Journal Title
XX(X):1–13
©The Author(s) 2020
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Trajectory Planning with Mid-air Collision Avoidance for Quadrotor Unmanned Aerial Vehicles

Yuhang Jiang[1], Shiqiang Hu[1] and Christopher J. Damaren[2]

Abstract

Flight collision between unmanned aerial vehicles (UAVs) in mid-air poses a potential risk to flight safety in low altitude airspace. This paper transforms the problem of collision avoidance between quadrotor UAVs into a trajectory-planning problem using optimal control algorithms, therefore achieving both robustness and efficiency. Specifically, the pseudospectral method is introduced to solve the raised optimal control problem, while the generated optimal trajectory is precisely followed by a feedback controller. It is worth noting that, the contributions of this paper also include the introduction of the normalized relative coordinate, so that UAVs can obtain collision-free trajectories more conveniently in real time. The collision-free trajectories for a classical scenario of collision avoidance between two UAVs are given in the simulation part by both solving the optimal control problem and querying the prior results. The scalability of the proposed method is also verified in the simulation part by solving a collision avoidance problem among multiple UAVs.

Keywords

UAV collision avoidance, trajectory planning, optimal control, flight safety

Introduction

With the rapid development of technologies, unmanned aerial vehicles (UAVs) have been recently applied in both military and civilian areas[1]. As a result, the flight amount of UAVs is increasing significantly, which brings many risks to flight safety in low-altitude airspace. Mid-air flight collision is one of the main flight accidents that occur frequently. Therefore, it is urgent to have a suitable UAV collision avoidance system, and the collision avoidance algorithm is the core of it.

The research focus of this paper is the collision avoidance problem between mid-air UAVs. Although cooperative collision avoidance method for more realistic situations has achieved many research in recent years[2–4], it is assumed that all information (such as its position and velocity) of the controlled UAV (the UAV we concerned) can be obtained through the sensing component for simplicity in this paper. The information of the target drone (other UAVs in the scene) can be achieved through air traffic control components (such as automatic dependent surveillance (ADS-B)).

Although the actual collision avoidance scene may usually contain more than two UAVs, solving collision avoidance problems between two UAVs is the basis for solving the collision avoidance problems among multiple UAVs. Therefore, the research of this paper will first focus on the collision avoidance problems among two UAVs. After that, the achieved conclusion will be used to solve the collision avoidance problems among multiple UAVs.

From the perspective of trajectory planning, the collision-free trajectory of a UAV is a flight path connecting the initial point and the finish point, which ensures the minimum distance between the UAV and other objects[5]. Many different philosophies have been studied to find such trajectories[6–8].

The sense-and-avoid algorithm, also called conflict detection and resolution, is one of the traditional methods to deal with the collision avoidance problem. It is mainly derived from the Traffic Collision Avoidance System (TCAS) used by the air traffic management for commercial airplanes[9,10]. In addition to the perception of the environment, the core idea of this method is the analysis of the relative motion relationship of two UAVs in geometric space[11], followed by certain decision processes such as Markov decision[12,13] and trajectory tracking algorithms[14,15]. This kind of method is simple, intuitive, and low cost. However, the resulting trajectories are not optimal. Furthermore, the method does not tell the UAV how to return to its original planned route which is quite important for UAV tasks.

The potential field algorithm has also been widely studied. In this algorithm, UAVs are set to move under the influence of an artificial potential field determined by the goal position and the obstacles. The research of this method usually focuses on how to set the parameters correctly, so as to ensure that the method can converge to the global optimal solution[16,17].

The optimal control method is also adopted by many researchers because of its ease of application and high scalability. It uses the idea of optimal control to transfer the

[1] School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, China
[2] Institute for Aerospace Studies University of Toronto, Toronto, Ontario, Canada

Corresponding author:
Shiqiang Hu, School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, 200240, China
Email: sqhu@sjtu.edu.cn

collision avoidance problem into a collision-free trajectory planning problem[18–21]. This kind of methods can yield an optimal result but require a large amount of calculation. Many different optimization algorithms, such as mixed integer linear programming (MILP)[22–24], sequential convex programming (SCP)[25,26], rolling optimization algorithm and model predictive control (MPC)[27–29], have been applied to effectively reduce the computational complexity of the method.

Therefore, each method has its most suitable application scenarios. Taking the advantages of different algorithms to find an algorithm suitable for UAV collision avoidance in low-altitude airspace is the goal of this research.

Motivated by the above analysis, an innovative method is proposed in this paper to give the collision-free trajectories for quadrotor UAVs. The optimal control is chosen as the core of the collision-avoidance algorithm because it can make an excellent trade-off between maintaining safe distances from other UAVs and decreasing the deviation from the original trajectory[30]. Pseudospectral discretization method has received more research interest in recent years. It leads to a significant improvement over the traditional Euler or Runge-Kutta methods because its high-accuracy quadrature can provide exponential (compared to polynomial) convergence rate[31–33]. Therefore, the pseudospectral method is applied in this paper to solve the optimal control problem to improve the computational efficiency of the algorithm. However, the cumulative error generated by the pseudospectral methods may lead to the divergence of trajectories under open loop control. Therefore, a feedback controller (e.g. an LQR controller which has been widely used in UAV systems[34–36]) is applied in this paper to ensure the optimal trajectories being precisely tracked.

Even with the above techniques, it is not an efficient approach to establish and solve a specific optimal control problem for each different collision avoidance scenario. Therefore, in this research, a method for calculating the collision-free trajectories by only querying the prior calculation results is proposed. First of all, a relative coordinate system is introduced in this paper to simplify the description of collision avoidance scenarios. Besides, we implemented normalization to eliminate the impact of UAV velocity on the minimum safety distance. In addition, considering the symmetry of the collision-free trajectory, the established optimal control problem is in the finite time domain rather than in the infinite time domain, which may bring a lot of convenience to the solution process.

The main contributions of this paper could be concluded by the following aspects:

1. A feasible optimal control method is proposed to solve the collision avoidance problem between mid-air quadrotor UAVs. The existence of the global optimal solution is also proved after considering an acceptable restriction.
2. To improve the computational efficiency, the pseudospectral method is applied in this research. A feedback controller is designed to deal with the trajectory deviation caused by the cumulative calculation error.
3. The normalized relative coordinate system is considered in this research to further improve the computational efficiency of proposed algorithm.

The rest of this paper is organized as follows. The upcoming section introduces the formulation of the optimal control problem. The main procedure of the pseudospectral methods, proof of the existence of the overall optimal solution, and the introduction of the LQR controller will also be included. The simplification of the collision avoidance scenario is introduced in detail in the subsequent section. Specifically, the symmetry of the collision-free trajectories will be proven. Finally, the proposed method is simulated with MATLAB programs to verify its effectiveness, and the conclusion of this paper is provided.

Establishing and Solving of the Optimal Control Problem

The collision-free trajectory is the basis to avoid the mid-air collision in the air. In this paper, these trajectories are obtained by transforming the collision avoidance problem into an optimal control problem under the constraint of the minimum safety interval. This section will introduce the process of establishing and solving the optimal control problem.

The Optimal Control Problem

The optimal control problem is to find the optimal control input $\mathbf{u}^*$, which makes the UAV follow an optimal trajectory $\mathbf{x}^*$ that minimizes an objective function J.

The Trajectory Dynamics of the Quadrotor UAV. In this research, a quadrotor UAV, which can be commonly seen in low-altitude airspace, is selected as the research object.

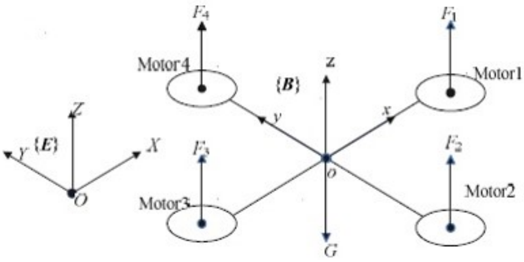

Figure 1. The structure model of the quadrotor. Two basic coordinate systems are established: the inertial coordinate system E(OXYZ) and aircraft coordinate system B(oxyz). The heading angle ψ denotes the angle between the projection of ox on the OXY plane and the X-axis. The pitch angle θ denotes the angle between the projection of oz on the OXZ plane and the Z-axis. The roll angle ϕ denotes the angle between the projection of oy on the OYZ plane and the Y-axis.

In the case of no wind and slow flight with zero drag coefficient and no additional disturbances, the simplified linear trajectory dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$ of the UAV is as following:[30]

$$
\begin{aligned}
\ddot{x} &= \theta g \\
\ddot{y} &= -\phi g \\
\ddot{z} &= U_1/m - g \\
\ddot{\phi} &= lU_2/I_x \\
\ddot{\theta} &= lU_3/I_y \\
\ddot{\psi} &= U_4/I_z
\end{aligned} \tag{1}
$$

in which θ, ϕ and ψ are Euler angles (defined in Figure 1), and the moment of inertia of the quadrotor is $\mathbf{I} = diag\{I_x, I_y, I_z\}$. U_i $(i = 1,2,3,4)$ are the control inputs of the UAV[37,38]. In detail,

$$
\mathbf{u} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} F_1 + F_2 + F_3 + F_4 \\ F_4 - F_2 \\ F_3 - F_1 \\ F_2 + F_4 - F_3 - F_1 \end{bmatrix} \tag{2}
$$

in which U_1 is vertical control input, U_2 is the rolling control input, U_3 is the pitch control input and U_4 is the heading control input. F_i $(i = 1,2,3,4)$ are the thrusts generated by the four motors as shown in Figure 1.

In this model, the complex nonlinear model of the quadrotor can be decomposed into four independent control channels. The whole model can be regarded as two subsystems: translation motion and angular motion. The two subsystems will not influence each other.

Objective Function. A typical objective function (also called cost function) for an optimal control problem is in the form of

$$
J = \Phi\left(\mathbf{x}(t_f), t_f\right) + \int_{t_0}^{t_f} g\left(\mathbf{x}(t), \mathbf{u}(t), t\right) dt \tag{3}
$$

in which g is related to the trajectory and control input, and Φ is related to the UAV states at the terminal time t_f.

For the problem of collision avoidance, there are many different objective functions that can be used. Using different objective functions often means getting a different optimal trajectory. In this paper, the objective function is chosen as following:

$$
\begin{aligned}
J &= \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt \\
&= \frac{1}{2}\int_{t_0}^{t_f} \left[\left(\mathbf{x} - \mathbf{x}_p\right)^T \mathbf{Q} \left(\mathbf{x} - \mathbf{x}_p\right) + \left(\mathbf{u}^T \mathbf{R} \mathbf{u}\right) \right] dt
\end{aligned} \tag{4}
$$

in which the first integral term penalizes the trajectory deviations from the planned route $\mathbf{x}_p(t)$ ($\mathbf{Q}$ is a positive-definite weighting matrix), and the second integral term is used to minimize the control input (also the energy cost of the system, $\mathbf{R}$ is another positive-definite weighting matrix).

Collision Avoidance Constraints. The own UAV must not collide with the target UAV during the transition from the initial state to the final state. Therefore, when the own UAV flies in the airspace, it should keep a minimum safety interval from other UAVs all the time. This constraint can be expressed by the following equation:

$$
r^2 - \left[(x - \tilde{x})^2 + (y - \tilde{y})^2 + (z - \tilde{z})^2 \right] \leq 0 \tag{5}
$$

in which r is the minimum safety interval. It is obtained by considering the physical conflicts between UAVs, the influence of the airflow, and the possible measurement errors in the actual physical scene. The quantities $(x(t), y(t), z(t))$ and $(\tilde{x}(t), \tilde{y}(t), \tilde{z}(t))$ are the positions of the own UAV and the object UAV, respectively, and both vary with time.

Other Constraints. The initial and final states of the own UAV are settled in this research, which introduces equality constraints to the problem.

Furthermore, to satisfy the physical constraints of the UAVs, it is necessary to consider the saturation of the control input. The control input of the UAV should satisfy

$$
U_{i,min} \leq U_i \leq U_{i,max} \tag{6}
$$

in which U_i $(i = 1,2,3,4)$ are defined by Equation (2), and $U_{i,min}$ and $U_{i,max}$ are the maximum and minimum of the control input U_i, respectively.

Problem Discretization: Pseudospectral Method

To solve the optimal control problem achieved above, direct numerical methods are more preferred because they do not require the derivation of the optimality conditions and they are more tolerant to initialization with low quality. The pseudospectral methods are introduced in this research because it can provide high accurate approximations for problems whose solutions are smooth. With pseudospectral methods, the optimal control problem can be transformed to a nonlinear programming problem by parameterizing the state and control input using global polynomials and collocating the differential-algebraic equations using nodes obtained from a Gaussian quadrature[31,39].

As stated above, the initial and final states of the controlled UAV are determined and do not need to be optimized. Therefore, it is reasonable in this paper to choose Legendre-Gauss (LG) points rather than Legendre-Gauss-Lobatto (LGL) points or Legendre -Gauss-Radau (LGR) points as the collocation points[40].

Legendre-Gauss points τ are the roots of the Legendre function[40] $P_N(\tau)$:

$$
P_N(\tau) = \frac{1}{2^N N!} \frac{\mathrm{d}^N}{\mathrm{d}\tau^N} (\tau^2 - 1)^N \tag{7}
$$

in which N is the number of the orthogonal collocation points.

By applying the pseudospectral method, the dynamic equations (1) can be written in the form of

$$
\sum_{i=0}^{N} \mathbf{D}_{ki} \mathbf{x}_i = \frac{t_f - t_0}{2} f(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \tag{8}
$$

in which

$$
\mathbf{D}_{ki} = \begin{cases} \frac{a_k}{a_i(\tau_k - \tau_i)} & \text{if } k \neq i \\ \sum_{j=0, j\neq i}^{N} \frac{1}{\tau_i - \tau_j} & \text{if } k = i \end{cases} \tag{9}
$$

where

$$
a_i = \prod_{\substack{j=0 \\ j \neq i}}^{N} (\tau_i - \tau_j) \tag{10}
$$

The cost function can also be transformed by the Gauss methods. Equation 4 can be written in the form of

$$\begin{aligned} J &= \frac{t_f - t_0}{2} \int_{-1}^{1} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \\ &= \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k g(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \end{aligned} \tag{11}$$

where w_k are the weights for Gauss-Legendre quadrature:

$$w_k = \frac{2}{(1-\tau_k^2)[P_N'(\tau_k)]^2} \tag{12}$$

in which $P_N(x)$ is defined by Equation (7).

Therefore, the original optimal control problem is converted into a nonlinear programming problem with the control and state variables to be optimized at the collocation points.

Existence of the Global Optimal Solutions

According to the convex optimization theory, for a convex programming problem, if x* is a local optimal solution of the problem, then it is the only global optimal solution of the problem[41].

For the collision-avoidance optimal control problem given by the previous section, it can be noted that the objective function (Equation (4)) is in a quadratic form. The trajectory dynamics (Equation (1)), together with the input saturation and boundary conditions constitute the equality constraints, and all of them are linear. However, the collision avoidance constraints of the problem are in the form of Equation (5), and it is a non-convex form.

Fortunately, if the collision avoidance maneuver is kept in one side of the half-plane containing the original trajectory, Equation (5) will be simplified to a convex form. For example, if the own UAV in level flight avoids the potential collisions only by reducing the flight height near the object UAV (i.e. let $z < z_0$, $\dot{x} = const$ and $\dot{y}$=0, the UAV moves only below the horizontal plane containing the original trajectory), the safety constraint Equation (5) can be simplified to $z < -\sqrt{r^2 - x^2 - y^2}$ (r, x and y are priori) which is a convex form. Such additional restrictions are acceptable for quadrotor UAVs.

Therefore, by adding such an acceptable restriction, all the objective function and the constraints are in the convex form, which makes the collision avoidance problem a convex programming problem. If we can find a local optimal solution x^* for the above-mentioned optimal control problem through the method such as sequential quadratic programming (SCP), then it is the only global optimal solution.

Result Correction

Through numerical calculation, the solution of the optimal control problem, including the collision-free optimal trajectory $\mathbf{x}^*$ and its corresponding optimal control input $\mathbf{u}^*$ can be achieved. However, if the result of the numerical solution is directly used as the control input of the UAV, the UAV will deviate from the calculated optimal collision-free trajectory (An example will be shown in the Simulation part.) Therefore, a closed-loop feedback controller LQR is introduced here to modify the UAV's control input so that the UAV can fly according to the calculated optimal trajectory.

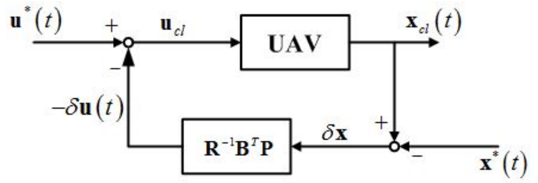

Figure 2. Brief structure of the feedback controller. The LQR controller is chosen here because it can make the UAV precisely follow the obtained optimal collision-free trajectories. Its stability and reliability have been verified in its wide range of applications. $\mathbf{v}$ is the possible measurement error and $\mathbf{y}$ is the output of the system.

According to the LQR theory, the closed-loop control input should be

$$\mathbf{u}_{cl} = \mathbf{u}^* - \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)\left[\mathbf{x}(t) - \mathbf{x}^*(t)\right] \tag{13}$$

in which $\mathbf{P}(t)$ is the solution of the Riccati equation,

$$\begin{aligned} &-\dot{\mathbf{P}}(t) = \mathbf{P}(t)\mathbf{A} + \mathbf{A}^T\mathbf{P}(t) - \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t) + \mathbf{Q} \\ &\mathbf{P}(t_f) = \mathbf{S} \end{aligned} \tag{14}$$

where $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{S}$ are positive-definite weighting matrices.

Hence, the regular algorithm of collision avoidance is as following.

Algorithm I for Collision Avoidance

```
1. Obtain the structural parameters of the UAV ($m$, $I$, $l$, etc.), the planned route $\mathbf{x}_p$ of the own UAV and the route $\tilde{\mathbf{x}}$ of the object UAV.
2. Determine the number of collocation points $N$, the minimum safety distance $r$ and the saturation of the input.
3. Establish the Optimal Control Problem (OCP) with the cost function (4) subject to (1),(5) and (6).
4. Transform the OCP into a nonlinear programming problem (NLP) using pseudospectral method.
5. Find solution (the collision-free optimal trajectory $\mathbf{x}^*$ and its corresponding optimal control input $\mathbf{u}^*$) to the NLP by SCP.
6. Modify the control input $\mathbf{u}_{cl}$ by the LQR controller.
```

Simplified Description of the Collision Avoidance Scenarios

Solving the optimal control problem using numerical iteration require a large amount of calculation, so it is often used in offline calculations. This research attempts to explore how the results of these offline calculations can be applied to the UAV real-time online collision avoidance process.

Consider a scenario between two UAVs. The target UAV maintains its motion state unchanged while the own UAV needs to change its height (or move sideways) to avoid potential collisions with the target UAV. According to the description in the previous section, once the initial position and velocity of both the own UAV and the target UAV are determined, the optimal collision-free trajectory of the own UAV can be uniquely determined by the optimal control method. This conclusion can be expressed by the following mapping:

$$\mathcal{F}: \left(\mathbf{x}_0, \dot{\mathbf{x}}_0, \tilde{\mathbf{x}}_0, \dot{\tilde{\mathbf{x}}}_0\right) \xrightarrow{\text{optimal control problem}} \mathbf{x}(t) \tag{15}$$

in which $\mathbf{x}(t) = (x(t), y(t), z(t))$. This relationship is similar to function mapping. It inspires us to think, if the optimal collision-free trajectories for scene A and scene B have been obtained by solving the corresponding optimal control problem, can we use these trajectories to directly achieve the collision-free trajectory for scene C without solving the specific complex optimal control problem? If so, we can calculate and store the optimal collision-free trajectories for some scenarios offline, and use these trajectories to solve the real-time collision avoidance problems for the own UAV online.

However, as shown in Equation (15), 12 variables (the position and velocity of both the own UAV and the object UAV in 3-dimensional directions) are needed to describe a specific collision avoidance scenario and to affect the final collision-free trajectory, which complicates the problem. Referring to the idea in ”sense-and-avoid” algorithms, a relative coordinate is applied here to simplify the description of the collision avoidance scenarios.

Relative Coordinate and Normalization

Considering the collision avoidance scenario between the own UAV and a single object UAV. The relative coordinate system is introduced as shown in Figure 3. It significantly reduces the number of variables required to describe the position and velocity of the two UAVs.

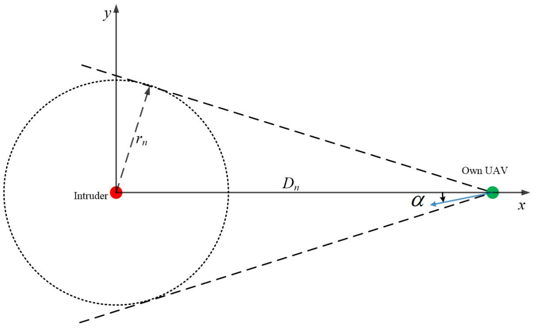

Figure 3. Plan view of the relative coordinate. The target UAV is set at the origin of the coordinate (the red dot) and is kept still, while the own UAV is set on the *x*-axis of the coordinate (the green dot) and is moving to the target UAV with the velocity of $\mathbf{v}_r$ (the blue vector). The angle between the direction of the relative velocity and the negative *x*-axis is denoted as α. r_n represents the minimum safety distance interval and D_n represents the initial distance between the two UAVs.

It is a common sense that the larger the relative velocity $\mathbf{v}_r$ is, the larger the minimum safety distance interval between the two UAVs should be. Therefore, normalization is used here to deal with this issue. Set the norm of relative velocity to be 1, and the initial distance between the own UAV and the object UAV will be D_n and the minimum safety distance interval between the two UAVs (the radius of the protect circle) will be r_n accordingly.

Hence the 12 variables in Equation (15) will be reduced to only 2 variables, namely, the initial distance D_n and the angle α.

$$\mathcal{F}: (D_n, \alpha) \xrightarrow{\text{optimal control problem}} \mathbf{x}(t) \tag{16}$$

Therefore, in the collision avoidance scenario of two UAVs, it is the variable D_n and α that determine the final collision-free trajectory. Several discussions about the effect of them will be made based on the simulation results.

Variable Domain

The relative geometric relationship in Figure 3 shows that the two UAVs will have potential collision risks only when the relative velocity is in the region enclosed by the two dashed lines, which means

$$\alpha \in [-\arcsin(r_n/D_n), \arcsin(r_n/D_n)] \tag{17}$$

If α lies in this range, the distance between the two UAVs will also be less than the minimum safety interval, which is very dangerous. Specifically, when $\alpha = 0°$, the relative velocity points directly at the object UAV and the two UAVs will have a direct mid-air collision. In addition, if the influence of other obstacles in the environment cannot be considered, only the upper part $[0, \arcsin(r_n/D_n)]$ needs to be considered according to the symmetry of the geometric space.

For the variable D_n, it should be at least greater than the given minimum safe interval r_n, i.e. $D_n \geq r_n$. More detailed range of D_n will be discussed later in the paper.

Symmetry of the Optimal Trajectory

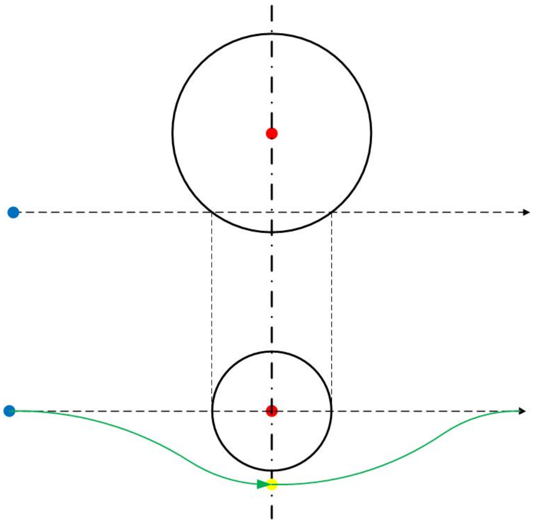

Figure 4. Illustration of the trajectory symmetry. The upper part is the top-view of the collision avoidance situation in the relative coordinate, while the lower part is the side view of it. The red mass points represent the object UAV and the circles around them show the minimum interval between the two UAVs. The dashed lines are the planned straight route of the own UAV.

In this research, the collision avoidance maneuver is strictly limited in a two-dimensional plane. For example, if the potential collision is avoided only by changing height, the resulting optimal collision-free trajectory is a two-dimensional curve (shown by the green curve in Figure 4) and it is symmetric about the vertical plane that contains the object UAV. Note that the initial and the final points of the resulting trajectory are on the planned route and in the equilibrium state of the own UAV, so that the trajectory can be smoothly connected with the original planned route. The theorem can be simply proved by the counter-evidence method.

Theorem 1. *Given the above conditions, the resulting optimal collision-free trajectory is symmetric about a plane that contains the object UAV and is perpendicular to the planned route.*

Proof. Since the research object of this paper is a quadrotor, which has strong symmetry, any flyable trajectory is reversible. In other words, if the UAV can fly from point A to point B with a certain procedure, then the UAV must be able to fly back to point A with the reverse procedure from point B. In the scenario involved in this paper, the UAV has the same flight state at the initial point and the ending point of collision avoidance and they are both on the planned route. Therefore, if the UAV can fly from the initial point to the ending point with a certain procedure, the UAV can be able to fly from the initial point to the ending point with the corresponding mirroring procedure. The two resulting trajectories have the same distance to the planned route. Therefore, the trajectory is optimal only when it is symmetrical about the plane. (Proof end.)

Corollary 1. *Since the trajectory of the UAV has to be continuous, the slope at the intersection point of the trajectory and the plane (the yellow point in Figure 4) has to be zero.*

The symmetry of the trajectory can bring many benefits.

First of all, since the initial distance between the two UAVs D_n is determined, the trajectory planning problem becomes much simpler in the finite time domain than in the infinite time domain by applying the trajectory symmetry.

Besides, when solving the problem numerically, the symmetry makes it possible to use less discrete points to get a better result, especially when polynomial discrete points are applied. For polynomial discrete points, there are more discrete points at the ends of the time domain than in the middle. Therefore, more discrete points are located in the region of interest near the object UAV due to the symmetry.

Hence, the algorithm of collision avoidance can be improved as Algorithm II.

Simulation and Discussion

In this section, the effectiveness of the proposed algorithms is simulated and verified by MATLAB programs in several scenarios. The first simulation illustrates a scenario in which the own UAV has to avoid a potential mid-air collision with an object UAV by solving the optimal control problem. The necessity of using a feedback controller is shown clearly in this example. In simulation II, the collision avoidance problem is set in the normalized relative coordinate systems. The own UAV has to avoid the object UAV by querying the offline calculation results. The effects of the initial distance D_n and different direction angle α are also discussed in this part. Finally, in simulation III, the proposed method is generalized to solve the problem of collision avoidance for multiple UAVs in a simple scenario.

Algorithm II for Collision Avoidance with Relative Coordinate and Normalization

```
1. Obtain the structural parameters of the UAV ($m$, $I$, $l$, etc.), the planned route $\mathbf{x}_p$ of the own UAV and the route $\tilde{\mathbf{x}}$ of the object UAV.
2. Calculate the initial distance $D_n$ and the angle $\alpha$.
3. Determine the number of collocation points $N$, the minimum safety distance $r_n$ and the saturation of the input.
4. Establish the Optimal Control Problem (OCP) with the cost function (4) subject to (1),(5) and (6) in the normalized relative coordinate.
5. Transform the OCP into a nonlinear programming problem (NLP) using pseudospectral method.
6. Find solution to the NLP by SCP.
7. Modify the control input $\mathbf{u}_{cl}$ by the LQR controller.
8. Convert the trajectory of the normalized relative coordinate to the actual coordinate.
```

In all the above simulation scenarios, the proposed algorithm can successfully give the collision-free trajectories of the UAVs. These trajectories can be precisely followed by the UAVs in actual environment since the algorithm has already taken the UAV dynamics into considerations and the additional closed-loop feedback controller has been introduced (as stated in the subsection ”Result Correction”). Therefore, the simulation results here can basically reflect the effectiveness of the proposed algorithm. The test under the real environment will be carried out in our future research.

Simulation I: A Classical Collision Avoidance Scenario

A classical scenario of collision avoidance is illustrated in Figure 5 and some of the parameters of the UAVs used in the simulation are listed in Table 1. It is easy to find out in Figure 5 that there will be a potential mid-air collision (the minimum distance between the two UAVs will be smaller than r) if no avoiding maneuver is conducted. In our simulation, the collision avoidance maneuvers are conducted by only the own UAV, while the object UAV just maintains its original trajectory. The collision-free trajectory is achieved by solving the corresponding optimal control problem.

Table 1. The System Parameters in Simulation I

UAV Parameters	Value
Mass (m)	1.2 kg
Gravity coefficient (g)	9.8 m/s^2
Length (l)	0.2 m
Moment of Inertia (I_x)	2.353×10^{-3} kg $\cdot$ m^2
Minimum Safety Distance (r)	25 m
Saturation of Input (U_2)	$-0.1mg \leq U_2 \leq 0.1mg$
No. of the Collocation Points (N)	41

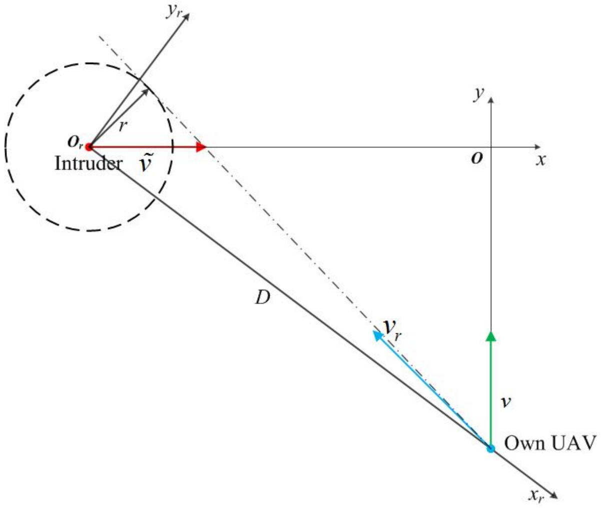

Figure 5. The plan view of the initial state in Simulation I. The object UAV is located at $(\tilde{x}_0, \tilde{y}_0, \tilde{z}_0) = (-120 \text{ m}, 0, 30 \text{ m})$, and the velocity of it is $(\dot{\tilde{x}}, \dot{\tilde{y}}, \dot{\tilde{z}}) = (3.536 \text{ m/s}, 0, 0)$. The own UAV is located at $(x_0, y_0, z_0) = (0, -90 \text{ m}, 30 \text{ m})$ with the initial velocity of $(\dot{x}, \dot{y}, \dot{z}) = (0, 3.536 \text{ m/s}, 0)$.

As described in the previous sections, the collision avoidance problem can be formed into an optimal control problem as following:

$$
\begin{aligned}
\min \quad & J = \frac{1}{2}\int_0^{60} \left[(10)x^2(t) + U_2^2(t)\right] dt \\
\text{s.t.} \quad & \ddot{x} = -\phi g \\
& \ddot{\phi} = lU_2/I_x \\
& \mathbf{x}(t) - \tilde{\mathbf{x}}(t)\|_2 \geq 25 \text{ m} \\
& x(0) = x(60) = 30 \text{ m} \\
& -0.1mg \leq U_2 \leq 0.1mg
\end{aligned}
\tag{18}
$$

Note that only part of the equations in Equations (1) are valid here because the own UAV is set to avoid the potential collisions by moving sideways and the height and forward speed of the UAV remain unchanged.

By solving the optimal control problem, the collision avoidance process in this scenario is shown in Figure 6. The own UAV avoids the mid-air collision successfully.

To verify if the achieved optimal collision-free trajectory meets all the requirements, some further result is shown in Figure 7. In the first graph of Figure 7, the blue curve shows the movements of the own UAV in x-direction. It is the spline interpolation of the optimal trajectory positions at the orthogonal collocation points. The red curve is obtained by applying the optimal input achieved by solving the optimal control problem to the system directly in the open-loop (simulated by applying the fourth-order Runge-Kutta method). Obviously, it is unstable and deviates from the optimal trajectory (the blue curve) after avoiding the object UAV. Therefore, it is necessary to introduce the LQR controller described in the subsection ”Result Correction”. The closed-loop result is shown by green curve in Figure 7 and it tracks the blue curve much better than the red one. The second graph in Figure 7 shows that the distance between the two UAVs is larger than the minimum safety distance set at all the collocation points.

In conclusion, this simulation shows that it is feasible to solve the collision avoidance problem by using the optimal control method. A nice optimal collision-free trajectory for the own UAV can be achieved through the proposed procedure.

It should also be pointed out that the necessity of introducing the LQR controller has been shown in this simulation. The control input got from the numerical calculation should be further revised by a closed-loop control system rather than applied to the UAV directly. This may due to the inherent drawbacks (the accumulation of truncation errors in computer simulation) of the pseudospectral method.

Simulation II: Using the Relative Coordinate and Normalization

In this part, the relative coordinate and normalization described in the third section are applied to simplified the description of the collision avoidance scenarios. The own UAV is set to avoid the potential collision avoidance by changing the flight height in this part of the simulation.

Some of the parameters of the UAVs used in the simulation are listed in Table 2.

Table 2. The System Parameters in Simulation II

UAV Parameters	Value
Mass (m)	1.2 kg
Gravity coefficient (g)	9.8 m/s^2
Minimum Safety Distance (r_n)	5
Saturation of Input (U_1)	$-mg \leq U_2 \leq 0.5mg$
N0. of the Collocation Points (N)	25

First, the effect of different initial distance D_n and the direction angle α to the resulting collision-free trajectory is discussed here.

The Initial Distance D_n. In the normalized relative coordinate, ten different initial distances $D_n = 15, 14, 13, 12, 10, 9, 8, 7, 6, 5.5$ are selected with respect to the fixed minimum safety interval $r_n = 5$, while α is kept to be zero for simplicity. The resulting collision-free trajectories are shown in Figure 8.

In Figure 8, all of the trajectories remain coincident with the planned route near the beginning and end of the collision avoidance (maintaining horizontal linear motion) and move downwards to avoid the conflicts as they approach the object UAV. Besides, the smaller the initial distance is, the larger the overshoot of the trajectory will be.

Moreover, it is clearly shown in Figure 9 that, from $D_n = 15$ to $D_n = 5.5$, the smaller the initial distance D_N is, the larger the change of the control input U_1 will be. When the initial distance is very close to the minimum safety interval (for example, $D_n = 5.5$ in Figure 9), the maximum of the corresponding control input is approaching its saturation value, which does not meet our expectation for optimizing the control input.

Therefore, in a realistic collision avoidance scenario, it is reasonable to set the value of the initial distance D_n to about 1.5 to 3 times the minimum safety distance r_n. When the

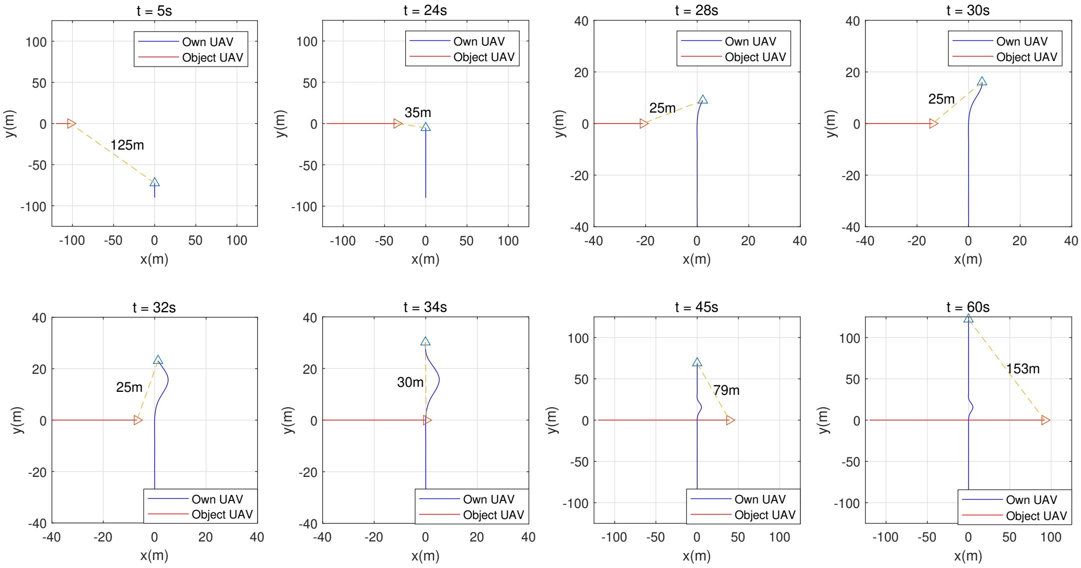

Figure 6. The collision avoidance process for Simulation I. The location and the trajectories of both the own UAV and the object UAV are shown in this figure at 8 different time points (t = 5s, 24s, 28s, 30s, 32s, 34s, 45s and 60s). The distance between the two UAVs is indicated by a dashed line and the value is marked.

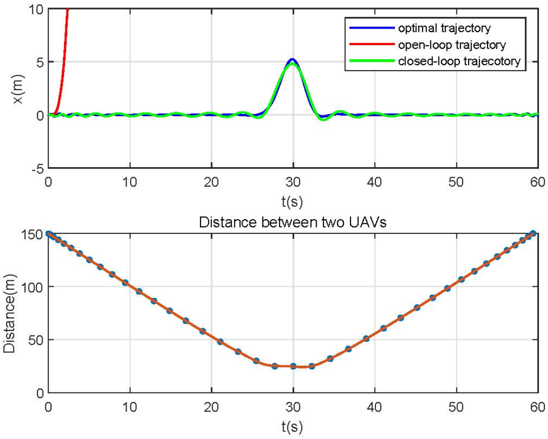

Figure 7. Verification of the result in Simulation I. The upper part shows the movements of the own UAV in x-direction. It is clear that the closed-loop trajectory follow the optimal collision-free trajectory much better than the open-loop trajectory. Therefore, it is necessary to introduce the feedback controller. The second graph shows that the minimum distance between the two UAVs is 25 m, which meets the requirement.

distance between the two UAVs is greater than that value, the own UAV can maintain its original flight state. The own UAV starts the collision avoidance operation according to the planned trajectory until D_n reaches the set value.

The Direction Angle α. In the normalized relative coordinate, ten uniformly distributed angles were selected within the domain with respect to the fixed minimum safety interval $r_n = 5$. (D_n is set to three times the r_n.) The resulting collision-free trajectories are shown in Figure 10.

In Figure 10, when the angle α increases from 0 to $\arcsin r_n/D_n$, the deviation of the resulting collision-free trajectory to the planned route becomes smaller. The essential reason for this phenomenon is that as the angle α increases, the distance between the object UAV and the closest point of approach of the own UAV becomes larger. When $\alpha = \arcsin r_n/D_n$, the planned route is tangent to the protected area, and the resulting trajectory is a straight line coinciding with the planned route.

Furthermore, it is worth noting that if more different values of angle α are selected, more collision-free optimal trajectories will be achieved. These trajectory curves can form a surface in a three-dimension area (as shown in Figure 11).

It can be shown that any point on this surface satisfies the safety distance constraint. Therefore, with the surface in Figure 11, the collision-free trajectories correspond to any other different angle α can be directly achieved by interpolation on the surface. Although the trajectory achieved by interpolation may not be optimal, it is acceptable. And the more optimal collision-free trajectories used to obtain the surface, the smaller the deviation between the collision-free trajectory obtained by interpolation and the optimal non-collision trajectory for certain angle α.

The Collision-free Trajectory Generated by Using Offline Results. Consider the collision avoidance scenario in ”Simulation I” again. This time, we redefine this collision avoidance scene in the normalized relative coordinate system, and try to obtain the collision-free trajectory by querying (interpolating) the offline collision-free trajectory that has been calculated in the previous two steps.

It can be easily got from Figure 5 that the relative speed between the two UAVs is $\|v_r\| = 5$ m/s. In the normalized relative coordinate, the angle between the relative velocity

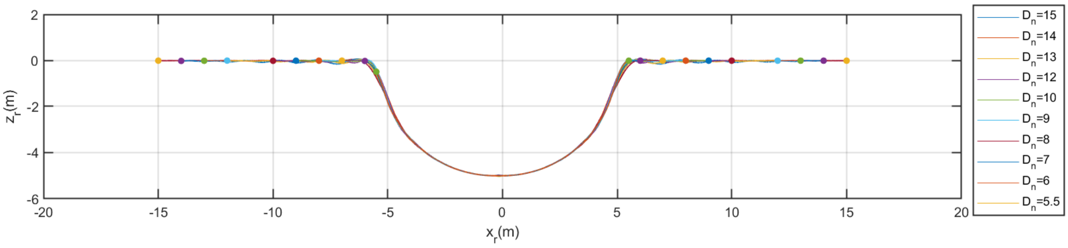

Figure 8. The collision-free trajectories with different initial distances ($D_n = 15, 14, 13, 12, 10, 9, 8, 7, 6, 5.5$) in the normalized relative coordinate.

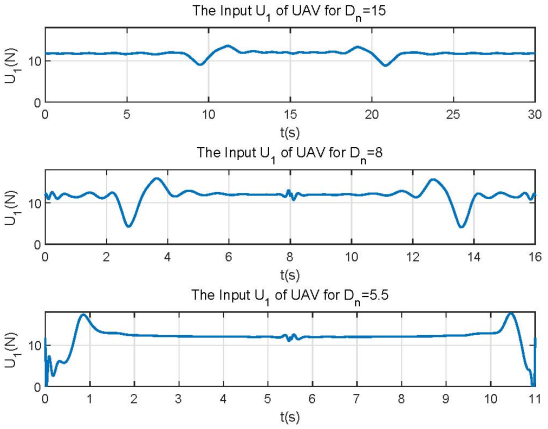

Figure 9. The corresponding control inputs to the collision-free trajectories with different initial distances ($D_n = 15, 8, 5.5$). It is clear that control input is approaching its saturation when the initial distance $D_n = 5.5$.

and the negative x-axis is $\alpha = 8.13^o$. The minimum safety interval is $r_n = 5$ and the initial distance $D_n = 30$.

Therefore, we can use the collision-free trajectories of $D_n = 3, \alpha = 6.49^o$ and $D_n = 3, \alpha = 8.65^o$ shown in Figure 10 to generate the collision-free trajectory of $D_n = 3$, $\alpha = 8.13^o$, and then extend both ends of this trajectory with a horizontal straight line until the distance from its initial point and end point to the object UAV satisfies $D_n = 30$. The resulting trajectory is shown in Figure 12 and it is collision-free with the object UAV.

Simulation III: Collision Avoidance with Multiple UAVs

In this simulation, a collision avoidance problem among multiple UAVs is solved by the proposed method. The scenario is illustrated in Figure 13. Four identical UAVs are involved in this scenario and their initial and terminal positions are given in Table 3. All the 4 UAVs have a speed of 5 m/s and point directly to their terminal position. They need to avoid the potential collisions with each other by changing the flight height.

To be noticed, we do not want to solve the optimal control problem directly online to get the collision-free trajectories in this simulation. Instead, some collision-free trajectories for certain "D_n - α" pairs have already calculated offline. As shown in Figure 14, D_n of the pre-planned trajectories is chosen to be 10, which is suitable according to the conclusion achieved in the former simulation. Three different angles $\alpha = 5^o, 10^0, 15^o$ are uniformly selected, which can roughly cover the entire feasible domain. The collision-free trajectories for the UAVs in the scene are then achieved by interpolating these pre-planned trajectories. This setting is made to illustrate that the pre-planned trajectory calculated offline strategy can provide the UAVs with collision-free trajectories during the real-time flight of the UAVs, thereby avoiding the case that the corresponding optimal control problem cannot be solved online.

Table 3. The original and terminal position of the UAVs in Simulation III

UAV	The original position	The terminal position
UAV1	(0,150m)	(150m,0)
UAV2	(0,100m)	(150m,150m)
UAV3	(0,50m)	(150m,50m)
UAV4	(0,0)	(150m,100m)

*All the 4 UAVs are in the same height at the initial and terminal time.

Besides, priorities are usually set in the multi-UAVs collision avoidance scenarios. In this simulation, UAV1 has the highest priority, followed by UAV2 and UAV3, and UAV4 has the lowest priority. Set the minimum safety interval to be $r_n = 5$.

In Figure 13, it seems that there will be 4 potential mid-air collisions to be avoided (namely, the collision between UAV1 and UAV2, UAV1 and UAV3, UAV1 and UAV4 and UAV3 and UAV4). There are no collision risks between UAV2 and UAV3 and between UAV2 and UAV4. However, by applying the proposed algorithm, the relative velocity v_r together with the initial distance D_n and direction angle α can be quickly calculated. They are listed in Table 4. According to the variable domain stated in the second section, only the UAV pairs "UAV1 & UAV2" and "UAV1 & UAV4" are going to have potential collisions. Since UAV 1 has the highest priority, only the collision-free trajectories of UAV2 and UAV4 need to be re-planned. UAV1 and UAV3 can just fly as planned.

Therefore, the collision-free trajectory of UAV2 can be achieved by interpolating the trajectories of "$D_n = 10$ - $\alpha = 10°$" and "$D_n = 10$ - $\alpha = 15°$", while the collision-free trajectory of UAV4 can be achieved by interpolating the trajectories of "$D_n = 10$ - $\alpha = 5°$" and "$D_n = 10$ - $\alpha = 10°$". The corresponding collision-free trajectories for the four UAVs are shown in Figure 15.

Figure 16 is used to check if the distances between each two UAVs really satisfy the minimum safety interval. It

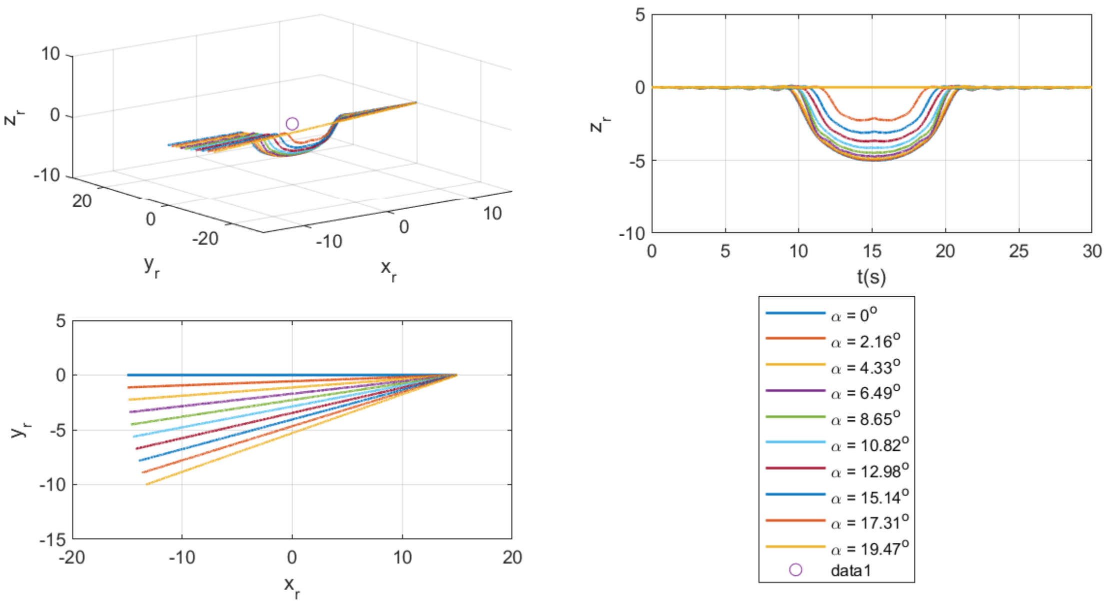

Figure 10. The collision-free trajectories of different angle α. The initial position of the own UAV is located at (15, 0, 0), and 10 different α are chosen (as shown in the lower left part of the figure.). The upper-left part of the figure shows the collision-free trajectories in 3D space. The change of the vertical height of the own UAV over time is shown in the upper right part of the figure.

Table 4. Analysis of the scenario in Simulation III

UAV pairs	v_r (m/s)	D_n	α	$\arcsin(r_n/D_n)$	Notes
UAV1 & UAV2	4.21	11.89	13.28°	24.87°	potential collision
UAV1 & UAV3	3.06	32.66	22.50°	8.81°	no collision
UAV1 & UAV4	5.07	9.86	5.65°	9.73°	potential collision
UAV2 & UAV3			$> 90°$		no collision
UAV2 & UAV4	1.06	94.17	$-26.06°$	3.04°	no collision
UAV3 & UAV4	2.32	21.57	$-16.85°$	13.40°	no collision

$^*\alpha$ should satisfy $|\alpha| < \arcsin(r_n/D_n)$ to have a potential collision.

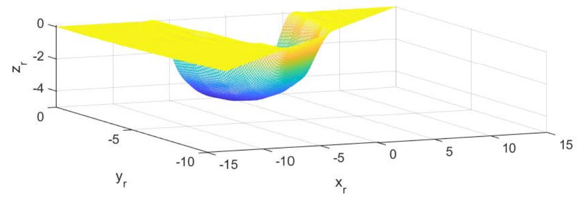

Figure 11. The surface formed by the collision-free trajectories of different angle α. It is clear that all the points on this surface satisfy the minimum distance interval with the object UAV.

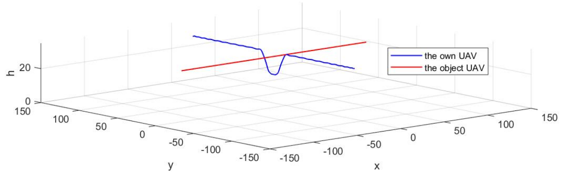

Figure 12. The collision-free trajectory for the own UAV in the scenario in Simulation I. The own UAV avoids the potential collision by changing its flight height. The trajectory is achieved by querying the offline result in Figure 10 rather than solving the optimal control problem again.

shows that the planned trajectories for each UAV satisfy the requirements.

Although the scenario of this problem is relatively simple, this simulation gives an example of applying the proposed method to solve a collision avoidance problem among multiple UAVs. It shows that for multiple UAV collision avoidance problem which can be decomposed into several 2 UAV collision avoidance problem, the proposed method can give a lot of convenience in both judging whether there will be potential collisions and giving the corresponding collision-free trajectory quickly online. For real application with more complex scenario, the proposed method needs some further improvements.

Conclusion

In this paper, optimal control method is applied to solve the problem of trajectory planning with mid-air collision avoidance for quadrotor UAVs. A feedback controller is applied to eliminate the accumulate error generated by the use of pseudospectral method. Furthermore, the introduction of the normalized relative coordinate system enables the

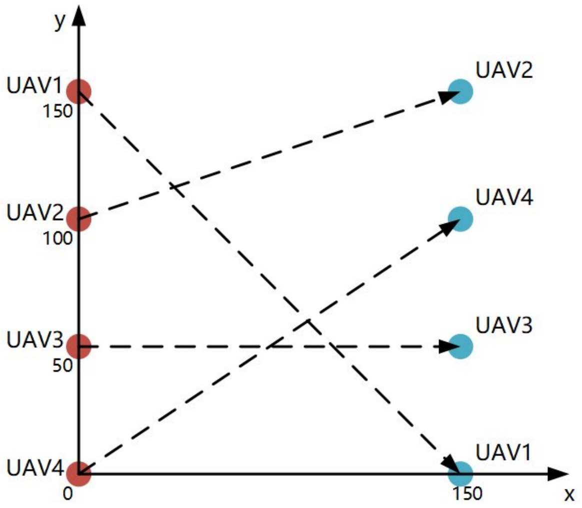

Figure 13. This is the plan view of the collision avoidance scenario in Simulation III. 4 identical UAVs are lined up on the y-axis and plan to fly straight to the target position on the line $x = 200$.

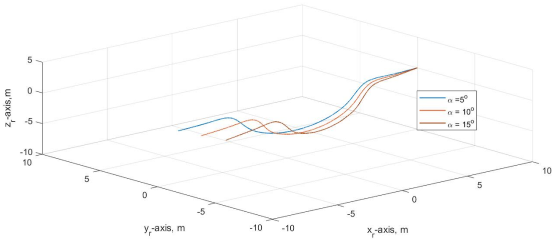

Figure 14. These are 3 offline pre-planned trajectories for Simulation III. D_n is set to be 10 and three different angles $\alpha = 5^o, 10^0, 15^o$ are chosen.

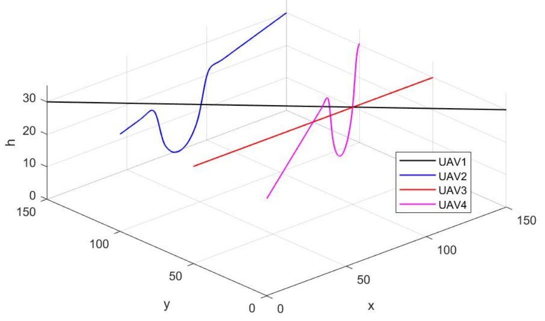

Figure 15. The collision-free trajectories of Simulation III. UAV2 and UAV4 changes their altitude to avoid the potential collisions with UAV1.

UAV to obtain collision-free trajectories more efficiently. The future work of this research includes the application of the proposed method to more complex collision avoidance scenario between multiple UAVs.

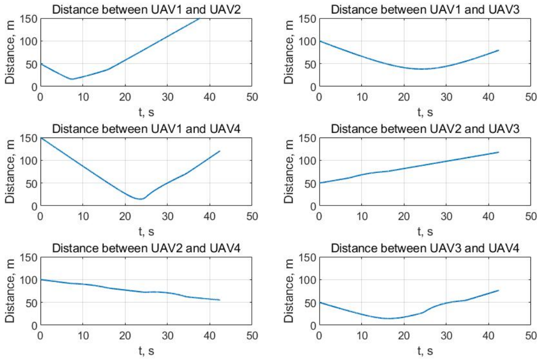

Figure 16. The distance between any two UAVs meets the minimum safety distance constraint.

References

1. Tahir A, Böling J, Haghbayan MH et al. Swarms of unmanned aerial vehicles—a survey. *Journal of Industrial Information Integration* 2019; 16: 100106.
2. Hoffmann GM and Tomlin CJ. Decentralized cooperative collision avoidance for acceleration constrained vehicles. *47th Ieee Conference on Decision and Control, 2008 (Cdc 2008)* 2008; : 4357–4363DOI:Doi10.1109/Cdc.2008.4739434. URL <GotoISI>://WOS:000307311604079.
3. Hafner MR, Cunningham D, Caminiti L et al. Cooperative collision avoidance at intersections: Algorithms and experiments. *Ieee Transactions on Intelligent Transportation Systems* 2013; 14(3): 1162–1175. DOI:10.1109/Tits.2013.2252901. URL <GotoISI>://WOS:000324336100013.
4. Behjat A, Paul S and Chowdhury S. Learning reciprocal actions for cooperative collision avoidance in quadrotor unmanned aerial vehicles. *Robotics and Autonomous Systems* 2019; 121. DOI:ARTN10327010.1016/j.robot.2019.103270. URL <GotoISI>://WOS:000491214500013.
5. Tsourdos A, White B and Shanmugavel M. *Cooperative path planning of unmanned aerial vehicles*. John Wiley & Sons, Ltd, 2010.
6. Mujumdar A and Padhi R. Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles. *Journal of Aerospace Computing Information and Communication* 2011; 8(2): 17–41. URL <GotoISI>://WOS:000287796700001.
7. Chand BN, Mahalakshmi P and Naidu VPS. Sense and avoid technology in unmanned aerial vehicles: A review. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (Iceeccot)*. pp. 512–517. URL <GotoISI>://WOS:000426976100095.
8. Huiying L, Cunru B, Guangjun Y et al. Application and analysis and discussion of autonomous collision avoidance techniques for unmanned aerial vehicle. *Advances in Aeronautical Science & Engineering* 2014; .
9. Lin CE and Wu YY. Tcas solution for low altitude flights. In *2010 Integrated Communications, Navigation, and Surveillance Conference*. pp. I4–1.
10. Prats X, Delgado L, Ramirez J et al. Requirements, issues, and challenges for sense and avoid in unmanned

aircraft systems. *Journal of Aircraft* 2012; 49(3): 677–687. DOI:10.2514/1.C031606. URL <GotoISI>://WOS:000305168000001.

11. Shin HS, Tsourdos A and White B. *UAS Conflict Detection and Resolution Using Differential Geometry Concepts*. John Wiley & Sons, Ltd, 2012.
12. Mueller ER and Kochenderfer M. Multi-rotor aircraft collision avoidance using partially observable markov decision processes. In *AIAA Modeling & Simulation Technologies Conference*. p. 3673.
13. Yu X, Zhou XB and Zhang YM. Collision-free trajectory generation and tracking for uavs using markov decision process in a cluttered environment. *Journal of Intelligent & Robotic Systems* 2019; 93(1-2): 17–32. URL <GotoISI>://WOS:000456791900003.
14. Chamseddine A, Zhang YM, Rabbath CA et al. Trajectory planning and replanning strategies applied to a quadrotor unmanned aerial vehicle. *Journal of Guidance Control and Dynamics* 2012; 35(5): 1667–1671. URL <GotoISI>://WOS:000314378900027.
15. Zhang N, Gai WD, Zhong MY et al. A fast finite-time convergent guidance observer for unmanned aerial vehicles law with nonlinear disturbance collision avoidance. *Aerospace Science and Technology* 2019; 86: 204–214. URL <GotoISI>://WOS:000463305200017.
16. Panyakeow P and Mesbahi M. Deconfliction algorithms for a pair of constant speed unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems* 2014; 50(1): 456–476. URL <GotoISI>://WOS:000336052700033.
17. Mac TT, Copot C, Hernandez A et al. Improved potential field method for unknown obstacle avoidance using uav in indoor environment. In *2016 Ieee 14th International Symposium on Applied Machine Intelligence and Informatics (Sami)*. pp. 345–350. URL <GotoISI>://WOS:000381795100059.
18. Smith NE, Cobb R, Pierce SJ et al. Optimal collision avoidance trajectories via direct orthogonal collocation for unmanned/remotely piloted aircraft sense and avoid operations. In *AIAA Guidance, Navigation, and Control Conference*. AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, p. 4619. DOI:doi:10.2514/6.2014-096610.2514/6.2014-0966. URL https://doi.org/10.2514/6.2014-0966.
19. Tang J, Fan LJ and Lao SY. Collision avoidance for multi-uav based on geometric optimization model in 3d airspace. *Arabian Journal for Science and Engineering* 2014; 39(11): 8409–8416. URL <GotoISI>://WOS:000344322900078.
20. Vilardaga S, Prats X, Duan P et al. Conflict-free trajectory optimization with target tracking and conformance monitoring. *Journal of Aircraft* 2018; 55(3): 1252–1260. URL <GotoISI>://WOS:000432802800027.
21. Kosari A and Teshnizi MM. Optimal trajectory design for conflict resolution and collision avoidance of flying robots using radau-pseudo spectral approach. In *2018 6th Rsi International Conference on Robotics and Mechatronics (Icrom 2018)*. ISBN 2377-679x, pp. 82–87. URL <GotoISI>://WOS:000465373900015.
22. Mellinger D, Kushleyev A and Kumar V. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*. ISBN 1050-4729, pp. 477–483. URL <GotoISI>://WOS:000309406700071.
23. Radmanesh M, Kumar M, Nemati A et al. Dynamic optimal uav trajectory planning in the national airspace system via mixed integer linear programming. *Proceedings of the Institution of Mechanical Engineers Part G-Journal of Aerospace Engineering* 2016; 230(9): 1668–1682. URL <GotoISI>://WOS:000380312300011.
24. Cai JL and Zhang N. Mixed integer nonlinear programming for aircraft conflict avoidance by applying velocity and altitude changes. *Arabian Journal for Science and Engineering* 2019; 44(10): 8893–8903. URL <GotoISI>://WOS:000486026700052.
25. Augugliaro F, Schoellig AP and D'Andrea R. 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)*. ISBN 2153-0858, pp. 1917–1922. URL <GotoISI>://WOS:000317042702073.
26. Yang J, Yin D, Shen LC et al. Cooperative deconflicting heading maneuvers applied to unmanned aerial vehicles in non-segregated airspace. *Journal of Intelligent & Robotic Systems* 2018; 92(1): 187–201. URL <GotoISI>://WOS:000443859600011.
27. Yu TY, Tang J, Bai L et al. Collision avoidance for cooperative uavs with rolling optimization algorithm based on predictive state space. *Applied Sciences-Basel* 2017; 7(4). URL <GotoISI>://WOS:000404447600019.
28. D'Amato E, Mattei M and Notaro I. Distributed reactive model predictive control for collision avoidance of unmanned aerial vehicles in civil airspace. *Journal of Intelligent & Robotic Systems* 2020; 97(1): 185–203. URL <GotoISI>://WOS:000512052700014.
29. Baca T, Hert D, Loianno G et al. Model predictive trajectory tracking and collision avoidance for reliable outdoor deployment of unmanned aerial vehicles. In *2018 IEEE/Rsj International Conference on Intelligent Robots and Systems (Iros)*. ISBN 2153-0858, pp. 6753–6760. URL <GotoISI>://WOS:000458872706029.
30. Jiang Y, Hu S and Damaren CJ. Collision avoidance algorithm between quadrotors using optimal control and pseudospectral method. In *AIAA Scitech 2019 Forum*. AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, p. 1415. DOI:doi:10.2514/6.2019-141510.2514/6.2019-1415. URL https://doi.org/10.2514/6.2019-1415.
31. Xu SB, Li SB and Cheng B. Theory and application of legendre pseudo-spectral method for solving optimal control problem. *Control & Decision* 2014; 29(12): 2113–2120.
32. Robinson DR, Mar RT, Estabridis K et al. An efficient algorithm for optimal trajectory generation for heterogeneous multi-agent systems in non-convex environments. *Ieee Robotics and Automation Letters* 2018; 3(2): 1215–1222. DOI: 10.1109/Lra.2018.2794582. URL <GotoISI>://WOS:000424646100048.
33. Vera S, Cobano JA, Heredia G et al. Collision avoidance for multiple uavs using rolling-horizon policy. *Journal of Intelligent & Robotic Systems* 2016; 84(1-4): 387–396. URL <GotoISI>://WOS:000390027900024.
34. Tahir A, Böling JM, Haghbayan MH et al. Comparison of linear and nonlinear methods for distributed control of a

hierarchical formation of uavs. *IEEE Access* 2020; 8: 95667–95680.

35. Tahir A, Böling J, Haghbayan MH et al. Development of a fault-tolerant control system for a swarm of drones. In *2020 International Symposium ELMAR*. IEEE, pp. 79–82.
36. Tahir A, Böling JM, Haghbayan MH et al. Navigation system for landing a swarm of autonomous drones on a movable surface. In *ECMS*. pp. 168–174.
37. Jun LI and Yuntang LI. Modeling and pid control for a quadrotor. *Journal of Liaoning Technical University* 2012; .
38. Mohanta JC, Parhi DR, Mohanty SR et al. A control scheme for navigation and obstacle avoidance of autonomous flying agent. *Arabian Journal for Science and Engineering* 2018; 43(3): 1395–1407. URL `<GotoISI>://WOS:000425983400030`.
39. Patterson MA and Rao AV. Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems. *Journal of Spacecraft and Rockets* 2012; 49(2): 364–377. URL `<GotoISI>://WOS:000302760100017`.
40. Garg D, Patterson M, Hager WW et al. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* 2010; 46(11): 1843–1851. URL `<GotoISI>://WOS:000284661000014`.
41. Boyd S and Vandenberghe L. *Convex Optimization*. Cambridge University Press, 2004.