International Journal of Aeronautical and Space Sciences
https://doi.org/10.1007/s42405-023-00629-w

IJASS International Journal of Aeronautical and Space Sciences

ORIGINAL PAPER

Trajectory Planning with Collision Avoidance for Multiple Quadrotor UAVs Using DMPC

Yuhang Jiang[1] · Shiqiang Hu[1] · Christopher Damaren[2] · Lingkun Luo[1] · Bing Liu[3]

Received: 25 April 2022 / Revised: 19 January 2023 / Accepted: 21 May 2023

Abstract
Trajectory planning with collision avoidance plays an important role for the safe application of multi-UAV systems in low altitude airspace. Although the synchronous DMPC algorithm had been widely applied in multi-agent systems due to its lower communication and computing cost, it generally suffers from the strict requirements. For example, the additional terminal conditions significantly reduce the maneuverability of the UAV in the fleet, whereas which ensure the stability of the algorithm and the feasibility of recursion. To remedy the raised issues, in this paper, a novel set of terminal conditions is proposed, which effectively reduces the conservativeness of the collision-free trajectories and satisfies the requirements of collision avoidance algorithms simultaneously. In this research, we also leverage the theoretical qualitative analysis, and thus providing initial guesses and related parameter settings for the algorithm. Finally, sufficient numerical simulations are proposed to verify the effectiveness and superiority of the proposed algorithm.

Keywords Collision avoidance · Trajectory planning · Unmanned aerial vehicle · DMPC

1 Introduction

Unmanned aerial vehicles (UAVs) have been widely applied in both military operations and civilian domains, e.g., border surveillance, law enforcement, fire fighting, precision agricultural, etc. [1]. By inheriting the merits from the multi-agent system techniques, their cooperative behavior and the ability to solve complex problems lead multi-UAV systems receive more research attention than single UAV systems in recent years [2].

With the continuous development of sensing technology, the flight information can be collected efficiently from spatially distributed UAVs through advanced sensors in a multi-agent UAV system. However, the coordination of UAV fleet is generally the core technology of a series of multi-agent UAV system applications [3, 4]. Therefore, the autonomous coordination capabilities of these systems receive significant attentions, and trajectory planning as well as the guidance for multiple UAVs is one of the crucial techniques in real application [5]. Specifically, generating the collision-free trajectories connecting the initial and final positions for every UAV in the fleet is a safety-critical task [6, 7].

In the research of trajectory planning with collision avoidance, the system hybridizes both physical constraints (e.g. the saturation of inputs) and task constraints (e.g. collision avoidance). Consequently, it is a common approach to formulate such task as an optimization problem due to its capability of handling constraints and ensuring required control performance. The optimization approach has been widely applied in solving the collision avoidance problem involving two UAVs [8, 9], and it has shown the apparent superiority compared with other algorithms such as sense-and-avoid approaches and potential field method.

✉ Shiqiang Hu
sqhu@sjtu.edu.cn

Yuhang Jiang
zjzl2001@sjtu.edu.cn

Christopher Damaren
damaren@utias.utoronto.ca

Lingkun Luo
lolinkun1988@sjtu.edu.cn

Bing Liu
liubingerbian@163.com

1 School of Aeronautics and Astronautics, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China

2 Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, ON M3H 5T6, Canada

3 China National Aeronautical Radio Electronics Research Institute, 432 Ziyue Road, Shanghai 200241, China

Published online: 11 July 2023

Through deep analyzing the different optimization algorithms, the Mixed Integer Linear Programming (MILP) [10, 11] and Mixed Integer Quadratic Programming (MIQP) [12] were first proposed to deal with collision avoidance problems between multiple UAVs. However, these methods require strict optimality guarantee, and thus unable to be well suited for large UAV swarm due to their computational complexity. Lately, the combinations of optimization and traditional gird-based planning methods, such as Multi-Agent Rapidly-exploring Random Tree (DMA-RRT), have also been investigated by many researchers [13]. However, these methods suffered due to the unpredictable calculation time and no guarantee of convergence within a finite time interval. Alternatively, the method based on Sequential Convex Programming (SCP) can achieve faster computation, while the simultaneous coordinated planning remains a challenge for large scales [14]. It is worth noting that, the decoupled incremental SCP (iSCP) method improves the scalability by using sequential prioritized planning to decouple inter-UAV constraints and shows improved scalability, while the success rate is reduced due to the increment of the scenario congestion [15].

Most of the algorithms mentioned above are mainly focused on solving the collision avoidance problem in a centralized manner [16]. However, considering the pairwise distance constraints of collision avoidance, distributed control strategies with lower communication and calculation cost have attracted great attention in recent years [17–20]. As a result, we embrace Distributed Model Predictive Control (DMPC) techniques to well address the required tasks in this paper.

1.1 Related Works

1.1.1 Distributed Model Predictive Control (DMPC)

So far as we know, a series of popular research has been conducted on the DMPC algorithm for trajectory planning of multi-UAV systems. In this scheme, each UAV in the fleet achieves its collision-free trajectory by solving an individual optimization problem. The inter-UAV collision avoidance constraints are satisfied by obtaining the assumed trajectory information of other UAVs. Three different DMPC algorithms have been derived from the way that the assumed trajectories are processed. In the sequential approach, the posterior UAVs can make use of the actual information of the anterior ones. In the iterative approach, each UAV solves its optimization problem and communicates with its neighbours iteratively in each sampling period. Both of them involve more time than a synchronous approach, in which the UAVs simultaneously update their predictions in each sampling period [21].

However, there is an uncertainty deviation between the actual trajectory and the planned trajectory. As a consequence, it is important to first ensure the feasibility of recursion and the stability of the algorithm before applying the DMPC algorithm. Li Dai et al. imposed compatibility constraint and terminal ingredients to provide a comprehensive proof of recursive feasibility of the optimization problem and closed-loop stability of the whole system [22, 23].

The additional compatibility constraint limits the distance between each UAV's real operation and its assumed trajectory to establish the agreement among UAVs. However, it also reduces the mobility of UAVs drastically, especially in cluttered area. Some recent studies have proposed many novel approaches on this issue. For example, R. Van Parys and G. Pipeleers use separating hyper-plane constraints between the UAVs as an alternative to perform the synchronous DMPC [19]. An on-demand collision avoidance strategy enhanced by the use of soft collision constraints is investigated to reduce the computational time of the task [20].

Similarly, although the use of terminal conditions can ensure that the UAV swarm satisfies the collision avoidance constraints, the trajectories obtained will be very conservative when the final distance of any two UAVs is closed to the minimum safety distance. To solve this issue, our research aims to propose a novel set of terminal conditions to relieve the conservativeness of the problem.

1.1.2 The Rolling-Horizon Implementation

The DMPC algorithm is generally implemented in a rolling-horizon manner [24, 25]. As an example, the rolling horizon strategy and Legendre pseudo-spectral collocation were proposed to iteratively re-plan the collision-free trajectories of multiple UAVs in a recent study [26].

It is worth noting that the choice of initial guess is particularly important in the implementation of the algorithm. A well-chosen initial feasible solution that satisfies all the constraints is still of great importance to speed up the calculation and ensure the recursive feasibility of the algorithm. Besides, the setting of the parameters such as the look-ahead time and the calculation frequency also plays the vital role in ensuring the successful implementation of the rolling-horizon algorithm.

However, the existing references rarely discuss the initial value selections and parameter settings in the implementation of the similar collision avoidance algorithm, which makes the search for the best candidate parameter settings significantly lack the theoretical guidance.

On the basis of the DMPC algorithm analysis, some useful theoretical qualitative analysis are discussed in this research, which greatly improves the previous research by enabling the instructions on the initial guess of the algorithm and the set of the relevant parameters.

1.2 Main Work and Contributions

Motivated by the above discussions, the research aims to develop an improved synchronous DMPC algorithm for trajectory planning with collision avoidance for multiple UAVs.

In this paper, quadrotor UAVs, which can be formulated by the linear, time-invariant and homogeneous dynamics are chosen so that the corresponding results can be shown clearly and the comparison with the existing results can be explicitly verified. For simplicity, the movement of the UAV are restricted to a 2D-plane in this research, and the collision avoidance strategy of UAVs in 3D-space can be obtained by the similar way.

The main work and contributions of the work can be summarized as follows:

1. By improving the design of terminal ingredients (especially the design of the terminal set), the feasible range of optimization is expanded, and the trajectory maneuverability of UAVs under dense target positions is improved.
2. Combined with the analysis of the algorithm stability and recursive feasibility, some beneficial guidelines of the initial guess selection and parameter settings in the DMPC algorithm are discussed for a better implementation of the proposed method.

1.3 Organization

The rest of this article is organized as follows: The next section 2 discusses the framework of the proposed synchronous DMPC algorithm. The improved design of the terminal ingredients will be explained in detail in this part. Some useful theoretical qualitative analysis on how to give the initial guess of the algorithm and how to set the relevant parameters are discussed in the section 3. Then, the proposed method is simulated with MATLAB programs to verify its effectiveness, while comparison with the existing results will also be given in this part. Finally, the conclusion of this article is provided.

2 Synchronous DMPC Algorithm

In the DMPC algorithm, each UAV in a swarm can obtain its collision-free trajectory by solving its optimal control problem. In this process, in order to meet the minimum safe distance constraints, UAVs need to obtain flight path information from other UAVs around, which is usually completed in the rolling time domain. Figure 1 gives an illustration of the rolling horizon policy.

As shown in Fig. 1, for the synchronous DMPC algorithm, all the UAVs update their predictions simultaneously at each sampling time. The influence of the uncertainty deviation between the actual trajectory and the planned trajectory should be eliminated by using the additional compatibility constraints and terminal ingredients.

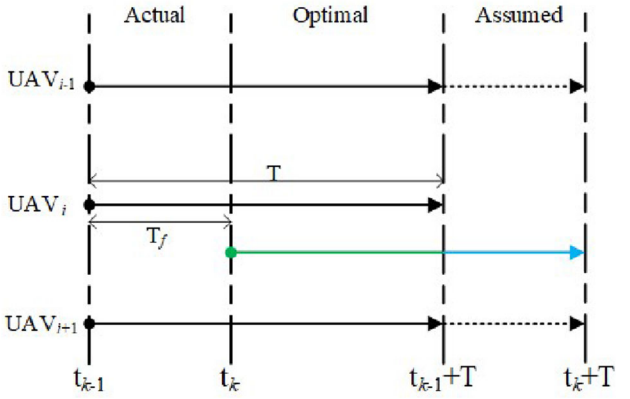

Fig. 1 Illustration of the rolling horizon policy. At any update time t_k, the UAVs transmit the related information with each other. For $\tau \in [t_k, t_{k-1} + T)$, the information can be achieved directly from the optimized results in the last iteration. For $\tau \in [t_{k-1} + T, t_k + T]$, the information should be achieved by predicting the flight status of each UAV

This chapter will introduce the overall framework of using the synchronous DMPC algorithm to solve the collision avoidance problem among multiple UAVs. In particular, an improved terminal ingredient design will be introduced in detail, which is also one of the main contributions of this article.

2.1 The DMPC Problem

The computationally tractable optimization problem for each UAV in a fleet can usually be formulated as following:

The DMPC Problem: At any update time t_k, given the initial state $\mathbf{x}_i(t_k)$, the desired state $\mathbf{x}_i^d$ for the UAV i and the assumed state of other UAVs $\{\hat{\mathbf{x}}_j\}_{j\in\mathcal{N}_i}$ (the set $\mathcal{N}_i$ contains all the neighbours of UAV i), the following MPC optimization problem is solved.

$$J_i^*(t_k, \mathbf{x}_i^*, \mathbf{x}_i^d, \mathbf{u}_i^*) = \min_{\mathbf{u}_i(\tau|t_k)} J_i(t_k, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i), \tag{1}$$

subject to for all $\tau \in [t_k, t_k + T]$,

$$\mathbf{x}_i(t_k|t_k) = \mathbf{x}_i(t_k), \tag{2}$$
$$\dot{\mathbf{x}}_i(\tau|t_k) = \mathbf{A}_i\mathbf{x}_i(\tau|t_k) + \mathbf{B}_i\mathbf{u}_i(\tau|t_k), \tag{3}$$
$$\mathbf{x}_i(\tau|t_k) \in \mathbb{X}_i, \tag{4}$$
$$\mathbf{u}_i(\tau|t_k) \in \mathbb{U}_i, \tag{5}$$
$$\left\|\hat{\mathbf{p}}_j(\tau|t_k) - \mathbf{p}_i(\tau|t_k)\right\| \geq 2R + \mu_{ij}(\tau|t_k), \forall j \in \mathcal{N}_i, \tag{6}$$
$$\left\|\hat{\mathbf{p}}_i(\tau|t_k) - \mathbf{p}_i(\tau|t_k)\right\| \leq \mu_i(\tau|t_k), \tag{7}$$
$$\mathbf{x}_i(t_k + T|t_k) \in \mathbb{X}_{if}. \tag{8}$$

For better understanding, Eqs. (1)–(8) are explained one by one as follows.

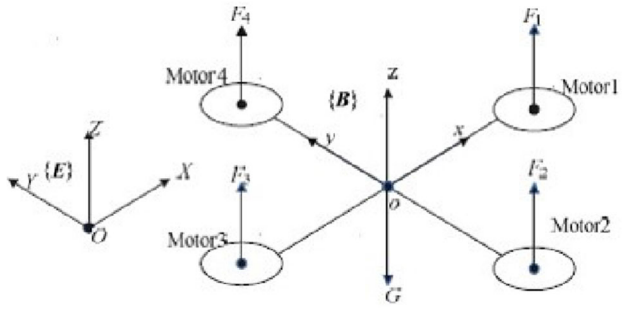

Fig. 2 The structure model of the quadcopter. Two basic coordinate systems are established: the inertial coordinate system E(OXYZ) and aircraft coordinate system B(oxyz). The heading angle ψ denotes the angle between the projection of ox on OXY plane and X-axis. The pitch angle θ denotes the angle between the projection of oz on OXZ plane and Z-axis. The roll angle ϕ denotes the angle between the projection of oy on OYZ plane and Y-axis

In the above problem, the cost function (Eq. (1)) is defined as

$$J_i(t_k, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i) = \int_{t_k}^{t_k+T} L_i(\tau|t_k, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i)\mathrm{d}\tau + V_{if}\left(\mathbf{x}_i(t_k+T|t_k), \mathbf{x}_i^d\right), \tag{9}$$

with the stage cost

$$\begin{aligned} L_i(\tau|t_k, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i) &= \left\|\mathbf{x}_i^d - \mathbf{x}_i(\tau|t_k)\right\|_{\mathbf{Q}_i}^2 + \|\mathbf{u}_i(\tau|t_k)\|_{\mathbf{R}_i}^2 \\ &= \|\Delta\mathbf{x}_i(\tau|t_k)\|_{\mathbf{Q}_i}^2 + \|\mathbf{u}_i(\tau|t_k)\|_{\mathbf{R}_i}^2, \end{aligned} \tag{10}$$

and the terminal cost $V_{if}\left(\mathbf{x}_i(t_k+T|t_k), \mathbf{x}_i^d\right)$, where $\mathbf{Q}_i$ is a given positive definite matrix and $\mathbf{R}_i$ is a given positive semi-definite matrix, $\Delta\mathbf{x}_i(\tau|t_k) = \mathbf{x}_i^d - \mathbf{x}_i(\tau|t_k)$. The terminal cost is a continuous, differentiable function to be designed in the subsequent part.

In this paper, the quadcopters, which can be commonly seen in low-altitude airspace, are selected as the research objects (shown in Fig. 2).

It is assumed that all the quadcopters in the fleet have the same dimensions and structure. The 2-dimensional dynamic model (Eq. (3)) of each agent can be expressed as following:

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ &= \begin{bmatrix} 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&g&0&0&0&0&0 \\ 0&0&0&-g&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix}\mathbf{x}_i + \begin{bmatrix} 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 0&0 \\ 1&0 \\ 0&1 \end{bmatrix}\mathbf{u}_i, \end{aligned} \tag{11}$$

in which g is the acceleration of gravity, $\mathbf{x}_i = \left[x_i, y_i, \theta_i, \phi_i, \dot{x}_i, \dot{y}_i, \dot{\theta}_i, \dot{\phi}_i\right]^T \in \mathbb{R}^8$ and $\mathbf{u}_i = \left[U_{i\theta}, U_{i\phi}\right]^T \in \mathbb{R}^2$, $U_{i\theta}$ is the pitch control input, $U_{i\phi}$ is the rolling control input.

$$\begin{bmatrix} U_{i\theta} \\ U_{i\phi} \end{bmatrix} = \begin{bmatrix} (F_4 - F_2)l/I_x \\ (F_3 - F_1)l/I_y \end{bmatrix}, \tag{12}$$

where F_i $(i = 1, 2, 3, 4)$ are the thrusts generated by the four motors, I_x and I_y are the moment of inertia in the two directions and l is the characteristic length of the UAV.

It should be noted that the pair $(\mathbf{A}_i, \mathbf{B}_i)$ is controllable. Therefore, existing $\mathbf{u}_i = \mathbf{K}_{i1}\mathbf{x}_i$, $\mathbf{K}_{i1} \in \mathbb{R}^{2\times 8}$, such that all the eigenvalues $\lambda(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_{i1})$ lie strictly in the left half open region of the complex plane.

Equation (4) gives the state saturation and $\mathbb{X}_i$ is a compact set. Equation (5) gives the control input saturation and $\mathbb{U}_i$ is a closed set.

To guarantee the satisfaction of the collision avoidance, Eq. (6) is included. $\mathbf{p}_i = [x_i, y_i]^T$ and

$$\mu_{ij}(\tau|t_k) = \frac{\left\|\hat{\mathbf{p}}_j(\tau|t_k) - \hat{\mathbf{p}}_i(\tau|t_k)\right\| - 2R}{2}, \tag{13}$$

in which the hat symbol implies that the information is assumed and $2R$ is the minimum safety distance between 2 UAVs in the problem.

An additional position compatibility constraints (Eq. (7)) is included in the problem to ensure the consistency between the real position and the assumed position of the UAV. In Eq. (7),

$$\mu_i(\tau|t_k) = \min_{j\in\mathcal{N}_i} \mu_{ij}, \tag{14}$$

Finally, Eq. (8) is the terminal constraints to be designed and $\mathbb{X}_{if} \subseteq \mathbb{X}_i$.

2.2 Design of the Terminal Ingredients

The terminal ingredients (including terminal controller, terminal cost, and terminal set) are of great significance to ensure recursive feasibility and closed-loop stability.

2.2.1 Terminal Controller

In the synchronous update scheme, all the UAVs in the fleet optimize simultaneously at each time step. Therefore, an assumed state should be introduced so that it can be transmitted among the UAVs instead of the real one.

The assumed control input is defined as

$$\hat{\mathbf{u}}_i(\tau|t_k) = \begin{cases} \mathbf{u}_i^*(\tau|t_{k-1}) & \tau \in [t_k, t_{k-1}+T) \\ \boldsymbol{\kappa}_i(\mathbf{x}_i^\kappa(\tau|t_{k-1})) & \tau \in [t_{k-1}+T, t_k+T], \end{cases} \tag{15}$$

where $\boldsymbol{\kappa}_i(\mathbf{x}_i^{\kappa}(\tau|t_{k-1}))$ is the terminal controller and $\mathbf{x}_i^{\kappa}(\tau|t_{k-1})$ is the state obtained along with Eq. 11 using the control input $\boldsymbol{\kappa}_i$ from the initialization $\mathbf{x}_i^*(t_{k-1}+T|t_{k-1})$.

In this paper, the design of the terminal controller adopts a traditional method. It is designed as

$$\boldsymbol{\kappa}_i(\mathbf{x}_i^{\kappa}(\tau|t_{k-1})) = \mathbf{K}_{i1}\mathbf{x}_i^{\kappa}(\tau|t_{k-1}) + \mathbf{K}_{i2}\mathbf{x}_i^d. \tag{16}$$

Substituting it into Eq. (11) and using the definition of $\Delta\mathbf{x}_i$, the error system

$$\begin{aligned}\Delta\dot{\mathbf{x}}_i(\tau) &= -\dot{\mathbf{x}}_i(\tau)\\ &= \mathbf{A}_i\Delta\mathbf{x}_i(\tau) - \mathbf{A}_i\mathbf{x}_i^d - \mathbf{B}_i\mathbf{K}_{i1}\mathbf{x}_i(\tau) - \mathbf{B}_i\mathbf{K}_{i2}\mathbf{x}_i^d\\ &= (\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_{i1})\Delta\mathbf{x}_i - (\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_{i1} + \mathbf{B}_i\mathbf{K}_{i2})\mathbf{x}_i^d.\end{aligned} \tag{17}$$

Therefore, if $\Phi_i = \mathbf{A}_i + \mathbf{B}_i\mathbf{K}_{i1}$ is stabilized and $(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_{i1} + \mathbf{B}_i\mathbf{K}_{i2})\mathbf{x}_i^d = \mathbf{0}$, the above equation can be further implified as

$$\Delta\dot{\mathbf{x}}_i(\tau) = \Phi_i\Delta\mathbf{x}_i(\tau). \tag{18}$$

Applying $\hat{\mathbf{u}}_i$, we can get the assumed state trajectory of agent i

$$\hat{\mathbf{x}}_i(\tau|t_k) = \begin{cases}\mathbf{x}_i^*(\tau|t_{k-1}) & \tau \in [t_k, t_{k-1}+T)\\ \mathbf{x}_i^{\kappa}(\tau|t_{k-1}) & \tau \in [t_{k-1}+T, t_k+T].\end{cases} \tag{19}$$

2.2.2 Terminal Cost

The terminal cost is designed as

$$V_{if}(\mathbf{x}_i, \mathbf{x}_i^d) = \left\|\mathbf{x}_i^d - \mathbf{x}_i\right\|_{\mathbf{P}_i}^2, \tag{20}$$

in which $\mathbf{P}_i$ is a positive definite matrix satisfying the Lyapunov equation

$$\mathbf{P}_i\boldsymbol{\Phi}_i + \boldsymbol{\Phi}_i^T\mathbf{P}_i + \mathbf{Q}_i + 2\mathbf{K}_{i1}^T\mathbf{R}_i\mathbf{K}_{i1} = \mathbf{Q}_0, \tag{21}$$

where $\mathbf{Q}_0$ is a given negative definite matrix.

It can be shown that V_{if} is a local control-Lyapunov function (refer to the Appendix for detailed proof) such that

$$\sum_{i\in\mathbb{N}_a}\left(\dot{V}_{if}(\mathbf{x}_i, \mathbf{x}_i^d) + L_i(t, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i)\right) \le 0, \forall\mathbf{x}_i \in \mathbb{X}_{if}. \tag{22}$$

2.2.3 Terminal Set

The main function of the terminal set $\mathbb{X}_{if}$ is to ensure the satisfaction of the collision avoidance constraints in the iterations. The design of the terminal set should be depend on the structure of the system model and the control objective.

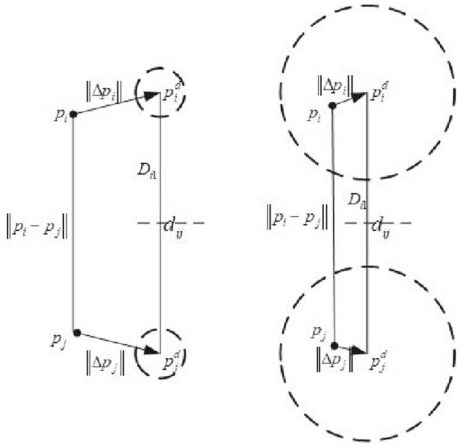

Fig. 3 Illustration of the terminal set $\mathbb{X}_{if1}$. In the left part, the desired positions of the two UAVs are farther apart, and the value of D_{i1} is larger. However, in the right part, the distance between the desired positions of the two UAVs is very close to the minimum value, and the value of D_{i1} is close to 0

A simple form of terminal set is proposed by Li Dai et al. as following

$$\mathbb{X}_{if1} = \left\{\mathbf{x}_i \middle| \left\|\mathbf{p}_i^d - \mathbf{p}_i\right\| \le D_{i1}\right\}, \tag{23}$$

in which

$$D_{i1} = \min_{j\in\mathcal{N}_i}\frac{\left\|\mathbf{d}_{ij}\right\| - 2R}{2}, \tag{24}$$

and $\mathbf{d}_{ij} = \mathbf{p}_j^d - \mathbf{p}_i^d$ denotes the desired relative position between UAV i and UAV j.

It has been strictly proven that $\mathbb{X}_{if1}$ is a positively invariant set with respect to the error system (Eq. (18)) and all states in this set definitely satisfy the collision avoidance constraint [23]. Besides, $\mathbb{X}_{if1}$ is only related to the position of each UAV, rather than the velocity.

However, the size of $\mathbb{X}_{if1}$ depends on the value of D_{i1}. As shown in Fig. 3, once the destinations of two UAVs are closely attached (the distance between them is approximated to the minimum safety distance $2R$), there will be a very small margin for $\mathbf{x}_i(t_k+T|t_k)$ to ensure the terminal constraints (Eq. (8)), which result in a degree of conservativeness of the trajectory.

Therefore, an improved terminal set is proposed as follows, so that UAVs can obtain a better collision-free trajectory when the desired positions of the UAVs are in dense:

$$\mathbb{X}_{if2} = \left\{\mathbf{x}_i \middle| \left\|\mathbf{p}_i - \frac{\mathbf{p}_i(t_{k-1}+T|t_k) + \mathbf{p}_i^d}{2}\right\| \le D_{i2}\right\}, \tag{25}$$

in which

$$D_{i2} = \min_{j \in \mathcal{N}_i} \frac{\left\| \mathbf{d}_{ij} + \mathbf{p}_j(t_{k-1}+T) - \mathbf{p}_i(t_{k-1}+T) \right\| - 4R}{4}. \tag{26}$$

The following proof can be used to show that the new-designed $\mathbb{X}_{if2}$ meets the system's requirements for the terminal set:

Theorem 1 *The terminal set $\mathbb{X}_{if2}$ is a positively invariant set with respect to the error system (Eq. (18)) when $t_k \to \infty$, and all states in this set satisfy the collision avoidance constraint.*

Proof of Theorem (1) (1) For any $\mathbf{p}_i \in \mathbb{X}_{if2}$ and $\mathbf{p}_j \in \mathbb{X}_{if2}$, $i \neq j$, according to the triangle inequality, we have

$$\begin{aligned}
&\left\| \mathbf{p}_i - \mathbf{p}_j \right\| \\
&= \left\| \left(\mathbf{p}_i - \frac{\mathbf{p}_i(t_{k-1}+T) + \mathbf{p}_i^d}{2} \right) - \left(\mathbf{p}_j - \frac{\mathbf{p}_j(t_{k-1}+T) + \mathbf{p}_j^d}{2} \right) \right. \\
&\quad \left. - \left(\frac{\mathbf{p}_j(t_{k-1}+T) + \mathbf{p}_j^d}{2} - \frac{\mathbf{p}_i(t_{k-1}+T) + \mathbf{p}_i^d}{2} \right) \right\| \\
&\geq \frac{1}{2} \left\| \mathbf{p}_j(t_{k-1}+T) - \mathbf{p}_i(t_{k-1}+T) + \mathbf{p}_j^d - \mathbf{p}_i^d \right\| \\
&\quad - \left\| \mathbf{p}_i - \frac{\mathbf{p}_i(t_{k-1}+T) + \mathbf{p}_i^d}{2} \right\| - \left\| \mathbf{p}_j - \frac{\mathbf{p}_j(t_{k-1}+T) + \mathbf{p}_j^d}{2} \right\| \\
&\geq \frac{1}{2} \left\| \mathbf{p}_j(t_{k-1}+T) - \mathbf{p}_i(t_{k-1}+T) + \mathbf{d}_{ij} \right\| \\
&\quad - \frac{1}{2} \left\| \mathbf{d}_{ij} + \mathbf{p}_j(t_{k-1}+T|t_k) - \mathbf{p}_i(t_{k-1}+T|t_k) \right\| + 2R \\
&\geq 2R
\end{aligned}$$

Therefore, the collision avoidance constraints hold for all the states in the terminal set $\mathbb{X}_{if2}$.

(2) With the designed terminal controller, we have $\Delta\dot{\mathbf{x}}_i(\tau) = \Phi_i \Delta\mathbf{x}_i(\tau)$. Therefore, the norm-bound on the error $\mathbf{x}_i^d - \mathbf{x}_i$ is monotonously non-increasing. Since the state set $\mathbb{X}_i$ is a compact set, $\left\| \mathbf{p}_i^d - \mathbf{p}_i(t_{k-1}+T|t_k) \right\|$ is non-increasing when $t_k \to \infty$. Therefore, $\left\| \mathbf{p}_i - \frac{\mathbf{p}_i(t_{k-1}+T|t_k)+\mathbf{p}_i^d}{2} \right\|$ is monotonously non-increasing when $t_k \to \infty$. As a consequence, the terminal set is a positively invariant set with respect to the error system.

Remark 1 From Eq. (26), there is a very important prerequisite for using the newly designed terminal set, that is

$$\left\| \mathbf{d}_{ij} + \mathbf{p}_j(t_{k-1}+T) - \mathbf{p}_i(t_{k-1}+T) \right\| \geq 4R. \tag{27}$$

In the UAV collision avoidance problem, $\left\| \mathbf{d}_{ij} \right\| \geq 2R$ is naturally satisfied. $\left\| \mathbf{p}_j(t_{k-1}+T) - \mathbf{p}_i(t_{k-1}+T) \right\| \geq 2R$ holds if the result of the last iteration is feasible. Therefore, satisfaction of Eq. (27) requires careful setting of initial guesses which will be discussed in the next chapter.

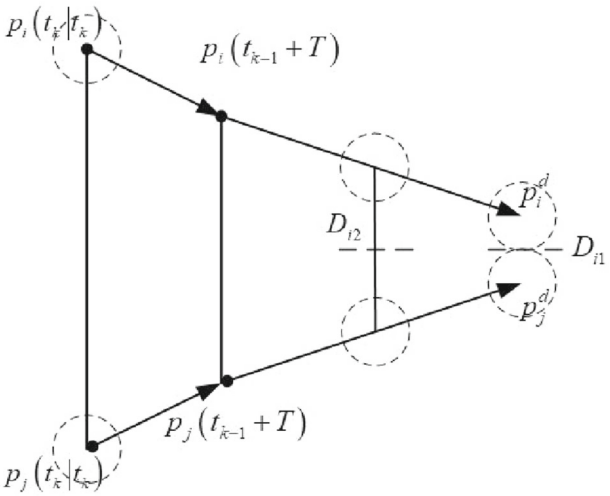

Fig. 4 Illustration of the terminal set $\mathbb{X}_{if2}$. The distance between the desired positions of the two UAVs are $2R$ (the required minimum distance). Therefore, the value of D_{i1} is 0 which implies the fixed final positions in each iteration of the algorithm with $\mathbb{X}_{if1}$. However, the margin for final positions to satisfy the terminal constraints in the algorithm with the new designed $\mathbb{X}_{if2}$ is more relaxed

Although the form of $\mathbb{X}_{if2}$ is relatively more complicated, the new-designed terminal set is still only related to the position of each UAV. As shown in Fig. 4, the size of D_{i2} is not only related to the desired position of each UAV, but also related to the final position within the last time interval. This change makes the selection of UAV final positions more flexible at the end of each iteration, and thereby improving the maneuverability of the overall collision-free trajectory.

This design provides the possibility of operation - when the desired positions of the UAVs are very close to the minimum safe distance, a set of final positions with relatively relaxed distances can be set in the initial guess first according to the environment and position conditions. Then, the UAVs gradually approaches the actual desired position as the iteration progresses. This operation can alleviate the conservativeness of the final trajectory.

2.3 The Complete Algorithm

Algorithm 1 gives the complete steps to solve the collision avoidance problem among multiple UAVs using the proposed synchronous DMPC problem.

The proof of the algorithm's recursive feasibility and closed-loop stability is similar to some existing research. See the Appendix for details. According to the above discussions and related proofs, the proposed algorithm is well converged and thus obtaining the complete collision-free trajectory once a set of initial feasible solutions is fixed.

According to Algorithm 1, it should be noted that a large number of optimal control problems (composed of Eqs. (1)–

Algorithm 1 The DMPC Algorithm for Collision Avoidance among Multiple UAVs

Require: For each agent i, given its parameters $\mathcal{N}_i$, d_{ij}, $\mathbf{x}_i^d$, $\mathbf{Q}_i$, $\mathbf{R}_i$, determine $\mathbf{K}_{i1}$, $\mathbf{K}_{i2}$, $\mathbf{P}_i$, D_{i2}.
1: At the initialisation time t_0, give the initial optimal input $\mathbf{u}_i^*(\tau|t_0)$ and the initial optimal state trajectory $\mathbf{x}_i^*(\tau|t_0)$ for $\tau \in [t_0, t_0+T]$;
2: Over the update interval $[t_k, t_{k+1})$, $k \geq 0$,
3: **for** each UAV i **do**
4: (i) Implements the actual trajectory $\mathbf{x}_i(t) = \mathbf{x}_i^*(t|t_k)$ for $t \in [t_k, t_{k+1})$;
5: (ii) Computes the assumed control input $\hat{\mathbf{u}}_i(\tau|t_{k+1})$ and the assumed state trajectory $\hat{\mathbf{x}}_i(\tau|t_{k+1})$ for $\tau \in [t_{k+1}, t_{k+1}+T]$;
6: (iii) Transmits $\hat{\mathbf{x}}_i(\cdot)$ to its neighbours and receives $\hat{\mathbf{x}}_j(\cdot)$ from every neighbours $j \in \mathcal{N}_i$;
7: **end for**
8: At each sampling time t_{k+1}, $k \geq 0$,
9: **for** each UAV i **do**
10: (i) Samples its current state $\mathbf{x}_i(t_{k+1})$ and computes $\mu_{ij}(\tau|t_{k+1})$, $\mu_i(\tau|t_{k+1})$;
11: (ii) Solves **The DMPC Problem** and obtains the optimal control input $\mathbf{u}_i^*(t|t_{k+1})$ and the optimal state trajectory $\mathbf{x}_i^*(t|t_{k+1})$ for $\tau \in [t_{k+1}, t_{k+1}+T]$;
12: **end for**
13: Return to ***Step.2***
Output: The collision-free trajectory $\mathbf{x}_i(t)$ and the corresponding control input.

(8) need to be solved for the final collision-free trajectory of UAV groups. In this study, the $\mathbb{GPOPS}-\mathbb{II}$ [27] solver is chosen to solve these optimal control problems.

3 The Initial Guess and Parameter Settings

In the actual process of using the DMPC algorithm to solve the collision avoidance problem between multiple UAVs, the initial feasible trajectory and the selection of some parameters are of great significance to ensure the recursive feasibility and the computational efficiency of the algorithm.

3.1 The Initial Feasible Trajectory

As mentioned above, if the algorithm adopts the first terminal set $\mathbb{X}_{if1}$, the final positions of the initial feasible trajectories need to be placed directly near the desired positions to satisfy Eq. (23) when the distance between any two desired positions of the UAVs is very close to the minimum safe distance. This is equivalent to planning a global trajectory at the beginning of the calculation. When the number of UAV increases, this is a very big challenge.

On the other hand, if the newly designed terminal set $\mathbb{X}_{if2}$ is adopted in the algorithm, although there is no need to directly plan the global trajectory, it does not mean that the final positions of the initial trajectories can be arbitrarily selected.

As discussed in the previous section, with the fix set of initial feasible solutions, the algorithm converges to a complete collision-free trajectory. The following are two basic principles that need to be followed when giving the initial feasible trajectories if $\mathbb{X}_{if2}$ is applied.

Principle 1. *The final positions of the initial feasible guess should satisfy the prerequisite (Eq. (27)) for using* $\mathbb{X}_{if2}$.

The importance of the prerequisite (Eq. (27)) has been stated in the last chapter. If it is not satisfied, the terminal set $\mathbb{X}_{if2}$ will be an empty set ($D_{i2} < 0$ and Eq. (25) can not be satisfied) and no feasible trajectories can be achieved through the algorithms.

Principle 2. *It is necessary to ensure that the midpoints between the final positions of the initial guess trajectories and the expected positions of the UAV are reachable in the next iteration interval.*

The proposed terminal set $\mathbb{X}_{if2}$ actually uses the result of the previous iteration to limit the range of the final positions of the collision-free trajectories in the subsequent iteration. To ensure the satisfaction of Eq. (25), the midpoints between the final positions of the initial guess trajectories and the expected positions of the UAVs $(\mathbf{p}_i(t_{k-1}+T|t_k)+\mathbf{p}_i^d)/2$ must be reachable. Otherwise, the recursive feasibility of the algorithm will be destroyed.

It seems that the use of the proposed terminal set requires higher requirements for the initial feasible trajectories. However, it actually eliminates the demanding of directly designing the global feasible trajectories of all UAVs. In the scene where multiple UAVs meet and avoid collisions with each other, it provides the possibility to decompose and simplify the collision avoidance scenarios of UAVs. This can be reflected in subsequent simulation scenarios.

3.2 Parameter Settings

As described in Algorithm 1, the proposed method solve the collision avoidance problem using a rolling horizon policy.

As shown in Fig. 1, the look-ahead time, T, is the time in which each UAV knows the information of other UAVs. It defines the DMPC Problem of each single UAV. Time between two iterations $T_f = t_k - t_{k-1}$ (the frequency of computation) determines how long the UAVs may implements their optimized result. The settings of these two parameters have an important impact on the execution of the entire algorithm and the overall calculation load. However, in the existing literature, their selection is mainly based on experience or experimental comparison [26]. This article attempts to give a qualitative theoretical analysis of these parameter settings based on the theoretical discussion in the above chapters.

3.2.1 The Look-ahead Time

Traditionally, the look-ahead time is considered from the knowledge of the environment in every iteration and the maximum speed of the UAV that is the worst case for trajectory re-planning. Now, with the analysis of the synchronous DMPC algorithm in the previous chapter, this principle can be set more accurately.

Principle 3. In order to satisfy the constraints of terminal set, the look-ahead time should be in the set of

$$\mathbb{T} = \{T | \mathbf{x}_i(t_k + T|t_k) \in \mathbb{X}_f\}. \tag{28}$$

From this point of view, the setting of the look-ahead time is closely related to conditions such as the UAV's current speed and its input saturation. If the given look-ahead time is not long enough for the UAV to enter the terminal set at the end of the iteration, the effectiveness of the algorithm cannot be guaranteed. It is important especially when the terminal set $\mathbb{X}_{if1}$ is applied in the algorithms. On the other hand, if the look-ahead time is chosen to be very long, the computational load of the algorithm will be increased and the efficiency of the UAV will be reduced.

3.2.2 The Frequency

When setting the time interval between two iterations, there are generally some basic considerations. First of all, it should be settled to ensure that each UAV does not fly the whole trajectory computed in the previous iteration during the corresponding time (Otherwise, the compatibility constraint will lose its original effect.). Therefore,

$$T_f < T. \tag{29}$$

Besides, it should be larger than the computation time of each iteration

$$T_f > T_c, \tag{30}$$

and it can be easily understand from Fig. 1 that the computation time, T_c, should fulfill

$$T_c < T - T_f. \tag{31}$$

Fortunately, with the improvement of hardware capabilities, computing time has gradually become negligible.

In addition, the process of the UAV moving from the final position of the previous iteration to the terminal set of the next iteration is mainly completed by the terminal controller. The action time of the terminal controller is T_f. Therefore, the realization of principle 2 is actually closely related to the saturation of the UAV control inputs and the size of T_f.

It should also be noticed that the dynamic model of the UAV in this research is continuous rather than discrete. Therefore, the pseudospectral method is generally used to deal with the problem. The collocation points (e.g. LG points or LGR points) used in this method are usually not uniformly distributed. They are densely distributed at both ends of the time domain and sparsely distributed in the middle. Based on this feature, if the value of t_f enables the UAV to execute the trajectory generated in a piece with more collocation points, the trajectory of the entire UAV will be more accurate and safer.

4 Simulation

In this section, the proposed algorithm is simulated and verified by MATLAB programs in two scenarios. The scenario in Simulation I is quite simple (contains only two drones). Two different terminal set (i.e. $\mathbb{X}_{if1}$ and $\mathbb{X}_{if2}$) are used to solve the flight collision avoidance problem in this scenario. The effect of using different terminal sets (and the subsequent different settings of initial feasible guess and related parameters) on solving the collision avoidance problem will be discussed and analyzed in detail in this simulation. Simulation II with a 5-UAV scenario further illustrates the effectiveness of the proposed method for solving the collision avoidance problem among multiple UAVs.

4.1 Simulation I

In this simulation, the two different terminal sets will be used to solve the same collision avoidance problem, so as to show the advantages of the new-designed terminal set through comparison.

The main parameters of the quadrotor UAVs used in the simulations are listed in Table 1 with position constraints as $-4 \le x, y \le 2$.

The scenario of collision avoidance is illustrated in Fig. 5 and the initial and final state of the 2 UAVs are listed in Table 2. It is easy to find out that there will be a potential mid-air collision if no avoiding maneuver is conducted. In this simulation, the collision-free trajectories should be designed

Table 1 The system parameters in simulation I

UAV Parameters	Value
Gravity Coefficient (g, m/s^2)	9.8
Angle Constraints (deg)	$-24 \le \theta, \phi \le 24$
Velocity Limits (m/s)	$-4 \le \dot{x}, \dot{y} \le 4$
Angular Velocity Limits (deg/s)	$-7 \le \dot{\theta}, \dot{\phi} \le 7$
Input Saturation (m/s^2)	$-0.5 \le U_{i\phi}, U_{i\theta} \le 0.5$

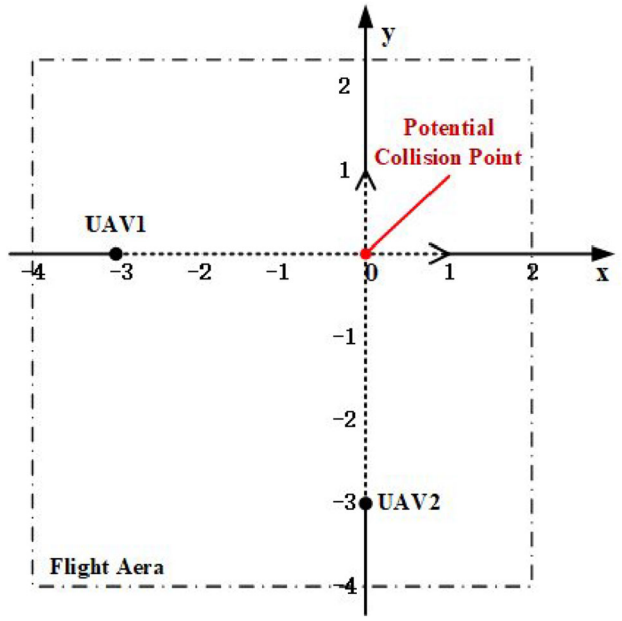

Fig. 5 The initial state in Simulation I. The UAV1 is located at $(-3\text{ m}, 0)$ with the velocity of $(0.3\text{ m/s}, 0)$. The UAV2 is located at $(0, -3\text{ m})$ with the initial velocity of $(0, 0.3\text{ m/s})$. It is clear that there will be a colliision at the point $(0, 0)$ if no action is conducted. The desired final positions of the two UAVs are $(1\text{ m}, 0)$ and $(0, 1\text{ m})$, respectively

Table 2 The initial and final state of the UAVs in Simulation I

UAV parameters	The initial value	The final value
UAV1 position	$(-3\,\text{m}, 0)$	$(1\,\text{m}, 0)$
UAV1 velocity	$(0.3\,\text{m/s}, 0)$	$(0, 0)$
UAV2 position	$(0, -3\,\text{m})$	$(0, 1\,\text{m})$
UAV2 velocity	$(0, 0.3\,\text{m/s})$	$(0, 0)$

for both of the UAVs to avoidance the potential collision. The minimum safety distance between the two UAVs is set to be $R = 0.25$ m.

4.1.1 Using the Algorithm with Terminal Set 1

According to the information of the desired final positions of the two UAVs in Table 2, it can be achieved that

$$D_{i1} = \frac{\|\mathbf{d}_{12}\| - 2R}{2} = 0.4571\text{ m}. \tag{32}$$

As discussed in the previous chapter, the look-ahead time needs to ensure that the UAV can be in the terminal set at the end of each optimization interval. It can usually be assumed that the collision-free optimal trajectory of the UAV is mainly adjusted locally based on the original trajectory to avoid potential collisions. Therefore, the average speed of the drone flying to the target point is roughly maintained near the initial speed. For this reason, when using the terminal set with D_{i1} to solve the problem, it is reasonable to set the look-ahead time T around 12 to 14 s. Therefore, $T_1 = 12$ s is selected in this simulation.

The calculation time of the algorithm is negligible. The time interval between two iterations T_f is set to be 1/10 of the look-ahead time.

To speed up the convergence of the algorithm, the initial guess of feasible trajectories is provided as follows: UAV1 keeps moving in a straight line towards the target position (its speed can be adjusted) while UAV2 avoids the potential collisions by moving sideways at the same height.

The simulation is implemented in MATLAB with the toolbox $\mathbb{GPOPS} - \mathbb{II}$ [27]. The collision avoidance process in this scenario is shown in Fig. 6. The two UAVs avoid the potential mid-air collisions successfully.

To verify if the achieved collision-free trajectories meet the safety constraints more clearly, Fig. 7 shows the distance between the 2 UAVs. The minimum distance between the two UAVs is larger than 0.5 m (the red line shown in Fig. 7) during the entire collision avoidance process. Besides, Fig. 8 also shows that the input of the UAVs meet the saturation requirements.

The closed-loop stability is further illustrated by Fig. 9, in which the object value of the two UAVs are shown. Both the curves decrease monotonically to zero.

4.1.2 Using the Algorithm with Terminal Set 2

By using the terminal set $\mathbb{X}_{if2}$ to solve the problem, the final position of the initial guess is first chosen.

In this simulation, $(0, -2\text{ m})$ is selected as the final position for UAV1 as its initial guess to start the first iteration, and $(0.5\text{ m}, 0)$ is selected as the final position for UAV2 as its initial guess to start the first iteration. Both of the choices satisfy the principles proposed in the previous section for selecting the initial guess. Besides, both UAVs can move to their right positions to avoid potential flight conflict.

Therefore, the radius of the terminal set with D_{i2} for the first iteration is

$$D_{i2} = \frac{\|\mathbf{d}_{12} + \mathbf{p}_1(T|0) - \mathbf{p}_2(T|0)\| - 4R}{4} = 0.5129\text{ m}. \tag{33}$$

Since the flight distance of the initial guess is shortened compared with the previous condition, the look-ahead time can be reduced to about 8 to 10 s accordingly. $T_2 = 8$ s is selected in this simulation. The shortening of the interval length will reduce the time required for each iteration and improve the operating efficiency of the algorithm. This is also one of the benefits of the newly designed terminal set in this article.

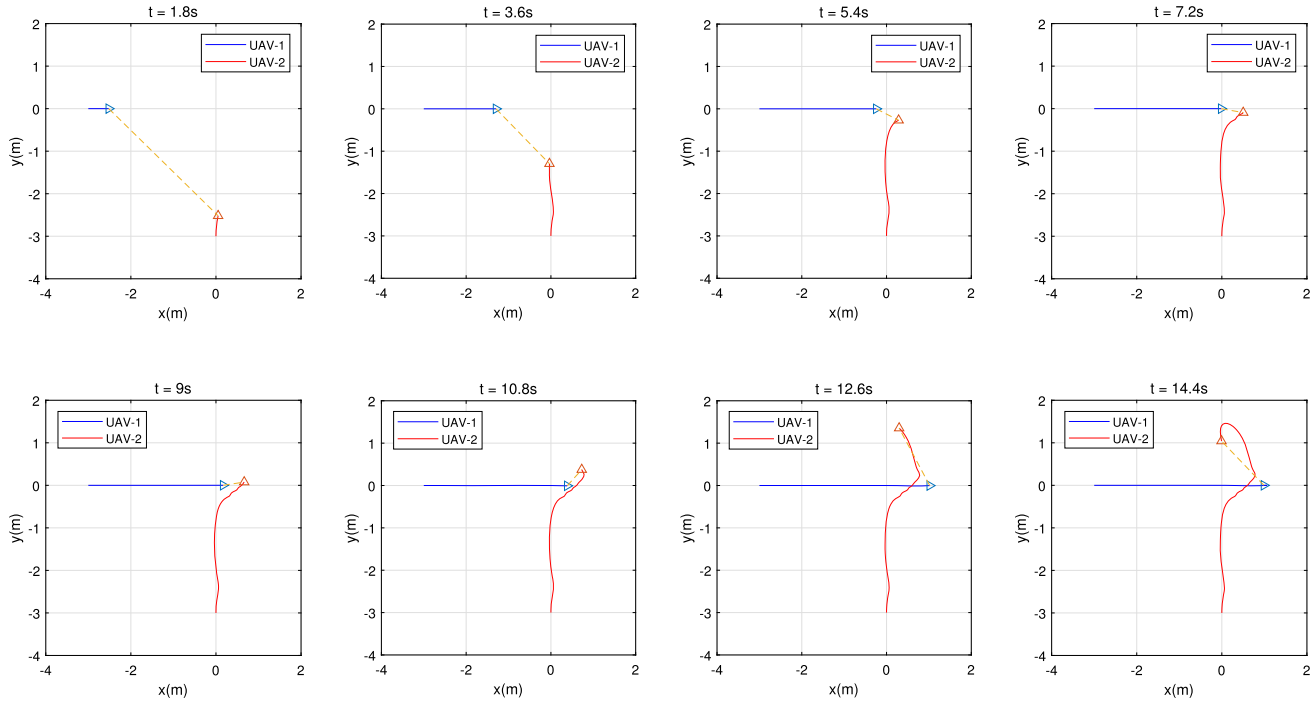

Fig. 6 The collision avoidance process for Simulation I using the terminal set $\mathbb{X}_{if1}$. The locations and the trajectories of the 2 UAVs are shown in this figure at 8 different time points (t = 1.8 s, 3.6 s, 5.4 s, 7.2 s, 9 s, 10.8 s, 12.6 s and 14.4 s). The distance between the two UAVs is indicated by a dashed line

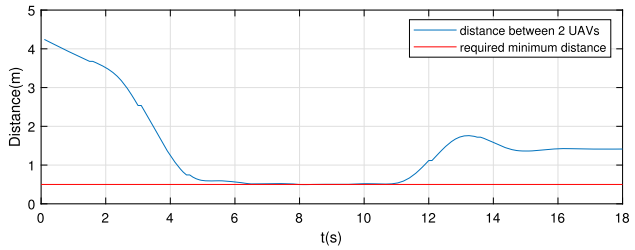

Fig. 7 The change of distance between the two UAVs in Simulation I using the terminal set $\mathbb{X}_{if1}$. It shows that the two UAVs maintained a distance greater than the required minimum distance during the entire collision avoidance process

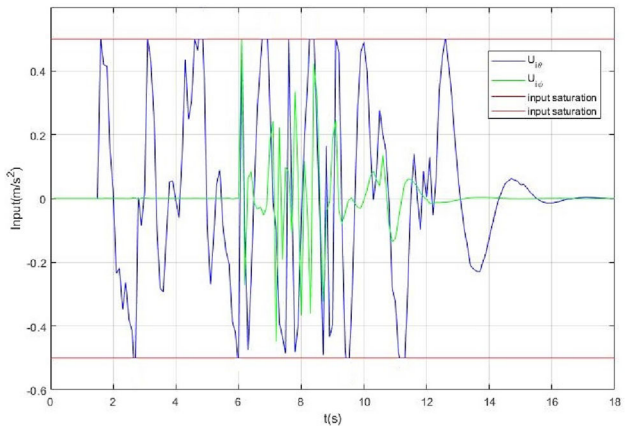

Fig. 8 The change of inputs in Simulation I using the terminal set $\mathbb{X}_{if1}$. It shows that they roughly meet the control input saturation requirements

By solving the corresponding DMPC problem, the collision avoidance process in this scenario with the new designed terminal set is shown in Fig. 10. The two UAVs also avoid the potential mid-air collisions successfully.

Figure 11 shows the distance between the 2 UAVs. The minimum distance between the two UAVs is larger than 0.5 m (the red line shown in Fig. 11) during the entire collision avoidance process. Besides, Fig. 12 also shows that the input of the UAVs meet the saturation requirements.

The closed-loop stability is illustrated by Fig. 13, in which the object value of the two UAVs are given. The object value of both UAVs decrease monotonically to zero, which implies the stability of the total algorithms.

4.1.3 Comparison of the Results.

Comparing the collision-free trajectories obtained by using different terminal sets, some interesting conclusions can be found.

Comparing Figs. 6 and 10, it can be clearly found that the flight trajectories of the UAVs in Fig. 10 are more flexible than those in Fig. 6 and there is not much overshoot. This is mainly credit to the two different terminal sets with different restrictions on the final positions in each iterations. Once the

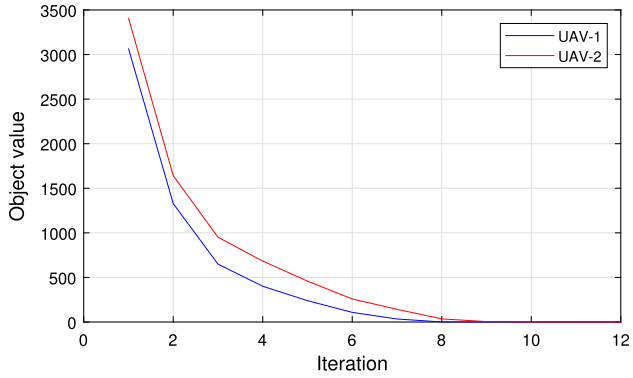

Fig. 9 The value of the cost function achieved by using the terminal set $\mathbb{X}_{if1}$ is illustrated. The object value decreases monotonically to zero which shows the closed-loop stability of the system

desired positions of the two UAVs are close to each other, the terminal set $\mathbb{X}_{if1}$ would require the algorithm to ensure a global trajectory of the UAV from the first iteration, then the final position of the UAV is restricted to a small margin. In contrast, terminal set $\mathbb{X}_{if2}$ gives the UAVs larger feasible ranges, resulting in more flexible trajectories.

It can also be witnessed from Figs. 9 and 13 that once the algorithm adopts the new designed terminal set $\mathbb{X}_{if2}$, the value of the cost function of the UAV optimization problem during each iteration is much lower than the value of leveraging the algorithm with terminal set $\mathbb{X}_{if1}$. The iterations required for the algorithms to converge to the final result are similar, while the computational efficiency of the proposed algorithm with $\mathbb{X}_{if2}$ is greatly improved compared with the original algorithm.

The average calculation time for each UAV of the proposed algorithm is 34 s, which is only 20% of the original algorithm (about 168 s).

The simulation verifies the superiority of the proposed algorithm by comprehensive comparisons.

4.2 Simulation II

The scenario of collision avoidance is illustrated in Fig. 14 and the initial and final positions of the 5 UAVs are listed in Table 3. This is a typical scenario for multi-UAV to change formation. It is easy to find out that there will be a potential mid-air collision if no avoiding maneuver is conducted. In this simulation, the minimum safety distance between the two UAVs is set to be $R = 0.2$ m. The related parameters of the simulations are the same with Simulation I (listed in Table 1) except that the position constraints are changed to $-4 \leq x, y \leq 4$. Only the algorithm with the new-designed terminal set $\mathbb{X}_{if2}$ is applied in this simulation.

Again, the final positions of the initial guess are first chosen. As discussed in the previous section, if the terminal set $\mathbb{X}_{if1}$ is applied, the restrictions of the final positions require the overall initial feasible guesses. Since the virtual

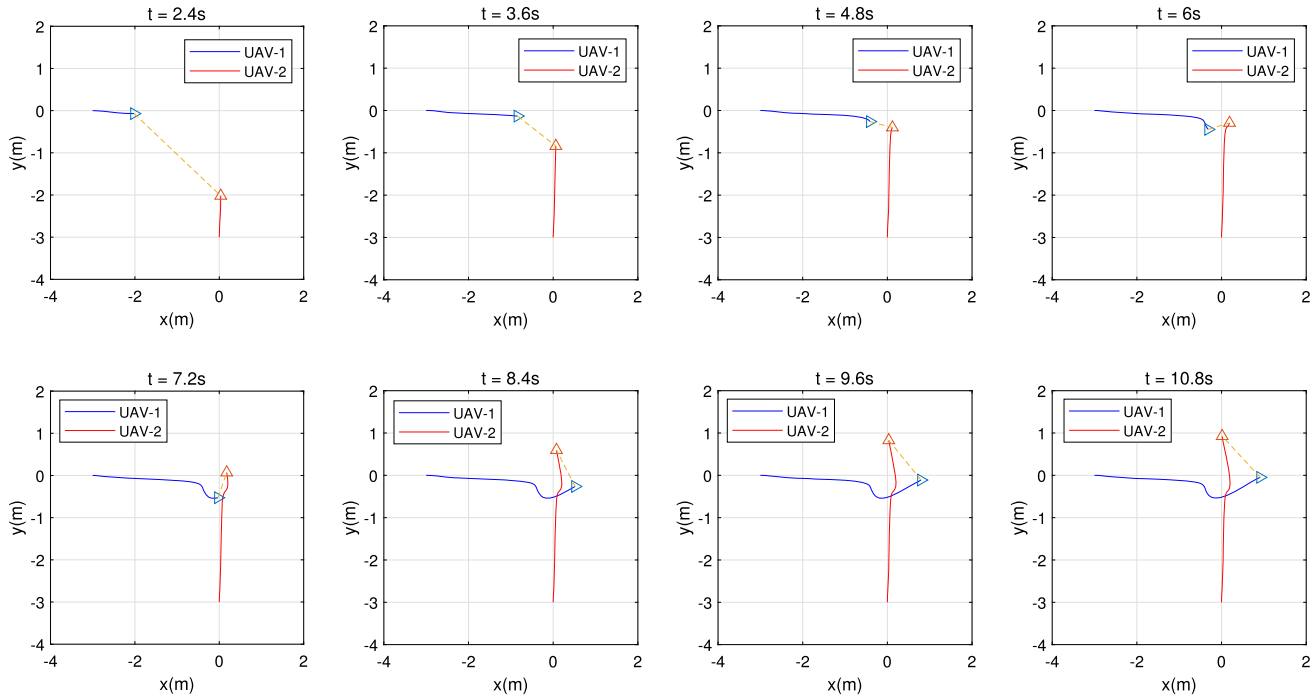

Fig. 10 The collision avoidance process for Simulation I using the terminal set $\mathbb{X}_{if2}$. The locations and the trajectories of the 2 UAVs are shown in this figure at 8 different time points (t = 2.4 s, 3.6 s, 4.8 s, 6 s, 7.2 s, 8.4 s, 9.6 s and 10.8 s). The distance between the two UAVs is indicated by a dashed line

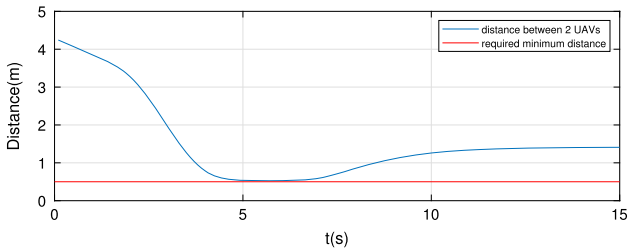

Fig. 11 The change of distance between the two UAVs in Simulation I using the terminal set $\mathbb{X}_{if2}$. It shows that the two UAVs maintained a distance greater than the required minimum distance during the entire collision avoidance process

Table 3 The initial and final positions of the UAVs in Simulation II

UAV	The initial position	The final position
UAV1	(− 3 m, 2 m)	(3.5 m, − 2 m)
UAV2	(− 3 m, 1 m)	(3.5 m, − 1 m)
UAV3	(− 3 m, 0)	(3.5 m, 0)
UAV4	(− 3 m, − 1 m)	(3.5 m, 1 m)
UAV5	(− 3 m, − 2 m)	(3.5 m, 2 m)

linear trajectories of the five UAVs converge at one point, all the UAVs may take the collision avoidance actions near the point (0.5 m, 0). Therefore, it will take high computational complexity and large amount of calculation to achieve the collision-free trajectories. However, with the terminal set $\mathbb{X}_{if2}$, the complex collision avoidance scenario can be decomposed and simplified by reasonably setting the initial feasible trajectory and final positions of each UAV.

In this simulation, the 5 UAVs are arranged to cross the x-axis from different positions at the same (or different) time. This settlement greatly simplifies the difficulty of solving the problem. The look-ahead time in this simulation is set to be $T = 10$ s and T_f is set to be 1 s.

The simulation is still implemented in MATLAB with the toolbox $\mathbb{GPOPS} - \mathbb{II}$. The collision avoidance process in this scenario is shown in Fig. 15. All the UAVs eventually arrive at their desired positions and avoid the potential mid-air collisions successfully.

Due to the space limitation, the UAV1 is taken as an example to show the guaranteed collision avoidance. As shown in Fig. 16, the relative distances between UAV1 and the other UAVs remain to be greater than the minimum safety distance $2R$ (which is depicted by the red line in Fig. 16)

The closed-loop stability is illustrated by Fig. 17, in which the object values of each UAV are shown. All the curves decrease monotonically to zero.

Therefore, the above simulation results verify the effectiveness of the proposed algorithms to solve the collision avoidance problems among multiple UAVs.

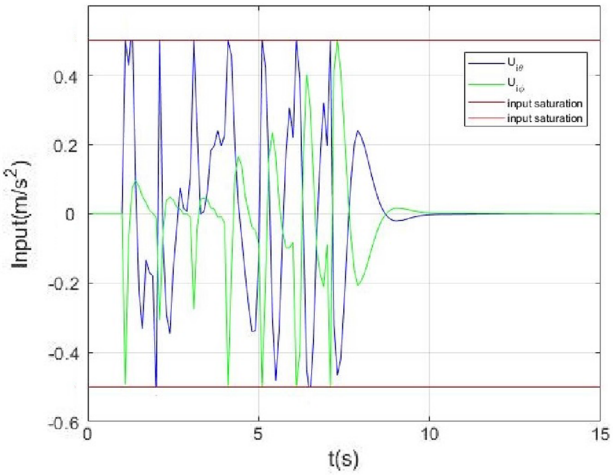

Fig. 12 The change of inputs in Simulation I using the terminal set $\mathbb{X}_{if2}$. It shows that they roughly meet the control input saturation requirements

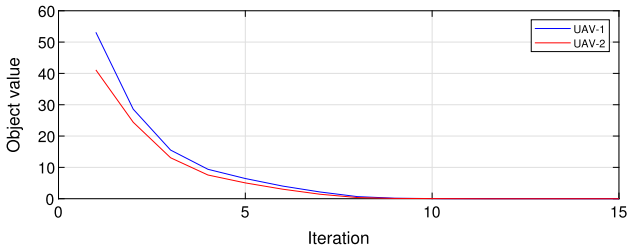

Fig. 13 The value of the cost function achieved by using the terminal set $\mathbb{X}_{if2}$ is illustrated. The object value decreases monotonically to zero which shows the closed-loop stability of the system

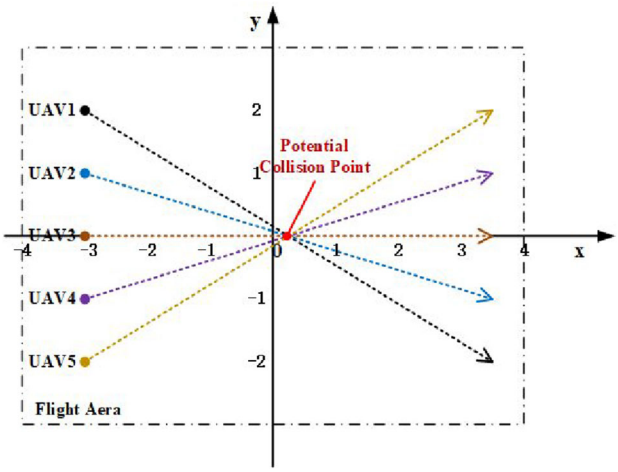

Fig. 14 The initial state in Simulation II. The 5 UAVs are located on the line of $x = -3$ m with the initial velocity of (0.5 m/s, 0). The distance between two adjacent drones is 1 m. The desired positions of the UAVs are on the line of $x = 3.5$ m and the position sequence is exactly opposite to the initial moment. It is clear that there will be a colliision at the point (0.5 m, 0) with all the UAVs if no action is conducted

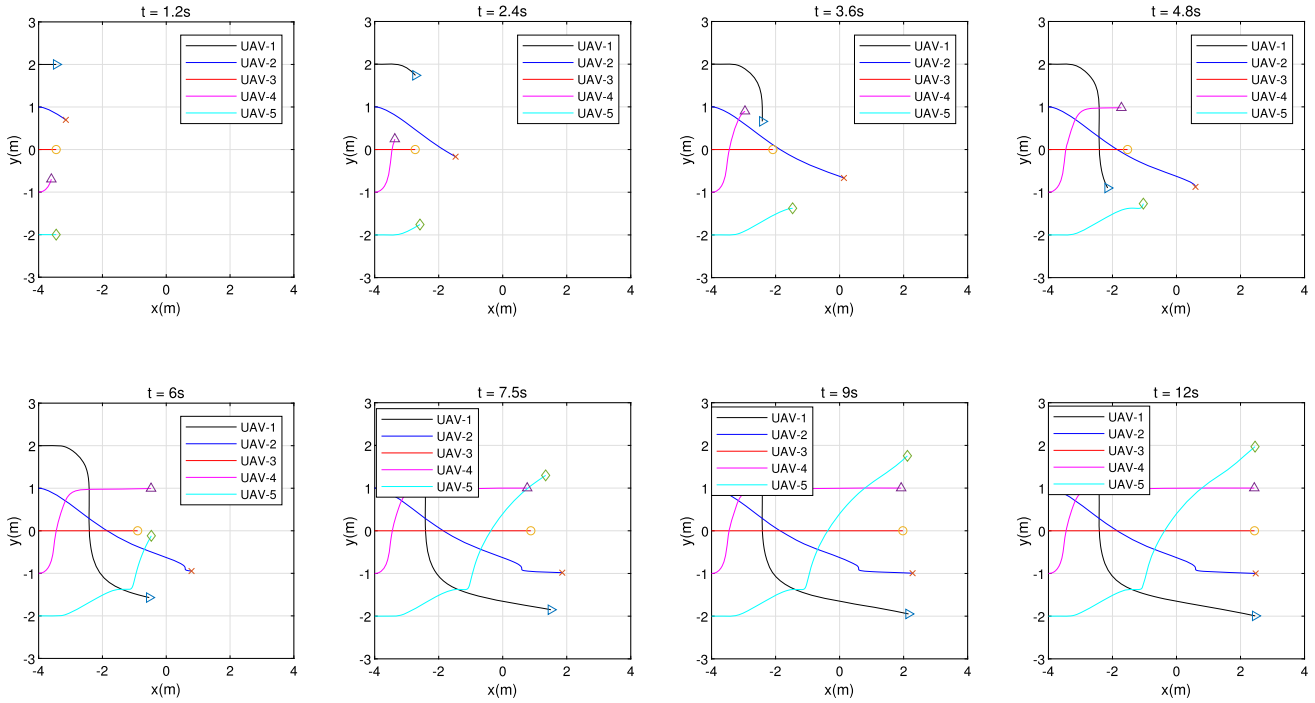

Fig. 15 The collision avoidance process for Simulation II using the terminal set $\mathbb{X}_{if2}$. The location and the trajectories of the 5 UAVs are shown in this figure at 8 different time points (t = 1.2 s, 2.4 s, 3.6 s, 4.8 s, 6 s, 7.5 s, 9 s and 12 s)

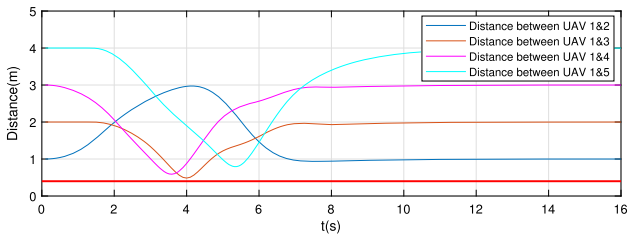

Fig. 16 The relative distances between the UAV1 and other UAVs in Simulation II. It shows that each pair of UAVs maintains a distance greater than the required minimum distance during the entire collision avoidance process

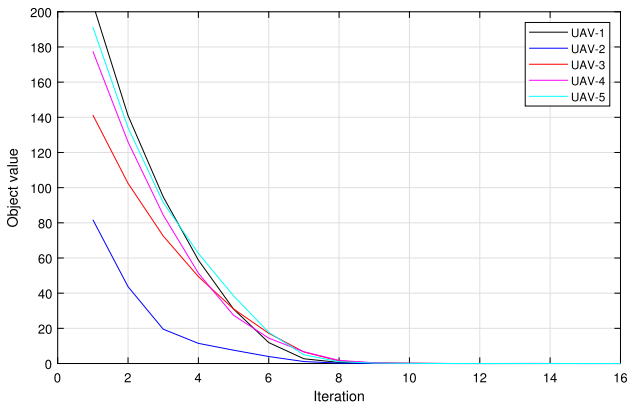

Fig. 17 The value of the cost function in Simulation II is illustrated. The object value decreases monotonically to zero which shows the closed-loop stability of the system

5 Conclusion

In this paper, an improved DMPC algorithm embedded with the specifically designed terminal ingredients are proposed to solve the problem of collision avoidance among multiple UAVs. It has been proved that the recursive feasibility and system stability remain under the improvements. Furthermore, some principles about how to set the initial feasible trajectries for the problem and how to set the related parameters are provided in this paper, which play the important roles in the implemention of the proposed method. Our future research are mainly foucsed on extending the proposed method to deal with the collision avoidance problem with more uncertainty.

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 China Aviation Science Foundation (20142057006) and the National Natural Science Foundation of China (61773262, 62006152).

Declarations

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

Appendix A A Lemma

Lemma: *The termial cost* $V_{if}(\mathbf{x}_i, \mathbf{x}_i^d)$ *designed in Eq.* (20) *is a local control-Lyapunov function satisfying the Eq.* (22).

Proof of Lemma Taking into consideration of the specific forms of $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{x}_i^d = [\mathbf{p}_i^{dT}\ \mathbf{0}]^T$, it is easy to find a suitable pair of $(\mathbf{K}_{i1}, \mathbf{K}_{i2})$ to meet the requirements of the terminal controller if the first 2 columns of $\mathbf{K}_{i2}$ is taken as the opposite numbers of that of $\mathbf{K}_{i1}$ (the other columns are unrestricted). Therefore, it easily follows that

$$\Lambda_{i2} = (\mathbf{K}_{i1} + \mathbf{K}_{i2})^T\, \mathbf{R}_i\, (\mathbf{K}_{i1} + \mathbf{K}_{i2}) = \mathbf{0}. \tag{A1}$$

Besides, if $\mathbf{P}_i$ is a positive definite solution to the Lyapunov equation (Eq. (21)), it follows immediately that

$$\Lambda_{i1} = \mathbf{P}_i \mathbf{\Phi}_i + \mathbf{\Phi}_i^T \mathbf{P}_i + \mathbf{Q}_i + 2\mathbf{K}_{i1}^T \mathbf{R}_i \mathbf{K}_{i1} < 0. \tag{A2}$$

Rewrite the terminal controller as

$$\boldsymbol{\kappa}_i(\mathbf{x}_i) = -\mathbf{K}_{i1}\Delta\mathbf{x}_i + (\mathbf{K}_{i1} + \mathbf{K}_{i2})\mathbf{x}_i^d, \tag{A3}$$

and use the inequality $\|\mathbf{x}_1 - \mathbf{x}_2\|_{\mathbf{Q}}^2 \le 2(\|\mathbf{x}_1\|_{\mathbf{Q}}^2 + \|\mathbf{x}_2\|_{\mathbf{Q}}^2)$. Then

$$\begin{aligned}
&\sum_i \left(\dot{V}_{if}(\mathbf{x}_i, \mathbf{x}_i^d) + L_i(t, \mathbf{x}_i, \mathbf{x}_i^d, \mathbf{u}_i)\right) \\
&= \sum_i \Big\{ \Delta\mathbf{x}_i^T \mathbf{P}_i \Delta\dot{\mathbf{x}}_i + \Delta\dot{\mathbf{x}}_i^T \mathbf{P}_i \Delta\mathbf{x}_i + \|\Delta\mathbf{x}_i(\tau|t_k)\|_{\mathbf{Q}_i}^2 \\
&\quad + \|\boldsymbol{\kappa}_i(\mathbf{x}_i)\|_{\mathbf{R}_i}^2 \Big\} \\
&\le \sum_i \Big\{ \Delta\mathbf{x}_i^T \mathbf{P}_i \mathbf{\Phi}_i \Delta\mathbf{x}_i + \Delta\mathbf{x}_i^T \mathbf{\Phi}_i^T \mathbf{P}_i \Delta\mathbf{x}_i + \|\Delta\mathbf{x}_i\|_{\mathbf{Q}_i}^2 \\
&\quad + 2\,\|\Delta\mathbf{x}_i\|_{\mathbf{K}_{i1}^T \mathbf{R}_i \mathbf{K}_{i1}}^2 + 2\left\|\mathbf{x}_i^d\right\|_{(\mathbf{K}_{i1}+\mathbf{K}_{i2})^T \mathbf{R}_i (\mathbf{K}_{i1}+\mathbf{K}_{i2})}^2 \Big\} \\
&\le \sum_i \left\{ \|\Delta\mathbf{x}_i\|_{\mathbf{\Lambda}_{i1}}^2 + 2\left\|\mathbf{x}_i^d\right\|_{\mathbf{\Lambda}_{i2}}^2 \right\} \le 0.
\end{aligned}$$

Appendix B The Proof of the Recursive Feasibility & the System Stability

Conclusion: At each sampling time t_k, all the subsystem with dynamic (11) solve their own optimization problems synchronously. If there exists a feasible solution at initial time t_0 for each UAV, then its optimization problem is feasible at any time t_k, $k \ge 1$. Furthermore, the whole system is asymptotically stable, i.e. for all UAVs, as $t \to \infty$, $\mathbf{x}_i \to \mathbf{x}_i^d$, and for $t \ge 0$, $\left\|\mathbf{p}_j(\tau|t_k) - \mathbf{p}_i(\tau|t_k)\right\| \ge 2R, \forall j \in \mathcal{N}_i$

Proof of the Conclusion (1) (**Feasibility of subsystem** i)

Let $\tilde{\mathbf{u}}(\tau|t_k)$ be the feasible input and $\tilde{\mathbf{x}}(\tau|t_k)$ be the feasible state trajectory for $\tau \in [t_k, t_k + T], k \ge 0$. If the optimization problem of UAV i at time t_k is feasible, from Eq. 15, we can get the assumed input $\hat{\mathbf{u}}(\tau|t_k)$ over $\tau \in [t_{k+1}, t_{k+1} + T]$. Let $\check{\mathbf{u}}(\tau|t_{k+1}) = \hat{\mathbf{u}}_i(\tau|t_{k+1})$ be the control input at time t_{k+1}. Accordingly, let $\check{\mathbf{x}}(\tau|t_{k+1}) = \hat{\mathbf{x}}_i(\tau|t_{k+1})$. It is easy to see that $\check{\mathbf{u}}(\cdot)$ and $\check{\mathbf{x}}(\cdot)$ satisfy the constraints (3)–(5) from t_{k+1} to $t_{k+1} + T$ (from the definition of the assumed control inputs). Furthermore, from Theorem 1, we can get that the terminal set is positive invariant and the states in the terminal set definitely satisfy the constraint (8). Hence, $\check{\mathbf{u}}(\cdot)$ and $\check{\mathbf{x}}(\cdot)$ can also ensure the satisfaction of (6) and (7) over the horizon $[t_{k+1}, t_{k+1} + T]$. To sum up, $\check{\mathbf{u}}(\cdot)$ is feasible for the optimization problem of UAV i at time t_{k+1} and the initial feasibility implies the recursive feasibility.

(2) (**Stability of the whole system**)

The Lyapunov function is defined as the sum of individual optimal cost functions, i.e.

$$J_{\sum}^*(t_k) = \sum_{i=1}^{N} J_i^*(t_k, \mathbf{x}_i^*, \mathbf{x}_i^d, \mathbf{u}_i^*). \tag{B1}$$

Denote $\tilde{\mathbf{u}}_i(\tau|t_{k+1}) = \hat{\mathbf{u}}_i(\tau|t_{k+1})$ as the feasible solution at t_{k+1}. Hence,

$$\begin{aligned}
&J_{\sum}^*(t_{k+1}) - J_{\sum}^*(t_k) \le \tilde{J}_{\sum}(t_{k+1}) - J_{\sum}^*(t_k) \\
&\le \sum_i \left(J_i(t_{k+1}, \tilde{\mathbf{x}}_i, \mathbf{x}_i^d, \tilde{\mathbf{u}}_i) - J_i^*(t_k, \mathbf{x}_i^*, \mathbf{x}_i^d, \mathbf{u}_i^*)\right) \\
&= \sum_i - \int_{t_k}^{t_{k+1}} L_i(\tau|t_k, \mathbf{x}_i^*, \mathbf{x}_i^d, \mathbf{u}_i^*)\mathrm{d}\tau \\
&\quad + \int_{t_k+T}^{t_{k+1}+T} L_i(\tau|t_k, \mathbf{x}_i^\kappa, \mathbf{x}_i^d, \boldsymbol{\kappa}_i)\mathrm{d}\tau \\
&\quad + V_{if}(\mathbf{x}_i^\kappa(t_{k+1} + T|t_k)) - V_{if}(\mathbf{x}_i^*(t_k + T|t_k)) \\
&= -\sum_i \int_{t_k}^{t_{k+1}} L_i(\tau|t_k, \mathbf{x}_i^*, \mathbf{x}_i^d, \mathbf{u}_i^*)\mathrm{d}\tau \le 0.
\end{aligned} \tag{B2}$$

References

1. Tahir A, Boling J, Haghbayan MH, Toivonen HT, Plosila J (2019) Swarms of unmanned aerial vehicles - a survey. J Ind Inf Integr 16. https://doi.org/10.1016/j.jii.2019.100106
2. Dubay S, Pan Y-J (2018) Distributed mpc based collision avoidance approach for consensus of multiple quadcopters. In: 2018 IEEE 14th International Conference on Control and Automation (ICCA), pp. 155–160. https://doi.org/10.1109/ICCA.2018.8444273
3. Hafner MR, Cunningham D, Caminiti L, Del Vecchio D (2013) Cooperative collision avoidance at intersections: Algorithms and experiments. IEEE Trans Intell Transp Syst 14(3):1162–1175. https://doi.org/10.1109/Tits.2013.2252901

4. Tan CY, Huang SN, Tan KK, Teo RSH (2020) Three dimensional collision avoidance for multi unmanned aerial vehicles using velocity obstacle. Journal of Intelligent & Robotic Systems 97(1):227–248
5. Wei ZL, Huang CQ, Ding DL, Huang HQ, Zhou H (2018) Ucav formation online collaborative trajectory planning using hp adaptive pseudospectral method. Mathematical Problems in Engineering **2018**
6. Tsourdos A, White B, Shanmugavel M (2010) Cooperative path planning of unmanned aerial vehicles. Wiley, New York
7. Leonard J, Savvaris A, Tsourdos A (2017) Distributed reactive collision avoidance for a swarm of quadrotors. Proceedings of the Institution of Mechanical Engineers Part G-Journal of Aerospace Engineering 231(6):1035–1055
8. Jiang Y, Hu S, Damaren CJ (2019) Collision avoidance algorithm between quadrotors using optimal control and pseudospectral method. In: AIAA Scitech Forum. AIAA SciTech Forum, p. 1415. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2019-1415
9. Jiang YH, Hu SQ, Damaren CJ (2021) Trajectory planning with mid-air collision avoidance for quadrotor unmanned aerial vehicles. Proceedings of the Institution of Mechanical Engineers Part G-Journal of Aerospace Engineering. https://doi.org/10.1177/09544100211044046
10. Radmanesh M, Kumar M (2016) Flight formation of uavs in presence of moving obstacles using fast-dynamic mixed integer linear programming. Aerosp Sci Technol 50:149–160
11. Cai JL, Zhang N (2019) Mixed integer nonlinear programming for aircraft conflict avoidance by applying velocity and altitude changes. Arab J Sci Eng 44(10):8893–8903
12. Mellinger D, Kushleyev A, Kumar V (2012) Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 477–483. Go to ISI://WOS:000309406700071
13. Desaraju VR, How JP (2012) Decentralized path planning for multi-agent teams with complex constraints. Auton Robot 32(4):385–403. https://doi.org/10.1007/s10514-012-9275-2
14. Augugliaro F, Schoellig AP, D'Andrea R (2012) Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In: 2012 IEEE/Rsj International Conference on Intelligent Robots and Systems (Iros), pp. 1917–1922. Go to ISI://WOS:000317042702073
15. Chen Y, Cutler M, How JP (2015) Decoupled multiagent path planning via incremental sequential convex programming. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5954–5961. Go to ISI://WOS:000370974905131
16. Robinson DR, Mar RT, Estabridis K, Hewer G (2018) An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments. Ieee Robotics and Automation Letters 3(2):1215–1222. https://doi.org/10.1109/Lra.2018.2794582
17. D'Amato E, Mattei M, Notaro I (2020) Distributed reactive model predictive control for collision avoidance of unmanned aerial vehicles in civil airspace. Journal of Intelligent & Robotic Systems 97(1):185-203
18. Wang P, Ding BC (2014) Distributed rhc for tracking and formation of nonholonomic multi-vehicle systems. IEEE Trans Autom Control 59(6):1439–1453. https://doi.org/10.1109/Tac.2014.2304175
19. Parys RV, Pipeleers G (2017) Distributed model predictive formation control with inter-vehicle collision avoidance. 2017 11th Asian Control Conference (Ascc), 2399–2404
20. Luis CE, Schoellig AP (2018) Trajectory generation for multiagent point-to-point transitions via distributed model predictive control. IEEE Robotics & Automation Letters 4(2):375–382
21. Wang P, Ding B (2014) A synthesis approach of distributed model predictive control for homogeneous multi-agent system with collision avoidance. Int J Control 87(1):52–63
22. Gao YL, Xia YQ, Dai L (2015) Cooperative distributed model predictive control of multiple coupled linear systems. IET Control Theory Appl 9(17):2561–2567. https://doi.org/10.1049/iet-cta.2015.0096
23. Dai L, Cao Q, Xia Y, Gao Y (2017) Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance. Journal of the Franklin Institute-Engineering and Applied Mathematics 354(4):2068–2085
24. Biggs B, Stilwell DJ, McMahon J (2020) Extended performance guarantees for receding horizon search with terminal cost. 2020 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros), 6741–6748. https://doi.org/10.1109/Iros45743.2020.9341582
25. Biggs B, McMahon J, Baldoni P, Stilwell DJ (2021) Multi-agent receding horizon search with terminal cost. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 9086–9092. https://doi.org/10.1109/ICRA48506.2021.9561643
26. Vera S, Cobano JA, Heredia G, Ollero A (2016) Collision avoidance for multiple uavs using rolling-horizon policy. Journal of Intelligent & Robotic Systems 84(1–4):387–396. https://doi.org/10.1007/s10846-015-0291-2
27. Patterson MA, Rao AV (2014) Gpops - ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. Acm Transactions on Mathematical Software **41**(1). https://doi.org/10.1145/2558904