{V}_k denote the equivalent feasible set,

$$\mathcal{V}_k = \{\mathbf{v} | \mathbf{A}(\mathbf{x}_k + \mathbf{v}) = \mathbf{b}_k, f_i(\mathbf{x}_k + \mathbf{v}) \leq 0, i = 1, \dots, n_c\}.$$

And for all $\mathbf{x} \in \mathcal{X}_k$,

$$\nabla f_0(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) \geq 0. \quad (41)$$

Since \mathbf{x}_k is feasible, every \mathbf{x} has $\mathbf{x} = \mathbf{x}_k + \mathbf{v}, \mathbf{v} \in \mathcal{V}_k$. Therefore, the optimal condition (41) can be expressed as

$$\nabla f_0(\mathbf{x}_k)^T \mathbf{v} \geq 0, \text{ for all } \mathbf{v} \in \mathcal{V}_k. \quad (42)$$

Let \mathbf{b} be $\mathbf{b}(\boldsymbol{\theta})$. Substituting $\mathbf{x}(\boldsymbol{\theta})$ into constraints yields

$$\mathbf{A}\mathbf{x}(\boldsymbol{\theta}) = \sum_{k=1}^q \theta_k \mathbf{A}\mathbf{x}_k = \sum_{k=1}^q \theta_k \mathbf{b}_k = \mathbf{b}(\boldsymbol{\theta}),$$

$$f_j(\mathbf{x}(\boldsymbol{\theta})) \leq \sum_{k=1}^q \theta_k f_j(\mathbf{x}_k) \leq 0, j = 1, \dots, n_c.$$

Therefore, $\mathbf{x}(\boldsymbol{\theta})$ is feasible.

Step 2. For all $\mathbf{x} \in \mathcal{X}_\theta$, it has equivalent constraint $\mathcal{V}_\theta = \sum_{k=1}^q \theta_k \mathcal{V}_k$. Therefore, for all $\mathbf{v}_\theta \in \mathcal{V}_\theta$, there exist specific $\mathbf{v}_k \in \mathcal{V}_k$ so that

$$\mathbf{v}_\theta = \sum_{k=1}^q \theta_k \mathbf{v}_k.$$

In the following, we will show $\nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T \mathbf{v}_\theta \geq 0$.

When $q = 2$,

$$\begin{aligned} & \nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T \mathbf{v}_\theta \\ &= (\theta_1^2 + \theta_1 \theta_2) \nabla f_0(\mathbf{x}_1)^T \mathbf{v}_1 + (\theta_2^2 + \theta_1 \theta_2) \nabla f_0(\mathbf{x}_2)^T \mathbf{v}_2 \quad (43) \\ &+ \theta_1 \theta_2 (\nabla f_0(\mathbf{x}_1) - \nabla f_0(\mathbf{x}_2))^T (\mathbf{x}_1 - \mathbf{x}_2). \end{aligned}$$

Since f_0 is a convex function, there have

$$f_0(\mathbf{x}_1) \geq f_0(\mathbf{x}_2) + \nabla f_0(\mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2), \quad (44a)$$

$$f_0(\mathbf{x}_2) \geq f_0(\mathbf{x}_1) + \nabla f_0(\mathbf{x}_1)^T (\mathbf{x}_2 - \mathbf{x}_1). \quad (44b)$$

Combining (44a) and (44b), we obtain

$$(\nabla f_0(\mathbf{x}_1) - \nabla f_0(\mathbf{x}_2))^T (\mathbf{x}_1 - \mathbf{x}_2) \geq 0. \quad (45)$$

Combining equation (43) with (45) results in

$$\nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T \mathbf{v}_{\boldsymbol{\theta}} \geq 0, \text{ for all } \mathbf{v}_{\boldsymbol{\theta}} \in \mathcal{V}_{\boldsymbol{\theta}}. \quad (46)$$

Suppose that Equation (46) holds when $q = n$. Then, when $q = n + 1$, we have

$$\begin{aligned} \mathbf{v}_{\boldsymbol{\theta}} &= \theta_{n+1} \mathbf{v}_{n+1} + \sum_{k=1}^n \theta_k \mathbf{v}_k \\ &= \theta_{n+1} \mathbf{v}_{n+1} + \sum_{k=1}^n \theta_k \sum_{k=1}^n \frac{\theta_k}{\sum_{k=1}^n \theta_k} \mathbf{v}_k \\ &= \theta_{n+1} \mathbf{v}_{n+1} + (1 - \theta_{n+1}) \sum_{k=1}^n \tilde{\theta}_k \mathbf{v}_k, \\ \mathbf{x}_{\boldsymbol{\theta}} &= \theta_{n+1} \mathbf{x}_{n+1} + (1 - \theta_{n+1}) \sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k. \end{aligned} \quad (47)$$

We substitute (47) into (46) to obtain that

$$\begin{aligned} \nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T \mathbf{v}_{\boldsymbol{\theta}} &= \theta_{n+1}^2 \nabla f_0(\mathbf{x}_{n+1})^T \mathbf{v}_{n+1} \\ &\quad + \theta_{n+1}(1 - \theta_{n+1}) \nabla f_0(\mathbf{x}_{n+1})^T \mathbf{v}_{n+1} \\ &\quad + (1 - \theta_{n+1})^2 \nabla f_0 \left(\sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right)^T \sum_{k=1}^n \tilde{\theta}_k \mathbf{v}_k \\ &\quad + \theta_{n+1}(1 - \theta_{n+1}) \nabla f_0 \left(\sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right)^T \sum_{k=1}^n \tilde{\theta}_k \mathbf{v}_k \\ &\quad + \theta_{n+1}(1 - \theta_{n+1}) \left(\nabla f_0(\mathbf{x}_{n+1}) - \nabla f_0 \left(\sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right) \right)^T \\ &\quad \cdot \left(\mathbf{x}_{n+1} - \sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right). \end{aligned}$$

In the same way, we could obtain

$$\left(\nabla f_0(\mathbf{x}_{n+1}) - \nabla f_0 \left(\sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right) \right)^T \left(\mathbf{x}_{n+1} - \sum_{k=1}^n \tilde{\theta}_k \mathbf{x}_k \right) \geq 0.$$

It is derived that

$$\nabla f_0(\mathbf{x}(\boldsymbol{\theta}))^T \mathbf{v}_{\boldsymbol{\theta}} \geq 0, \text{ for all } \mathbf{v}_{\boldsymbol{\theta}} \in \mathcal{V}_{\boldsymbol{\theta}}. \quad (48)$$

Therefore, $\mathbf{x}(\boldsymbol{\theta})$ is optimal.