Collision Avoidance for Multiple UAVs using Rolling-horizon Policy

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

Abstract—This paper addresses the problem of collision avoidance in scenarios with multiple aerial vehicles and proposes a method based on a Legendre pseudospectral collocation in order to compute the solution trajectories and guarantee that the safety distance between them is always maintained. The method uses a rolling horizon policy in which trajectories are planned up to a given time horizon, thus considering a much smaller problem space. Then, the system is applied iteratively. Studies have been performed to set the values of the look-ahead time and the number of collocations points. The computational load and scalability of the method are also studied in randomly generated scenarios to test its application in real time. Experiments have been also carried out in the multivehicle aerial testbed of the Center for Advanced Aerospace Technologies (Seville, Spain).

I. INTRODUCTION

Multiple UAVs are being cooperatively used in the last years to carry out coordinated missions [1] [2]. Coordination and collision avoidance arises as critically important aspects in this kind of applications in dynamic environments. Therefore, a method to plan collision-free trajectories with low computational load should be implemented in order to ensure the safety in these environments. Concretely, an efficient re-planning is needed to solve the collisions detected.

Many works on planning algorithms and collision avoidance methods have been published. A detailed survey on the former is presented in [3] and [4] reviews papers on the latter. Planning methods include Rapidly-exploring Random Trees (RRT)

[1]S. Vera, J. A. Cobano, G. Heredia and A. Ollero are with the Robotics, Vision and Control Group, Engineering School, University of Seville, 41092 Seville, Spain {svera,jcobano,guiller,aollero}@us.es

[5], particle swarm optimization [6], evolutionary computation methods [7], ant colony optimization methods [8]. The main drawback of these methods is that the computation time is not predictable and the convergence to a solution is not ensured in a finite time interval. Therefore, they are not good candidates to plan collision-free trajectories in a dynamic environment and a small time horizon.

An interesting option could be to use collocation methods for trajectory generation, which have been increasingly employed in the last years. In collocation methods trajectory generation is posed as an optimal control problem and the solution is approximated by polynomials. Then, the differential equations and constraints are enforced in collocation points, and the optimal control problem is transformed into a nonlinear programming problem.

Among the most used collocation techniques are the direct collocation and the pseudospectral methods. The former divides time in several segments, computes the solution considering a fixed degree polynomial state approximation in each segment and the convergence is achieved by increasing the number of segments [9]. The latter uses a single segment and convergence is achieved by increasing the degree of the polynomial. Both methods choose the collocation points based on accurate quadrature rules and the basic functions are typically Chebyshev or Lagrange polynomials. The more commonly used pseudospectral methods are the Gauss pseudospectral method (GPM) [10], the Radau pseudospectral method [11] (RPM), and the Lobatto pseudospectral method [12] (LPM). Pseudospectral and Direct Collocation methods have been applied to compute aircraft trajectories [13] [14] [15] but their computation times are too large to plan trajectories in dynamic and uncertain environments. Moreover, most of published works consider trajectory generation for a standalone vehicle [13] [14].

A modified pseudospectral method is called hp-adaptive [16]. In this method, the number of segments and the degree of the polynomial can be increased within a segment to achieve an error less than the tolerance error allowed. However, each segment adds collocation points in each iteration, so the number of collocation points could quickly grow and that provokes larger computation times.

This paper addresses the problem of collision avoidance with multiple UAVs in dynamic environments using a rolling horizon policy. The goal is to ensure the safety and reliability of the mission. The proposed method is based on pseudospectral collocation techniques and is iterative. The information of all the UAVs is known up to a flight time defined by the look-ahead time and the dynamics of the UAVs is considered to compute more realistic trajectories. Look-ahead times are determined by a rolling horizon approximation in order to quickly compute solution trajectories to the sub-problem considered. The maneuvers allowed to solve the detected collisions are changes of speed and heading.

The main characteristic of the proposed method is the low computational load and the scalability. The look-ahead time and the number of collocation points influence the time of computation and the safety of the solution. The feasible values of these parameters are studied in this paper.

Other important improvement in the implementation is the evaluation of the whole solution trajectories because pseudospectral techniques only compute the solution valid in the collocation points. That is, the minimum separation among vehicles is maintained in these points. Therefore, an evaluation considering a model of vehicle should be carried out in order to ensure that the minimum separation is not violated during the whole trajectory to the sub-problem considered in each instant.

The paper is organized into seven sections. Section II describes the problem formulation. The proposed method is explained in Section III. Simulations and experiments performed are showed in Section IV and V, respectively. Finally, the conclusions are detailed in Section VI.

II. Path planning for multiple UAVs

The problem of collision avoidance of multiple UAVs to perform the coordinated missions is considered in this paper. The goal is to assure that the UAVs do not collide with each other. The proposed method to solve the collisions allows changes of the speed profile and the heading of the UAVs involved in the conflict.

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

The inputs of the method are the following:
- Model of each UAV
- Look-ahead time
- Number of nodes per segment

Number of nodes per segment, also known as collocation points, defines the degree of the polynomial of interpolation used in each segment. Look-ahead time is the time in which each UAV knows the information of the rest of UAVs. This time interval defines the sub-problem of the trajectory planning of each UAV. Both look-ahead time and the number of collocation points should be analyzed. The best values of both parameters should ensure a safe solution in each computation and with low computational load to iteratively perform trajectory re-planning in dynamic environment.

Different optimization criteria can be considered. In this paper, two criteria have been implemented: minimize the changes of the heading angle of each UAV and minimize the changes of the control inputs (pitch and roll angles).

III. Legendre pseudospectral method

The pseudospectral method numerically solves optimal control problems. 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. It computes a set of collocation points that provides an accurate approximation to the solution of the optimal control problem. The proposed method is based on a Legendre pseudospectral method known as DIDO [17]. Its novelty is how an iterative implementation is carried out in order to obtain an efficient trajectory re-planning.

The optimal problem is modeled as a Bolza problem in $\tau \epsilon [-1, 1]$ domain and the objective is to find

the control input vector $u(\tau)$ and the corresponding state $\chi(\tau)$ which minimize the cost function:

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

subject to the dynamic constraints

$$\dot{\chi} = f(\chi, u) \tag{2}$$

inequality path constraints

$$C(\chi, u) \leq 0 \tag{3}$$

and the boundary conditions

$$E(\chi(-1), \chi(+1)) = 0 \tag{4}$$

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

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f - t_0}{2} \tag{5}$$

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

$$J = \phi(\overline{\chi}^1, \overline{\chi}^N) + \sum_{j=1}^{N} \mathcal{L}(\overline{\chi}^j, \overline{u}^j)w_j \tag{6}$$

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 $\overline{\chi}^j = \chi(\tau_j)$.

LGL nodes are defined in the normalized time domain $\tau \epsilon [-1, 1]$ 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 $\overline{\chi}(\tau)$ and $\overline{u}(\tau)$:

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

$$u(\tau) \approx \overline{u}(\tau) = \sum_{j=0}^{N} \overline{u}^j L_j(\tau) \tag{8}$$

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

The proposed method is iterative in such a way that it computes an optimal solution to a sub-problem in every iteration. The look-ahead time, T_{la}, is determined by the knowledge of the environment in every iteration and the maximum speed of the vehicle that is the worst case for trajectory re-planning. The number of collocation points considered in each iteration, N_c, also influences the proposed method. This number affects the quality of the solution and the computation time. The quality means that the solution could be flown and the minimum separation distance should be met by all UAVs. If constraints are met in every piece of the path, it would ensure the minimum separation distance in the whole trajectory. Moreover, a low computational time is required. Also, the frequency of computation should be determined in order to ensure that each UAV does not fly the whole trajectory computed in the previous iteration during the corresponding time between two iterations, T_f. This time should be larger than the computation time of the UAV trajectories, T_c:

$$T_f > T_c \tag{9}$$

Figure 1 illustrates the performance of the proposed method. Five collocation points have been considered (blue points) and a look-ahead time is considered from the knowledge of the environment in every iteration and the maximum speed of the vehicle. First, the trajectory of each UAV is computed within the look-ahead time. An iteration is computed every T_f, so the computation time, T_c in the second iteration should fulfill:
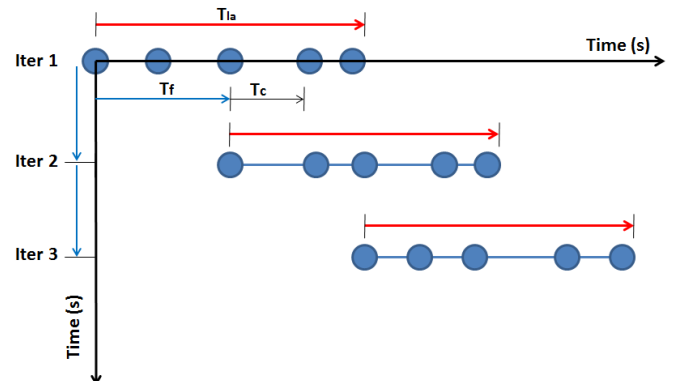
$$T_{la} - T_f > T_c \tag{10}$$

Figure 1. Description of the proposed method.

Finally, a model of UAV by considering a simplified dynamics is used to compute more realistic trajectories. The altitude is assumed to be constant. The state vector is defined by $(x_i, y_i, \Phi_i, \Theta_i, t_i)$ where x_i, y_i the 2D position of the aerial vehicle, Φ_i, Θ_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_{Φ_i}, u_{Θ_i}.

The model considered is:

$$\ddot{x}_i = \frac{T}{m} \cdot sin(\Phi_i) \tag{11}$$

$$\ddot{y}_i = \frac{T}{m} \cdot sin(\Theta_i) \tag{12}$$

$$\ddot{\Theta}_i = \frac{u_{\Theta_i}}{I_y} \tag{13}$$

$$\ddot{\Phi}_i = \frac{u_{\Phi_i}}{I_x} \tag{14}$$

where T is the thrust needed to maintain the constant altitude, m is the mass, Θ_i is the roll angle, Φ_i is the pitch angle, I_x and I_y are the moments of inertia with respect to the axes x and y, respectively.

The multi-UAV system is defined by concatenating the state of all the UAVs. Therefore, the state vector and control vector are defined as follows:

$$X = [x_1, y_1, \Phi_1, \dot{\Phi}_1, \Theta_1, \dot{\Theta}_1, t_1, ...,$$
$$x_n, y_n, \Phi_n, \dot{\Phi}_n, \Theta_n, \dot{\Theta}_n, t_n] \tag{15}$$

$$U = [u_{\Phi_1}, u_{\Theta_1}, u_{\Phi_2}, u_{\Theta_2}...., u_{\Phi_n}, u_{\Theta_n}] \tag{16}$$

where n is the number of UAVs.

The solution should satisfy constraints taking into account the physical limitations of the inputs of each UAV:

$$u_{\theta min} \leqslant u_{\theta_i} \leqslant u_{\theta max} \tag{17}$$

$$u_{\Phi min} \leqslant u_{\Phi_i} \leqslant u_{\Phi max} \tag{18}$$

And the separation between UAV_i and UAV_j should meet:

$$distance(UAV_i, UAV_j) \geq D \tag{19}$$

where D is the safety distance.

IV. SIMULATIONS

The method has been tested by carrying out many simulations. Different scenarios randomly generated with several UAVs have been considered.

The problems have been solved with the pseudospectral LGL collocation method, using the optimal control software named DIDO [17]. These 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 OS and the code has been implemented in Matlab.

First, the values of the look-ahead time, T_{la}, and the number of collocation points, N_c, should be set. A study considering one hundred random scenarios with five UAVs has been performed to analyze the best values. The chosen values should ensure a safe solution with a low computation time. The possible values are: $T_{la} = [1.0, 1.5, 2.0, 2.5, 3.0]s$ and $N_c = [3, 4, 5, 6, 7, 8]$ points. All the combinations of these values are explored in each scenario. The obtained mean computation time and its standard deviation for each combination which ensures a safe solution in all the scenarios (that is, satisfies all the constraints) is shown in Table I. Only eight combinations met it. In order to meet (9), T_f should be greater than $1.107s$ (the larger value of $T_c + \sigma_{Tc}$ in Table I), so $T_f = 1.25s$ is considered. First, second, fourth and sixth cases only meet (10). Among these options, $T_{la} = 2.5s$ and $N_c = 4$ are chosen because they ensure a minimum computation time.

Table I
COMBINATIONS THAT ENSURES A SAFE SOLUTION.

$T_{la}(s)$	N_c	T_c (s)	σ_{Tc} (s)
2.5	4	0.808	0.207
2.5	3	0.819	0.212
1.0	4	0.815	0.258
3.0	3	0.824	0.304
1.5	3	0.837	0.197
2.5	6	0.847	0.219
1.0	7	0.851	0.207
2.0	6	0.873	0.234

Once both parameters have been set, the scalability is analyzed, that is, how the computation time depends on the number of UAVs. Figure 2 shows the scenario considered with up to eight UAVs. Note

that in case of 7 or 8 UAVs, the density of UAVs in the scenario is very high.
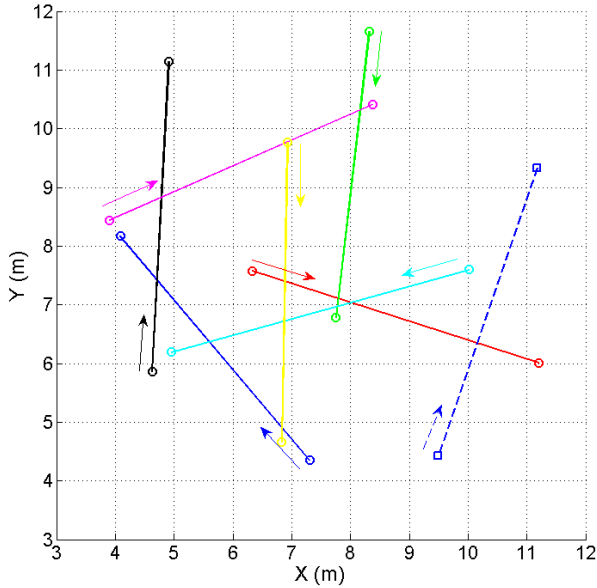

Figure 2. Scenario S1 considered in the simulations with up to eight UAVs: UAV1 in blue, UAV2 in red, UAV3 in black, UAV4 in green, UAV5 in pink, UAV6 in clear blue, UAV7 in yellow and UAV8 in dashed blue.

Table II shows the computation mean time and standard deviation of the scalability test. Note that scenario S1 is considered. The proposed method adapts well considering up to seven UAVs because (9) is met.

Table II
MEAN COMPUTING TIME WHEN THE NUMBER OF UAVS INCREASES BY CONSIDERING THE SCENARIO S1.

Number of UAVs	UAVs	Time (s)	σ_t (s)
3	1-3	0.308	0.150
4	1-4	0.374	0.204
5	1-5	0.480	0.269
6	1-6	0.689	0.279
7	1-7	0.961	0.294
8	1-8	1.297	0.472

Figure 3 presents the scenario S2 considered with four UAVs to show how the proposed method computes a solution every T_f. Figures 4, 5, 6 and 7 shows four instants which correspond to $t = 2.5, 7.5, 10.0, 16.5s$. The sub-problems solved are presented in every iteration. Note that each UAV trajectory converges to its goal waypoint.
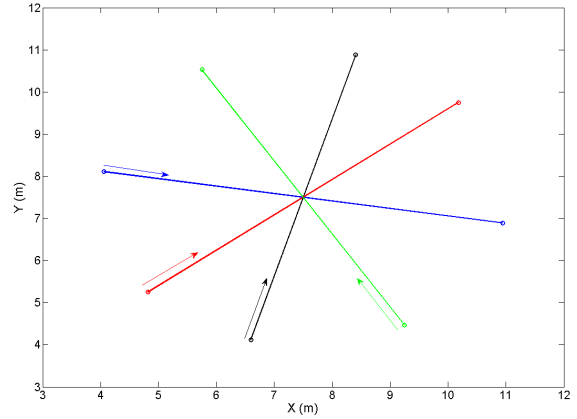
Figure 3. Scenario S2 with four UAVs: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.
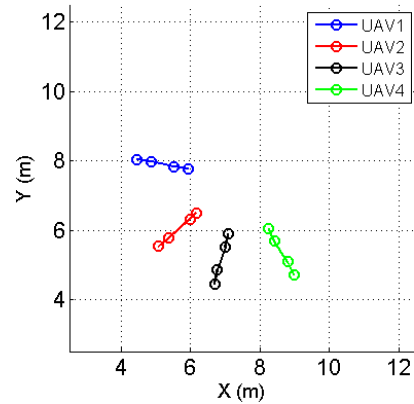

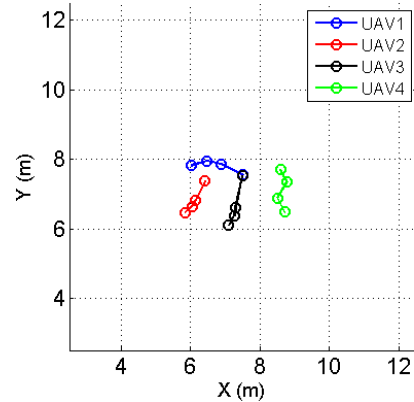Figure 4. Trajectory computed by the method at the instant $t = 2.5s$.

Figure 5. Trajectory computed by the method at the instant $t = 7.5s$.

The whole trajectories are shown in Figure 8. The speed profile obtained is presented in Figure 9.
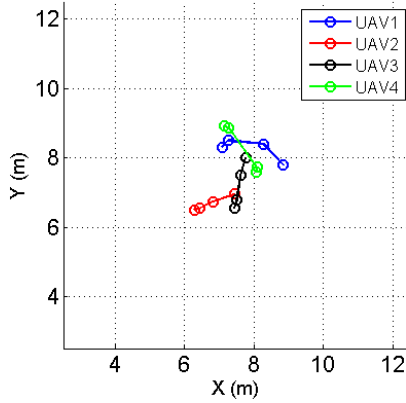
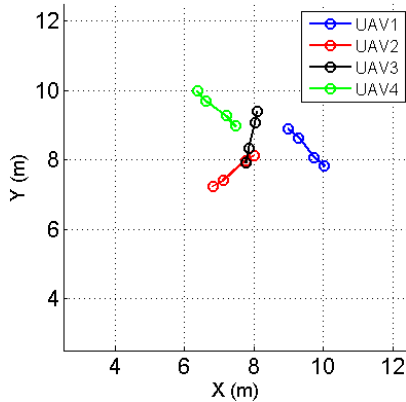Figure 6. Trajectory computed by the method at the instant $t = 10.0s$.

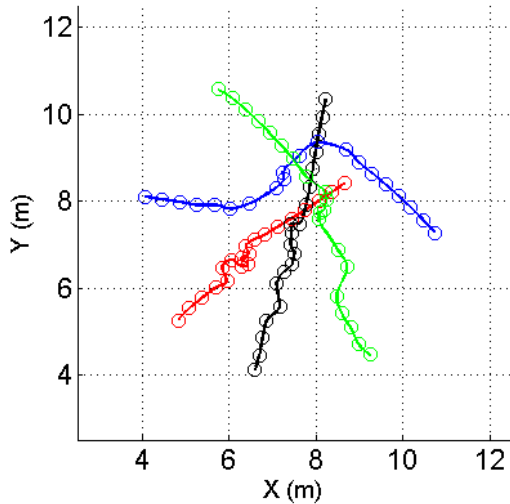Figure 7. Trajectory computed by the method at the instant $t = 16.5s$.

Figure 8. Final trajectory of scenario S2: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.

V. EXPERIMENTS

Experiments have been carried out in the indoor multi-UAV testbed of the CATEC with four Hum-
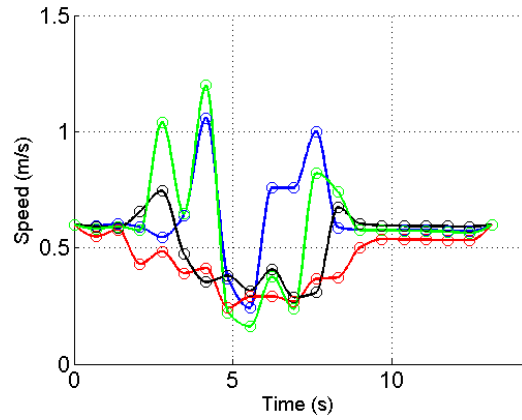

Figure 9. Final speed profile of scenario S2: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.

mingbird quadrotors (see Figure 10) with up to 20 minutes flight autonomy. The testbed has an indoor localization system based on 20 VICON cameras. This system is able to provide, in real time, the position and attitude of each UAV with centimeter accuracy.

Figure 10. Indoor multi-UAV testbed of the CATEC with Hummingbird quadrotor from Ascending technologies.

This section shows an experiment to demonstrate the performance of the proposed method. The minimum separation is $1.0m$. In this case, the optimization criterion is to minimize the heading changes. Figure 11 shows the scenario considered. Two collisions are detected. Four quad-rotors in their initial position are shown in Figure 12.

The whole solution trajectories computed are shown in Figure 13. The speed profile obtained is presented in Figure 14.
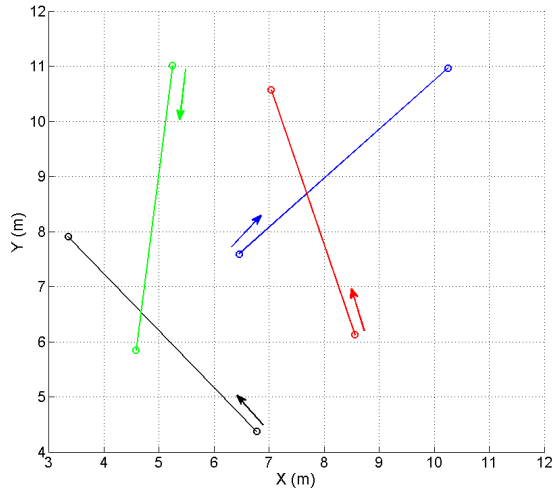
Figure 11. Scenario considered in the experiment with four UAVs: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.
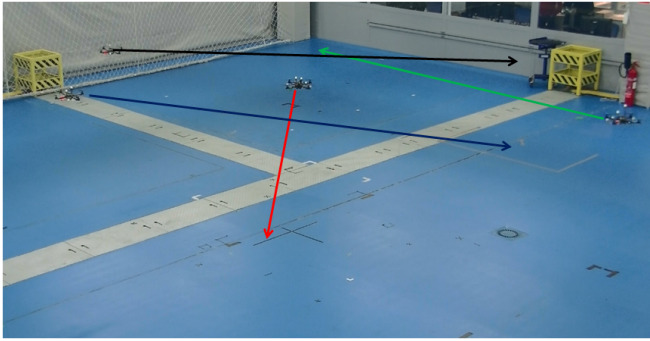

Figure 12. Four quad-rotors considered in the experiment. Each quad-rotor is in its initial position.
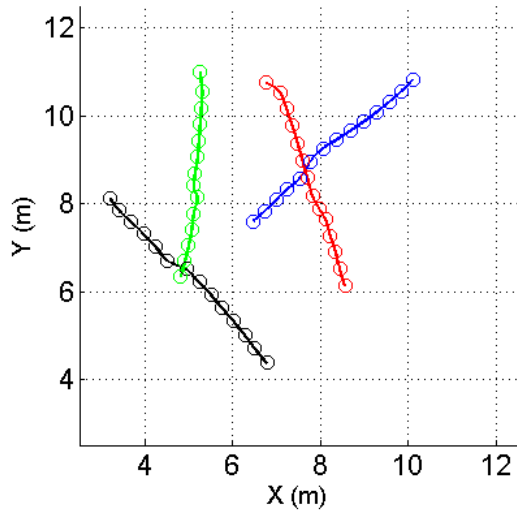

Figure 13. Solution trajectories defined by the waypoints computed: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.

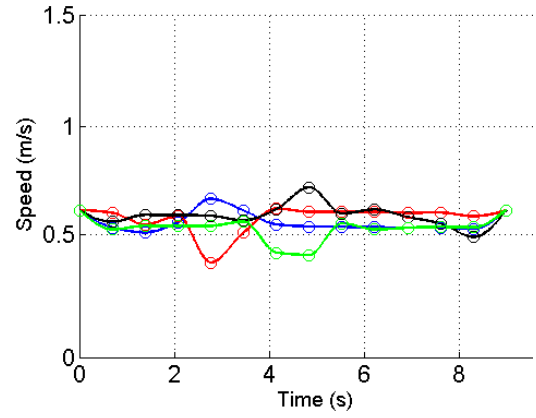Each UAV real trajectory is represented in Figure 15 and Figure 16 shows the separation among

Figure 14. Speed profile of each UAV: UAV1 in blue, UAV2 in red and UAV3 in black and UAV4 in green.

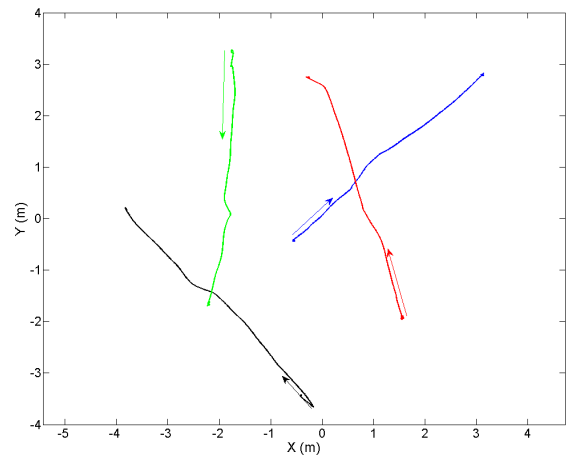UAVs. Each UAV maintains the minimum separation distance.

Figure 15. UAV real trajectories in the experiment: UAV1 in black, UAV2 in blue, UAV3 in red and UAV4 in green.

VI. CONCLUSIONS

This paper addresses the problem of collision avoidance with multiple UAVs in coordinated missions by ensuring the safety and reliability of the mission in dynamic environments. A method is proposed to efficiently re-plan each UAV trajectory and perform the mission. The method takes into account the dynamics of the vehicles to compute more realistic trajectories. It is based on a Legendre pseudospectral collocation to generate trajectories and a rolling horizon policy is used. It is applied iteratively and collision-free trajectories are planned in a sub-problem defined by the time horizon.
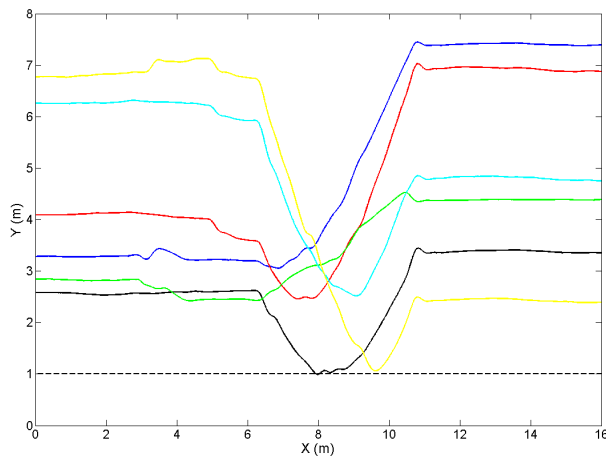
Figure 16. Separation between UAVs in the experiment: UAV1-UAV2 in black, UAV1-UAV3 in blue, UAV1-UAV4 in red, UAV2-UAV3 in green, UAV2-UAV4 in clear blue, UAV3-UAV4 in pink and the minimum separation in dashed black line.

Two maneuvers are allowed: changes of speed and heading.

The main advantage of the proposed method is its low computational load. Another novel aspect is to consider multiple UAVs and two possible maneuvers. Most of works published on collocation techniques for UAVs consider one vehicle [13] [16].

Look-ahead time and number of collocation points are the more relevant parameters of the method because of their influence on the computation time. A lot of tests have been carried out to ensure safe solutions and low computation times by considering random scenarios. The scalability of the method has been analyzed.

Finally, real experiments have been carried out in the multivehicle aerial testbed of the Center for Advanced Aerospace Technologies (Seville, Spain).

REFERENCES

[1] J. A. Cobano, J. R. Martínez-de Dios, R. Conde, J. M. Sánchez-Matamoros, and A. Ollero, "Data retrieving from heterogeneous wireless sensor network nodes using UAVs," *Journal of Intelligent and Robotic Systems*, vol. 60, no. 1, pp. 133–151, 2010.

[2] L. Merino, F. Caballero, J. M. de Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1, pp. 533–548, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10846-011-9560-x

[3] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent Robot Systems*, vol. 57, pp. 65–100, 2010.

[4] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 179–189, 2000.

[5] S. M. Lavalle, J. J. Kuffner, and Jr., "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.

[6] D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero, "Particle Swarm Optimization for collision-free 4d trajectory planning in unmanned aerial vehicles," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, USA, 28-31 May 2013, pp. 298–307.

[7] R. Conde, D. Alejo, J. A. Cobano, A. Viguria, and A. Ollero, "Conflict detection and resolution method for cooperating unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 495–505, 2012, 10.1007/s10846-011-9564-6.

[8] N. Durand and J. Alliot, "Ant colony optimization for air traffic conflict resolution," in *Proceedings of the Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, Napa, (CA, USA), 2009.

[9] J. T. Betts, *Practical Methods for Optimal Control using Nonlinear Programming*. SIAM Press: Philadelphia, 2011.

[10] D. A. Benson, G. T. Huntington, T. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.

[11] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[12] G. Elnagar, M. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.

[13] B. R. Geiger, J. F. Horn, G. L. Sinsley, J. A. Ross, and L. N. Long, "Flight testing a real time implementation of a UAV path planner using direct collocation," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, South Carolina, USA, 20-23 August 2007.

[14] G. Basset, Y. Xua, and O. A. Yakimenkob, "Computing short time aircraft maneuvers using direct methods," *Journal of Computer and Systems Sciences International*, vol. 49, no. 3, pp. 481–513, 2010.

[15] K. P. Bollino and L. R. Lewis, "Collision-free multi-UAV optimal path planning and cooperative control for tactical applications," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 18-21 August 2008, pp. 1–18.

[16] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 476–502, 2010.

[17] I. M. Ross and F. Fahroo, "Users manual for dido 2002: A matlab application package for dynamic optimization," in *NPS Technical Report AA-02-002,*, Department of Aeronautics and Astronautics, Naval Postgraduate School, Monterey, CA. (USA), 2002.