Safe Trajectory Planning for Multiple Aerial Vehicles with Segmentation-Adaptive Pseudospectral Collocation

J. A. Cobano, S. Vera, G. Heredia and A. Ollero

Abstract—This paper proposes a method called Segmentation-adaptive Pseudospectral collocation to address the problem of safe trajectory generation in missions with cooperating multiple aerial vehicles. Pseudospectral collocation can generate optimized collision-free trajectories, but for multiple aerial vehicles it cannot guarantee that the safety separation distance is maintained in the whole trajectories, since the constraints are only enforced in discrete points in the trajectory (collocation points). Hp-adaptive pseudospectral collocation increases iteratively the number of collocation points and the degree of the approximating polynomial, but this may lead to an exponential increase of the computational load. The proposed method solves the problem by selectively adding new collocation points where they are needed, only in the segments with conflicts in each iteration, thus effectively reducing the number of collocation points and the computation time with respect to other pseudospectral collocation formulations. The proposed method allows both changes of speed and changes of heading for each aerial vehicle to guarantee the safety distance between them. Its computational load and scalability are studied in randomly generated scenarios. Moreover, a comparison with other method is presented. Several experiments to test the validity of the approach have been also carried out in the multivehicle aerial testbed of the Center for Advanced Aerospace Technologies.

I. INTRODUCTION

Efficient trajectory planning by using methods of optimization has been extensively studied in the literature. This is an important aspect in coordinated missions with multiple Unmanned Aerial Vehicles (UAVs). Therefore, the use of safe trajectory optimization techniques plays an important role in this project.

A complete review of the literature on trajectory planning algorithms and collision avoidance among aerial vehicles is presented in [1]. Some of the more frequently used are integer programming [2], graph search like A* [3], D* [4], Rapidly-exploring Random Trees (RRT) [5], RRT* [6], particle swarm optimization [7] and evolutionary computation methods [8].

Moreover, the methods also differ in the kind of allowed maneuvers in order to solve the detected collisions and the computational load. Some methods allow only one maneuver [8] [9] and others a combination of them [2]. On the other hand, not all the methods could be used in real-time applications, depending on the time of computation.

This paper addresses the problem of planning collision-free trajectories with multiple UAVs by ensuring the safety of the mission and considering the dynamics of the vehicles to compute more realistic trajectories. It is formulated as an optimal control problem including state and control inequality constraints. Differential equations should be discretized in order to solve the problem. The more commonly used methods are: finite difference method, finite element method and spectral method. Another method called pseudospectral method considers both finite element and spectral method in order to solve this problem. This method is based on the idea of relating collocation points to the structure of orthogonal polynomials because a proper choice of basis functions and the distribution of collocation points is crucial. The goal is to do a polynomial approximation to the state and control functions in terms of their values at Legendre-Gauss-Lobatto points. Each of these polynomials is defined by using Legendre-Gauss-Lobatto points as the collocation points and Lagrange polynomials as the basis functions of the corresponding interpolating polynomial. State and control inequality constraints are converted to a set of algebraic inequalities at the Legendre-Gauss-Lobatto nodes or collocation points. The integral of a cost function considered is discretized based on Legendre-Gauss-Lobatto quadrature rules. Finally, the algebraic nonlinear programming problem is solved by applying optimization tools [10] [11].

The proposed method is based on these pseudospectral techniques but some modifications have been implemented to improve the solution for multiple aerial vehicles and guarantee that the safety separation distance is maintained at all times. Two maneuvers are allowed to solve the detected collisions: change of speed and change of heading. The analytical method computes an optimal solution from a cost function and its main characteristic is the low computational load.

Some works have implemented collocation or pseudospectral methods reducing the number of dimensions of the problem [12] [13] [14] [15]. A pseudospectral method computes the solution considering the problem as a single segment, time interval, and the convergence is achieved by increasing the degree of the polynomial, that is the number of collocation points. The collocation points are chosen based on accurate quadrature rules and the basis functions are typically Chebyshev or Legendre polynomials. The more commonly used pseudospectral methods are the Gauss pseudospectral method (GPM) [16] and the Legendre pseudospectral method.
The hp-adaptive pseudospectral method proposed in [10] can increase the number of segments and the degree of the polynomial within a segment to achieve an error less than the tolerance error allowed. This method is suitable for collision avoidance problems between multiple UAVs because of the flexibility to increase segments and/or collocation points. The localization of the segments and the number of collocation points depend on where the conflicts take place. However, each segment adds collocation points in each iteration, so the number of collocation points could grow too large and the computation load increases. Therefore, only the segments where the conflicts take place should add new collocation points in each iteration to converge to the solution. Thus, a good segmentation should be done.

This paper considers the collision-free trajectory generation for multiple UAVs and an evaluation considering a dynamic model of UAV is carried out in order to ensure that the minimum separation is not violated during the flight. This evaluation is essential because the solution computed by the pseudospectral methods enforces the fulfillment of the constraint only in the collocation points.

The paper is organized into six sections. Section II describes the trajectory planning problem. The proposed method is explained in Section III. Simulations and experimental test are presented in Section IV and V, respectively. Finally, the conclusions are detailed in Section VI.

II. TRAJECTORY PLANNING

The problem considered in this paper is cooperative safe trajectory planning of multiple UAVs to perform coordinated missions. The proposed method to plan collision-free trajectories allows changes of the speed profile and the heading of the aerial vehicles involved in the conflict by considering the dynamic of each vehicle. The magnitude of each kind of maneuver depends on the cost function considered.

The trajectory of each aerial vehicle is given by an initial waypoint and a final waypoint. Each waypoint is defined by: 2D coordinates (x,y), speed from that waypoint (v), and the Estimated Time of Arrival (ETA) to the waypoint, t that should be met. It is assumed that all UAV trajectories are known. We consider that the UAVs maintain the safety separation if they are separated by a minimum distance, D.

The problem is solved like an optimal control problem where different global criteria can be considered. The inputs of the method are the following: Initial UAV trajectory, dynamic UAV model and ETA of each UAV. The objective is to find collision-free trajectories minimizing a cost function.

III. SEGMENTATION-ADAPTIVE PSEUDOSPECTRAL METHOD

Pseudospectral collocation methods numerically solve optimal control problems by using non-linear programming. The basic approach is to transform the optimal control problem into a sequence of nonlinear constrained optimization problems by discretizing the state and control variables [17].

The optimal problem is modeled as a Bolza problem in $\tau \in [-1,1]$ domain and the objective is to find the control input vector $u(\tau)$ and the corresponding state $\chi(\tau)$ which minimize the cost function:

$$J = \phi(\chi_{-1}, \chi_{+1}) + \int_{-1}^{1} \mathcal{L}(\chi(\tau), u(\tau), \tau) d\tau$$  \hspace{1cm} (1)

subject to the dynamic constraints

$$\dot{\chi} = f(\chi, u)$$  \hspace{1cm} (2)

inequality path constraints

$$C(\chi, u) \leq 0$$  \hspace{1cm} (3)

and the boundary conditions

$$E(\chi_{-1}, \chi_{+1}) = 0$$  \hspace{1cm} (4)

where $\phi$, $C$ and $E$ are functions. The normalized time $\tau \in [-1,1]$ and the time $t \in [t_0, t_f]$ are related by:

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f - t_0}{2}$$  \hspace{1cm} (5)

Eq. (1) should be approximated by applying quadrature rules. In this paper Legendre-Gauss-Lobatto (LGL) quadrature rule is used, so:

$$J = \phi(\chi^1, \chi^N) + \sum_{j=1}^{N} \mathcal{L}(\chi^j, \pi^j)w_j$$  \hspace{1cm} (6)

where $w_j$ are the LGL quadrature weights, and $N$ is the number of nodes or collocation points. In the used notation, the overline means discrete variables and the superscript means the collocation point used $\chi^j = \chi(\tau_j)$.

LGL nodes are defined in the normalized time domain $\tau \in [-1,1]$ as $\tau_0 = -1 < \tau_1 < \tau_2 < ... < \tau_N = 1$ where $1, 2, ..., N - 1$ are the roots of the derivative of the N-th order Legendre polynomial. The roots of the derivative of the Legendre polynomials are zeros in these nodes because they are orthogonal polynomials.

Therefore, $\chi(\tau)$ and $u(\tau)$ could be approximated by $\chi(\tau)$ and $\pi(\tau)$:

$$\chi(\tau) \approx \overline{\chi}(\tau) = \sum_{j=0}^{N} \overline{\chi}^j L_j(\tau)$$  \hspace{1cm} (7)

$$u(\tau) \approx \overline{\pi}(\tau) = \sum_{j=0}^{N} \overline{\pi}^j L_j(\tau)$$  \hspace{1cm} (8)

where $L_j(\tau)$ are the basis functions of the Lagrange interpolating polynomials of order $N$.

As it was mentioned previously, each segment adds collocation points in each iteration in the hp-adaptive pseudospectral method, presents some improvements. It considers the increase of collocation points only
within the time intervals where the constraints are not met and not in all timeline. Thus, the number of collocation points added in each iteration is reduced. The segmentation also considers the dynamics of the vehicle.

The initial segment is defined by \( t_i \) and \( t_f \). The segmentation process is executed when the solution computed in the first iteration is rejected in the validation step. The S-adaptive pseudospectral method divides the problem into segments. Each new segment is defined by time intervals where constraints are not met. One or more segments can be generated in every iteration. New collocation points are only added in each segment. Collocation points in time intervals which meet the constraints stay in the next iteration. Two parameters are considered to generate the segments:

- Horizon time: it determines how the segmentation is carried out and is added at the beginning and end of the segment. It depends on the dynamics of the UAV.
- Number of collocation points per segment: it defines how many collocation points are used and also define the degree of the polynomial of interpolation.

A validation of the whole trajectories in each iteration should check the safety distances between UAVs because the solution given by the pseudospectral method only meets the constraints in the collocation points. A discretization step is set and the positions are interpolated in time by a spline.

Figure 1 shows the performance of the S-adaptive pseudospectral method. The initial segment is defined by \( t_i \) and \( t_f \) in the first iteration (iter1). Eight collocation points are set in the initial segment (black points). The constraints are not met between the time \( t_{c1} \) and \( t_{c2} \) after the validation step, that is, collisions take place. The horizon time, \( t_h \) is added in order to define the segment (red ellipse). The three collocation points within this segment are removed and eight new collocation points are generated within of the segment in the second iteration (red points in the line iter2). The rest of collocation points are kept as in the first iteration (black points). After the validation process, two new segments do not meet the constraints in the iteration 2 (red ellipses). Again, eight new points are generated in each segment in the iteration 3 (red points). The rest of collocation points stay as the previous iteration (iter2). Now, the constraints are met after the validation step, so the iterative process stops and this is the solution of the problem.

A. Implementation

The quadrotor dynamic model used is based on [18], with several assumptions generally employed for trajectory generation. The state vector is defined by \( (x_i, \dot{x}_i, y_i, \dot{y}_i, \Phi_i, \dot{\Phi}_i, \Theta_i, \dot{\Theta}_i, t_i) \) where \( x_i, y_i \) are the 2D position of the quadrotor, \( \Phi_i, \Theta_i \) are the pitch and roll angles and \( t_i \) the time of arrival in each collocation point. The control inputs are the pitch and roll torques \( u_{\Phi_i}, u_{\Theta_i} \).

The equations of the model are the following:

\[
\dot{x}_i = \frac{T}{m} \sin(\Phi_i) \quad (9)
\]
\[
\dot{y}_i = \frac{T}{m} \sin(\Theta_i) \quad (10)
\]
\[
\dot{\Phi}_i = \frac{u_{\Phi_i}}{I_y} \quad (11)
\]
\[
\dot{\Theta}_i = \frac{u_{\Theta_i}}{I_x} \quad (12)
\]

where \( T \) is the thrust needed to maintain constant altitude, \( m \) is the total quadrotor mass, \( \Theta_i \) is the roll angle, \( \Phi_i \) is the pitch angle, and \( I_x \) and \( I_y \) are the moments of inertia with respect to the axes \( x \) and \( y \), respectively.

The total state and control vectors for a multi-UAV system are defined by concatenating the states of all the UAVs. The solution should satisfy constraints taking into account the physical limitations of each UAV and the separation between UAVs. The UAV speed will be constrained:

\[
v_{\text{min}} < v_{\text{cruise}} < v_{\text{max}} \quad (13)
\]
\[
\Phi_{\text{min}} < \Phi < \Phi_{\text{max}} \quad (14)
\]
\[
\Theta_{\text{min}} < \Theta < \Theta_{\text{max}} \quad (15)
\]

and the separation between \( UAV_i \) and \( UAV_j \) should meet:

\[
\text{distance}(UAV_i, UAV_j) \geq D \quad (16)
\]

where \( D \) is the safety distance. Moreover, the ETA should be met, so the flight time should be maintained.

IV. SIMULATIONS

The S-adaptive method has been implemented using DIDO collocation software [11]. The algorithms have been run in a PC with a CPU Intel Core i7-3770 @ 3.4 Ghz and 16 GB of RAM. The operating system used in the simulations was Windows 7 and the code has been implemented in Matlab.

First, the scalability of the method considering from two to ten UAVs is analyzed (see Table I). Fifty scenarios have been considered by randomly generating the trajectories in each case. The size of the scenarios is 15x15 meters and the cost function considered is \( J_{\text{a}} \) (see Eq. (18)) which minimizes the changes of pitch and roll angle references. The study is focused on the mean computing time as a function of the number of UAVs. Obviously, the computation time increases as the number of UAVs increases.

![Fig. 1. Several iterations of the S-adaptive pseudospectral collocation method.](image-url)
TABLE I
MEAN COMPUTING TIME WHEN THE NUMBER OF UAVS INCREASES.

<table>
<thead>
<tr>
<th>UAVs</th>
<th>Mean Time (s)</th>
<th>Standard deviation (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.4067</td>
<td>0.0733</td>
</tr>
<tr>
<td>3</td>
<td>0.7171</td>
<td>0.1621</td>
</tr>
<tr>
<td>4</td>
<td>1.2464</td>
<td>0.6382</td>
</tr>
<tr>
<td>5</td>
<td>1.7913</td>
<td>0.7876</td>
</tr>
<tr>
<td>6</td>
<td>2.6251</td>
<td>1.3515</td>
</tr>
<tr>
<td>7</td>
<td>3.0576</td>
<td>1.2229</td>
</tr>
<tr>
<td>8</td>
<td>5.0438</td>
<td>1.8169</td>
</tr>
<tr>
<td>9</td>
<td>8.7274</td>
<td>3.9441</td>
</tr>
<tr>
<td>10</td>
<td>10.2359</td>
<td>3.9312</td>
</tr>
</tbody>
</table>

Other important aspect to analyze is the optimal solution computed for different optimization criteria. Two different cost functions are evaluated: 1) minimization of the distance travelled, and 2) minimization of the changes of pitch and roll angle references. Figure 2(a) shows the scenario considered.

The cost function to minimize the distance, $J_d$ is:

$$J_d = \sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2}$$  (17)

where $M$ is the number of UAVs and $N$ is the number of collocation points.

Fig. 2. Scenarios considered in the simulations and experiments performed.

Figure 3 shows the trajectories defined by the collocation points of the solution considering $J_d$ and the scenario S1.

The speed profile computed is shown in Figure 4. In this case the changes of speed are less than the ones shown in Figure 4 because changes of heading take place.

The speed profile computed is shown in Figure 6. In this case the changes of speed are less than the ones shown in Figure 4 because changes of heading are also considered.

The S-adaptive pseudospectral method has been compared with a LGL pseudospectral method based on DIDO. The scenario considered is S2 (see Figure 2(b)). Figure 7 shows the trajectories defined by the collocation points computed by the LGL pseudospectral method and the separation among UAVs after the validation step. The number of collocation points is seven and the cost function considered is $J_d$. The separation constraint is not met between UAV2 and UAV3 (green line). The execution time is 1.326 seconds.

A valid solution is obtained by this method computing thirty collocation points, which needs several hours to obtain.

The same problem is solved with the S-adaptive pseudospectral method with eleven collocation points in 1.25 seconds approximately (see Figure 3 and Table I).
The next study analyzes how the value of the horizon time influences on the method by considering the scenario S2. The number of maximum iterations has been set to ten. The flight time of each UAV is eight seconds approximately considering the cruise velocity. The horizon time should be related to the flight time because the segmentation may tend to the initial segment of the first iteration from a given value of the horizon time. Table II shows the iterations performed to compute the solution and whether the convergence is reached. The number of iterations increases as the horizon time increases. It is demonstrated that greater horizon times need more iterations because the segmentation tends to the initial segment.

### Table II

<table>
<thead>
<tr>
<th>Horizon time (s)</th>
<th>Iterations</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>0.30</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>0.50</td>
<td>10</td>
<td>Yes</td>
</tr>
<tr>
<td>0.75</td>
<td>&gt;10</td>
<td>No</td>
</tr>
<tr>
<td>1.00</td>
<td>&gt;10</td>
<td>No</td>
</tr>
</tbody>
</table>

### V. Experiments

Several experiments have been carried out in the indoor multi-UAV testbed of the CATEC with four Hummingbird quadrotors. This system is able to provide, in real time, the position and attitude of each UAV with centimeter accuracy. The parameters used in the experiments are:

\[ v_{\text{cruise}} = 0.65 \text{m/s}; \quad v_{\text{min}} = 0.1 \text{m/s}; \quad v_{\text{max}} = 2.0 \text{m/s}. \]

This section only presents the experiments considering S1 and both cost functions, \( J_d \) and \( J_a \) (see Figure 2(a)).

\( J_d \) is considered in the first experiment. Figure 8 shows as every UAV maintains its initial spatial trajectory because the goal is to minimize the distance travelled. The safety distance is set to 1.2m. Figure 9 shows the separation between UAVs, demonstrating that the trajectories are safe.

In the second experiment, the S-adaptive pseudospectral method by considering \( J_a \) computes the solution. Changes of speed and heading for each UAV take place. Figure 10 shows the UAV real trajectories. The minimum separation is set to 0.8m. Figure 11 shows the separation between UAVs.
The main advantage of the method is its computational efficiency, its scalability and allow two maneuvers to solve the problem. Moreover, the characteristics of the method have been evaluated in many randomly generated scenarios. The more novel aspect of the paper is the real experimentation performed in order to verify the solution computed by the method.

REFERENCES