J Intell Robot Syst (2012) 65:495–505
DOI 10.1007/s10846-011-9564-6

# Conflict Detection and Resolution Method for Cooperating Unmanned Aerial Vehicles

**Roberto Conde · David Alejo · Jose Antonio Cobano · Antidio Viguria · Aníbal Ollero**

Received: 3 April 2011 / Accepted: 6 April 2011 / Published online: 16 August 2011


**Abstract** This paper presents a Conflict Detection and Resolution (CDR) method for cooperating Unmanned Aerial Vehicles (UAVs) sharing airspace. The proposed method detects conflicts using an algorithm based on axis-aligned minimum bounding box and solves the detected conflicts cooperatively using a genetic algorithm that modifies the trajectories of the UAVs with an overall minimum cost. The method changes the initial flight plan of each UAV by adding intermediate waypoints that define the solution flight plan while maintaining their velocities. The method has been validated with many simulations and experimental results with multiple aerial vehicles platforms based on quadrotors in a common airspace. The experiments have been carried out in the multi-UAV aerial testbed of the Center for Advanced Aerospace Technologies (CATEC).

**Keywords** Conflict detection and resolution · Cooperative aerial vehicles · Genetic algorithms

R. Conde · D. Alejo · J. A. Cobano (✉) · A. Ollero
Robotics, Vision and Control Group (GRVC), University of Seville, Seville, Spain
e-mail: jacobano@cartuja.us.es

R. Conde
e-mail: rconde@cartuja.us.es

D. Alejo
e-mail: dalejo@cartuja.us.es

A. Ollero
e-mail: aollero@cartuja.us.es

A. Viguria · A. Ollero
Center for Advanced Aerospace Technologies, (CATEC), Seville, Spain

A. Viguria
e-mail: aviguria@catec.aero

A. Ollero
e-mail: aollero@catec.aero

## 1 Introduction

The research and development activities on systems of multiple Unmanned Aerial Vehicles (UAVs) have experienced a significant increase in the last years [1–4]. In all these cases a common problem is to maintain a safety separation among all UAVs. Therefore, a Conflict Detection and Resolution (CDR) method is needed in all the Multi-UAV scenarios.

In [5] a detailed survey on CDR techniques is presented. In [6] all UAVs involved in a collision change their speed profile in a centralized way to solve the collision. Other methods are based on mixed-integer linear program (MILP) [7, 8]. In [7] the conflict is solved by changing speed to a large number of aerial vehicles. However, some conflicts cannot be solved just changing velocities of aerial vehicles which are subject to velocity change constraints. On the other hand, methods like the one presented in [8] do not consider mobile obstacles. A method for multiple aerial vehicle conflict avoidance is proposed in [9]. It is assumed that aerial vehicles cruise at constant altitude with varying velocities and that conflicts are solved in the horizontal plane using heading change, velocity change, or a combination thereof. The algorithm presented in [10] is based on a geometric approach for pairwise non-cooperative aircrafts collision avoidance.

Other CDR methods have been proposed such as application of game theory approach [11], ant colony optimization [12], evolutionary techniques [13], and multi-objective evolutionary algorithms [14].

It is important to point out that solving a CDR problem considering multiple mobile UAVs is NP-hard [15]. Some differential constraints given by the model of the aerial vehicle should be considered. Sampling-based techniques, as opposed to combinatorial techniques, are usually preferred in these NP-hard problems [16]. Therefore, the application of evolutionary techniques is an efficient and effective alternative for this problem [13].

In this paper, a cooperative CDR method has been implemented where the detection algorithm is based on axis-aligned minimum bounding box and the resolution algorithm is based on genetic algorithms. Each UAV changes its trajectory maintaining its velocity to solve the detected conflicts collaborating with the rest of vehicles. The method is validated with many simulations and flight experiments using four quad-rotors.

This paper is organized as follows. Section 2 states the problem formulation. Section 3 describes the proposed CDR method and its implementation. Section 4 describes the testbed used to carry out the experiments and the obtained results in simulations and experiments. Finally, conclusions and future work are described in Section 5.

## 2 Problem Formulation

Consider the detection and resolution of conflicts between unmanned aerial vehicles (UAVs) in a common airspace. The UAVs can fly in different flight levels and the separation among them should be greater than a given safety distance (horizontal and vertical minimum separation). It is also assumed that velocity changes are not allowed. The solution only considers the addition of intermediate waypoints. Therefore, after a possible collision is detected, the problem is solved when a collision-free trajectory for each UAV is computed, where the trajectory is defined by a sequence of waypoints. All UAVs cooperate to solve the problem changing their initial trajectory.

The information needed to solve the problem is the following:

1. the sequence of waypoints that each UAV will follow
2. the parameters of the model of each UAV in the airspace
3. the initial location and goal location of each UAV

The objective is to find collision-free trajectories while minimizing the changes of the trajectory of each aerial vehicle. The initial trajectories generating the conflict and the final collision-free trajectories than solve the problem should have the same initial and goal locations.

## 3 Proposed CDR Method

The proposed CDR method can be split into the following two steps.

### 3.1 Detection Algorithm

The proposed detection algorithm is based on axis-aligned minimum bounding box. This technique presents as advantages a low time of execution, which is required for real-time implementation, and the need of few parameters to describe the system: each box is defined by the three intervals, one by axis. A point will be in the box if each one of its coordinates belongs to the correspondent intervals of the box.

On the other hand the above technique presents two disadvantages: it is not very accurate and it depends on the coordinate axes.

Each aerial vehicle is represented with two joined boxes, horizontal and vertical box, with a common centre (see Fig. 1). The measurements

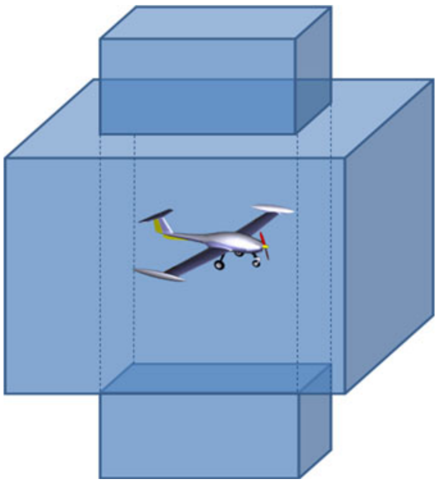


**Fig. 1** Detection algorithm based on axis-aligned minimum bounding box. Each aerial vehicle is described by two boxes

of the horizontal box are related to the minimum horizontal separation between aerial vehicles while the vertical box is related to the vertical separation. It is also possible to relate the dimensions of these boxes to the uncertainty in the predicted trajectories [6]. Thus, the minimum separation, $S$, between two aerial vehicles is defined by the dimension of both joined boxes.

A collision is detected when there is an overlap between the intervals that define each box (see Fig. 2). Thus, the 3D problem is reduced to three problems of overlaping intervals, one in each coordinate. Let us consider the intervals in one coordinate $A = [A_i, A_e]$ and $B = [B_i, B_e]$. The condition of overlap for this coordinate is given by:

$$(A_e > B_i) \wedge (A_i < B_e) \tag{1}$$

### 3.2 Resolution Algorithm

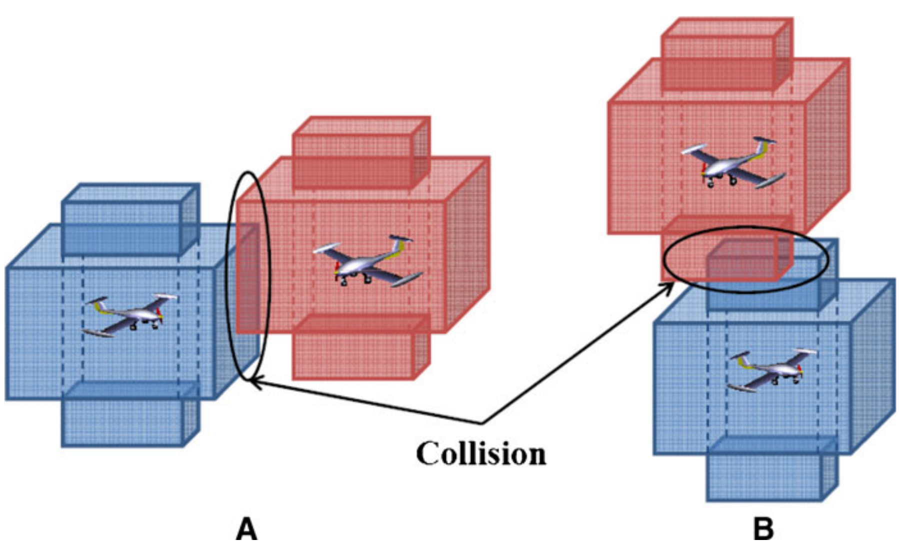
The proposed conflict resolution algorithm is based on genetic algorithms and is aimed to achieve an optimal, or near-optimal, solution under specific constraint conditions. The collision-free trajectory of each aerial vehicle is computed by generating intermediate waypoints between its initial location and its goal location.

Figure 3 shows a flow chart of the algorithm. The individuals are coded by sequences of waypoints that represent a possible trajectory for each aerial vehicle. The fitness of each individual is computed by means of the following cost function:

$$Cost_i = L_i + P_{i,collision} \tag{2}$$

where $i$ indicates the *ith* iteration, $L_i$ is the sum of the length of each aerial vehicle trajectory and $P_{i,collision}$ is the penalty added when a collision is detected. In this case, $P_{i,collision} = 10L_i$. The crossover and mutation operators are considered in the algorithm. By iteration of the selection and reproduction processes, the algorithm ends up

**Fig. 2** System described by boxes. $A$ and $B$ overlap (collision)



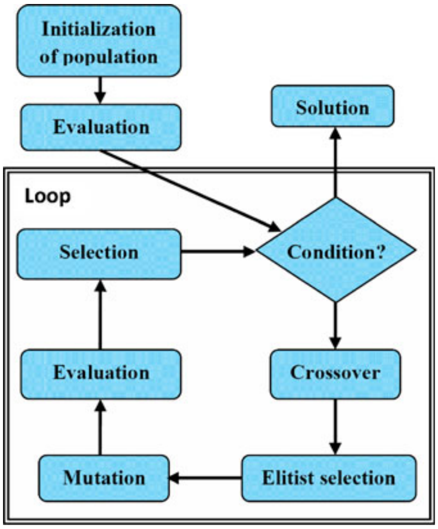



**Fig. 3** Flow chart of the resolution algorithm based on genetic algorithms

computing a near-optimal solution that ensures a collision-free trajectory for each aerial vehicle.

In particular, the chosen operators are uniform crossover and Gaussian mutation with standard deviation equal to one. The selection function is the Roulette Wheel [20] and the algorithm ends when it gets to a given number of iterations.

A model of the aerial vehicle is needed to simulate and evaluate the suitability of the generated trajectories. Different models [6, 17, 18] can be used in the proposed algorithm. The selection of the model should be done taking into account the allowable executing time and accuracy needed.

## 4 Simulations and Experiments

Many simulations and several experiments have been carried out to validate the proposed method. The experiments have been carried out in the multi-UAV testbed of the Center for Advanced Aerospace Technologies (CATEC) (see Section 4.1).

The algorithms have been run in a PC with a 2 GHz Dual Core processor and 2 GB of RAM. The operating system was Kubuntu Linux with kernel 2.6.32. The code has been written in the C++ language and compiled with gcc-4.4.1.

**Fig. 4** Hummingbird quadrotor from ascending technologies used in the experiments

Taking into account the characteristics of the aerial vehicles involved in the simulations and experiments (see Figs. 4 and 5), the detection algorithm considered the following dimensions of each box: horizontal box 1.5 m × 1.5 m × 1 m and the vertical box 0.8 m × 0.8 m × 1.6 m. Thus, the minimum separation, $S$, between two aerial vehicles is defined by the dimension of the box. The minimum horizontal separation between aerial vehicles in $XY$ plane was $S_{xy} = 1.5$ m, and the vertical separation in $Z$ axis, $S_z = 1.6$ m.

Next sections describe the multi-UAV testbed and the results obtained in simulations and experiments.

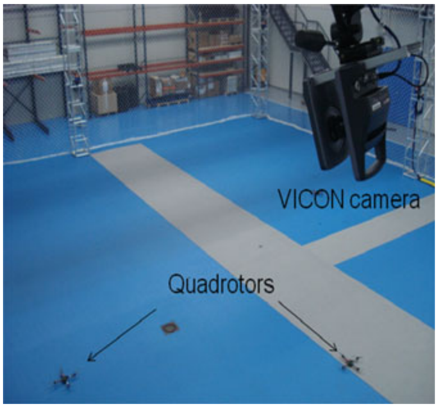



**Fig. 5** CATEC's testbed

### 4.1 Indoor Multi-UAV Aerial Testbed

CATEC facilities have an indoor multi-UAV testbed that can be used to develop and test cooperation algorithms applied to multiple aerial vehicles (see Fig. 5). The useful volume where tests can be conducted is a box with a base of 14 × 14 meters and 5 meters height. The testbed has an indoor localization system based on 20 VICON cameras (see Fig. 5) that only needs the installation of passive markers on each of the aerial vehicle. This system is able to provide, in real-time, the position and attitude of each aerial vehicle with centimeter accuracy, even if we are conducting test with more than 10 quadrotors.

The quadrotors used in these experiments (see Fig. 4) have 200gr payload and up to 20 minutes flight autonomy.

### 4.2 Simulations

In order to check the properties of the proposed CDR method, a comprehensive set of tests has been carried out. This section is divided in two subsections. The first is devoted to the design of the set of tests. The second one shows the results of the tests and an analysis of the method. The aim is to know the characteristics of the proposed CDR method with respect to time of execution, cost and number of iterations needed to compute a particular level of optimality. Thus, these parameters can be configured depending on the specifications of the problem.

#### *4.2.1 Test Set Design*

Whenever a new collision-free path planning algorithm is studied, a problem of the method arises: the definition of a metric to evaluate the results. In cases of difficult path or motion planning problems for only one mobile object, there are some *de facto* benchmark standards in the academic context, like the bug trap or the alpha test [19]. However, this is not the case when dealing with multiple object planning.

Therefore, a test set has been developed in a given scenario to validate the proposed method. This set provides a way to measure the properties of the method regarding time of execution, optimization and level of scalability with number of UAVs. Furthermore, the test set and the design methodology can be useful for comparison with other methods developed in the future.

The scenario has a base of 20 × 20 dimensional units and 10 dimensional units of height. Different problems are defined considering the same

**Fig. 6** Random test generation algorithm

```
1. For each test
2.   While the test is not valid
3.       For each UAV
4.           Choose a random entry face
5.           Choose a random exit face from the resting 3
6.           Choose entry and exit points from the
             corresponding entry and exit faces
7.           Add M random intermediate waypoints
8.           Check for the plan validity
9.       end for
10.  end while
11.end for
```

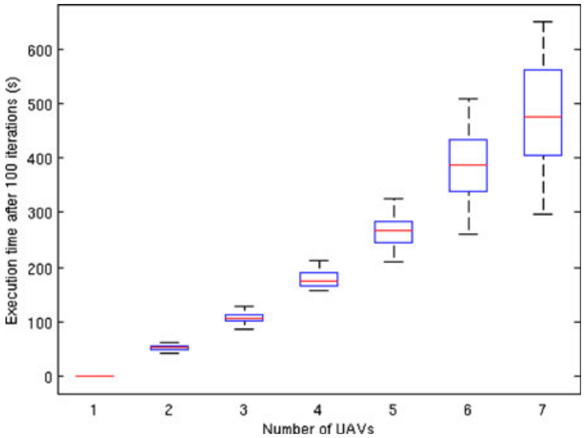



**Fig. 7** Time of execution against number of UAVs after 100 iterations

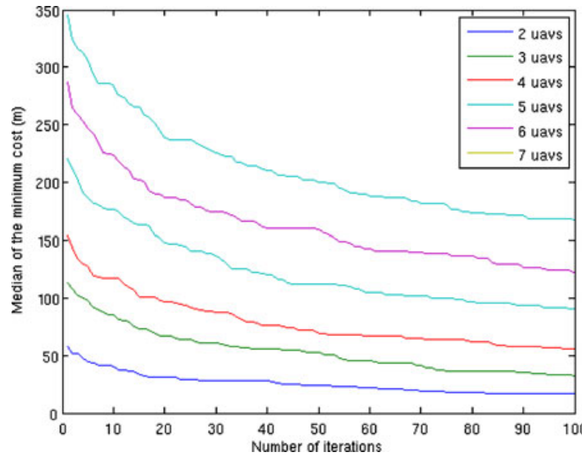



**Fig. 9** Median of minimum cost of the population throughout successive iterations

scenario, as well as the same random problem generation process.

Each problem is formulated as a set of entry and exit points located in one of the lateral faces of the scenario, i.e., nor entry nor exit points are allowed in the top and bottom faces of the scenario. The faces are sampled into a discrete grid in order to have a finite set of allowed entry and exit points.

The adopted strategy is regressive. Random candidate solutions are generated and the problem is defined using them when they are found.

The random generation process of the tests is performed following the algorithm in Fig. 6. For each UAV, an entry face is randomly chosen, selecting an uniformly random number between 1 and 4 (line 4). Then, the exit face is randomly selected from the resting 3 faces (line 5). Entry and exit points are randomly selected from the corresponding face grid (line 6). A certain number, *M*, of intermediate waypoints inside of the scenario along with the entry and exit points define the flight plan.

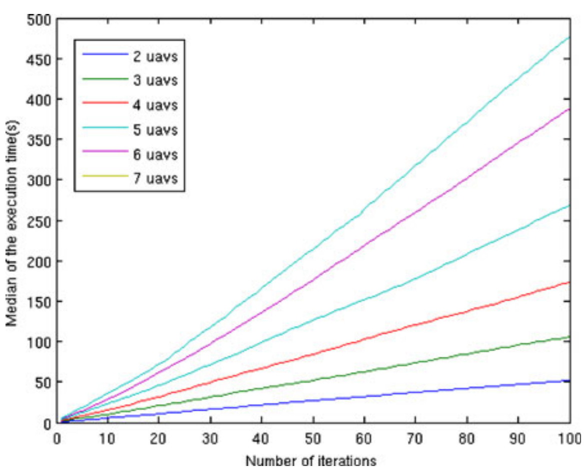



**Fig. 8** Time of execution against number of iterations. Parameterized for number of UAVs

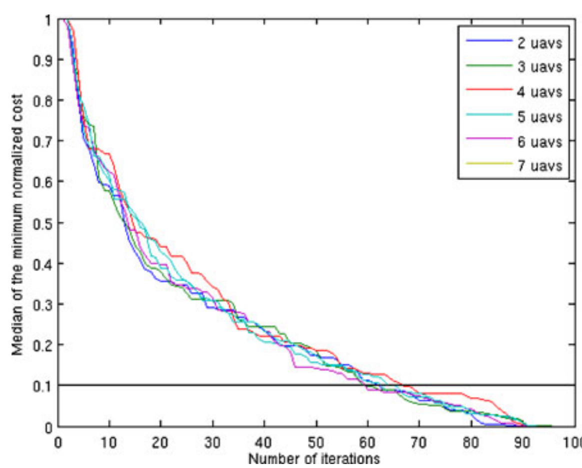



**Fig. 10** Normalized cost throughout successive iterations. The *line* marks the 90% optimality

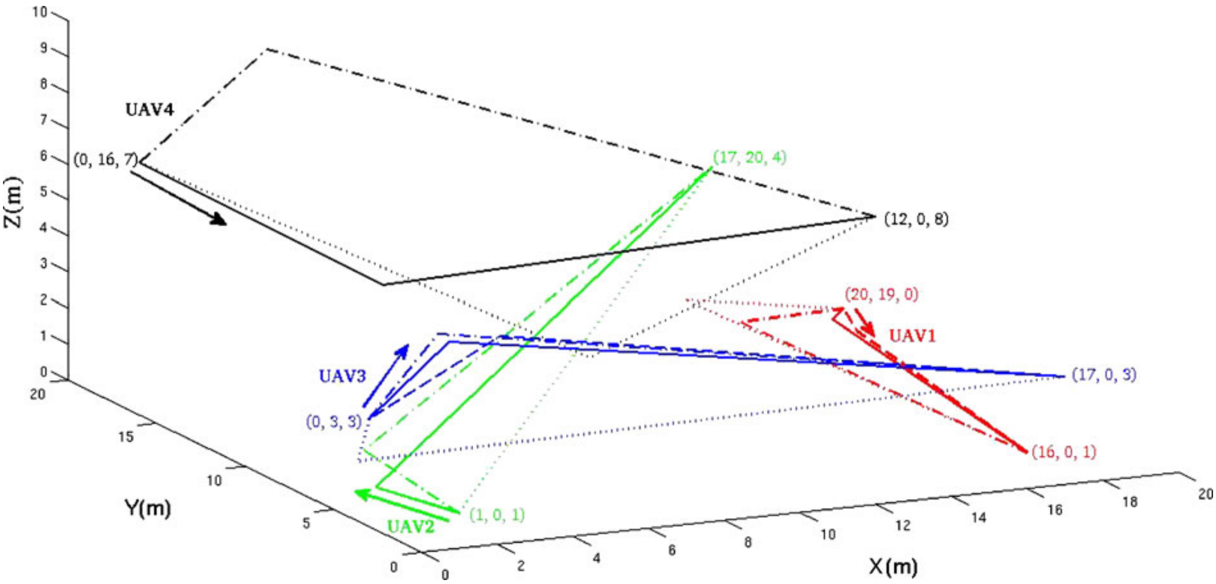



**Fig. 11** Evolution of 3D flight plans with 4 UAVs throughout various iterations: 6th in *dotted line*, 12th in *dash dotted line*, 18th in *dashed line* and 25th iteration in *solid line*

The line 8 of the algorithm should ensure the following:

- The solution is valid, i.e. UAVs do not collide
- The initial plans generate a conflict, i.e. the UAVs initial plans lead to collision

The test set consists of 90,000 different problems grouped by the number of UAVs involved, from 2 to 10, in subsets of 10,000 tests. This classification, using the number of UAVs, is useful to study the scalability characteristics of the method (see Section 4.2.2).

### *4.2.2 Test Execution Results*

The dimensions of the scenario are 20 m × 20 m × 10 m. The number of intermediate waypoints, $M$, is set to 1.

The tests have been carried out in the same computer (see Section 4) and under the same conditions. This is relevant for the time-related comparison and scalability performance figures.

40 tests have been performed for each subset.

Figure 7 shows the rise of the time of execution with the number of UAVs involved. Each box of the figure depicts statistics of the 40 tests performed for a given number of UAVs. The central mark is the median, the edges of each box are the 25th and 75th percentiles, and the whiskers extend to the extreme data points.

Another interesting indicator is the relation between the time of execution and the number of iterations performed, and its dependence with the number of UAVs (see Fig. 8). This relation should be linear and additive, showing that each iteration

**Fig. 12** Experiment with four quadrotors

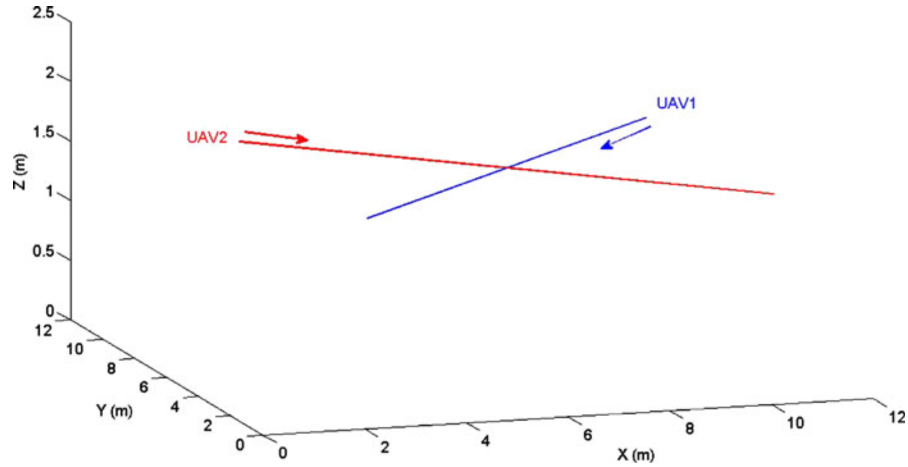



**Fig. 13** Initial trajectories in the experiment I. The height is the same for both aerial vehicles: z = 1.5 m

has the same computational cost. On the other hand, the slope usually depends on the number of UAVs.

The aim of the proposed method is to find a better solution as time passes, i.e. a smaller cost each iteration. Figure 9 shows the evolution of the minimum costs with the iterations. The median of the minimum costs computed in the population of all the tests has been chosen as statistical indicator.

This indicator can be useful to have a measure of how much time it would cost to achieve a certain level of optimality. This relates the cost in a given iteration to the obtained minimum cost in the corresponding problem. Figure 10 shows a normalization of the cost against the number of iterations. As an example, a line that marks the required number of iterations to compute for a 90% level of optimality is drawn. If the test set is executed in the same computer where the user has installed the proposed method, Fig. 8 will provide an estimation of the time needed for that number of iterations, and therefore, that level of optimality.

For the cost normalization, a linear transformation, $f(x) = ax+b$, is applied to the actual cost values to set them in the range [0,1]. Therefore $a$ and $b$ are chosen in such a way that the maximum cost equals to 1 and the minimum cost equals to 0. Therefore, $a = 1/(Cost_{max} - Cost_{min})$ and $b = Cost_{min}/(Cost_{min} - Cost_{max})$.

As an example of the general operation of the proposed CDR method, Fig. 11 shows the evolution of the flight plans of 4 UAVs. The flight plan of each UAV in a given iteration (6, 12, 18 and 25) is shown. The flight plans obtained in the same iteration are represented with the same line style. Note that this algorithm leads to shorter flight plans as the evolution goes on.

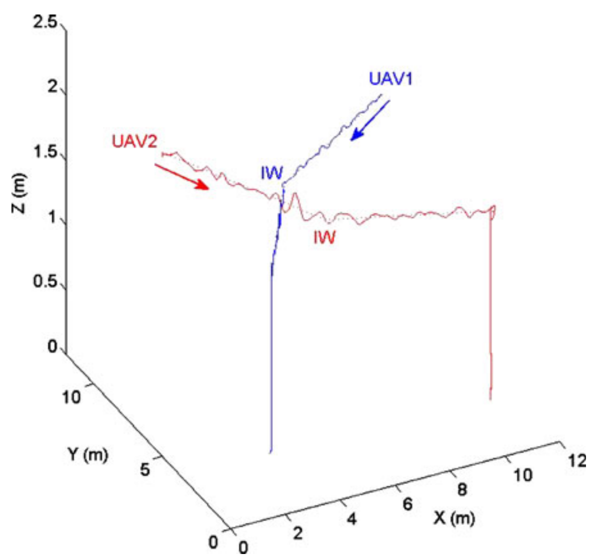



**Fig. 14** Trajectories computed by the CDR method for each aerial vehicle in the experiment I. Simulated trajectory (in *dotted line*) and actual trajectory (in *solid line*)

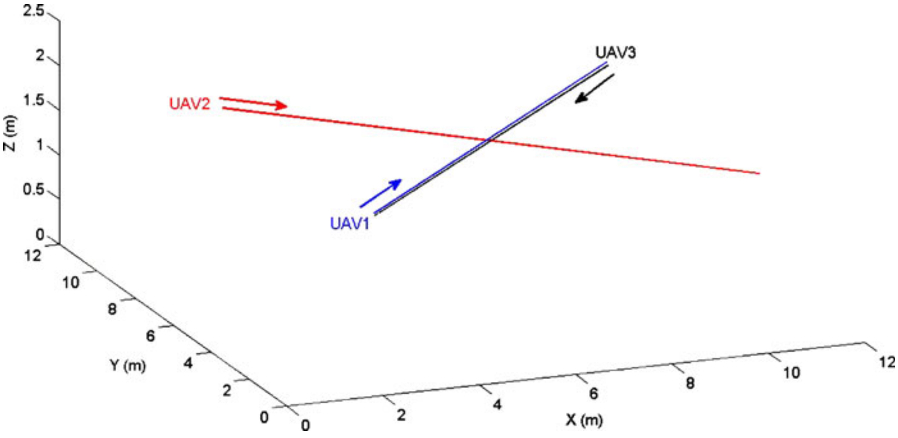



**Fig. 15** Initial trajectories in the experiment II. The height is the same for three aerial vehicles: z = 1.5 m

4.3 Experiments

Several experiments with the Hummingbird quadrotors (see Figs. 4 and 12) in the CATEC's indoor testbed described above have been also performed to validate the method. The quadrotors fly with constant speed, $v = 0.5$ m. Figure 13 shows the initial trajectories in Experiment I. Both quadrotors fly in the same flight level.

A possible collision is detected. The proposed method computes the changes of trajectories to avoid the conflict. Therefore, each aerial vehicle changes its initial trajectory to avoid the conflict in a cooperative way to minimize the total cost. Figure 14 shows the trajectories computed in the Experiment I with an intermediate waypoint (IW). The simulated trajectory can be compared with the real one.

Experiment II considers three aerial vehicles (see Fig. 15).

Several possible collisions are detected when the separation between aerial vehicles is less than the minimum horizontal separation, $S_{xy} = 1.5$ m. The proposed method computes new trajectories to solve the conflicts cooperatively (see Fig. 16) minimizing overall cost.

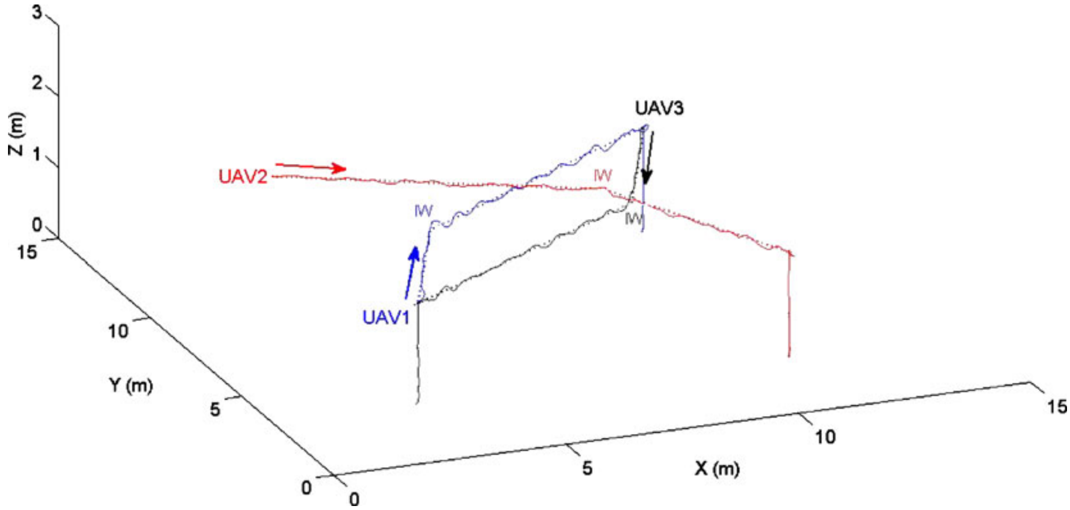



**Fig. 16** Trajectories computed by the CDR method for each aerial vehicle in the experiment II. Simulated trajectory (in *dotted line*) and actual trajectory (in *solid line*)

## 5 Conclusions and Future Extensions

In this paper, we have presented a CDR algorithm that solves possible trajectory conflicts in a common airspace by modifying the trajectories cooperatively to solve the conflicts with an overall minimum cost. The presented CDR method changes the initial flight plan of each UAV by adding intermediate waypoints to define the solution flight plan while maintain their velocities in order to avoid the detected conflict.

The proposed algorithms have been validated with many simulations and experiments performed in the CATEC multi-UAV aerial testbed. The main advantages of the algorithms are their low time of execution and scalability. In fact, the presented algorithm improves the solution as time passes, so it can be adapted to different applications that require different response times.

Future work considers the validation of these techniques with a larger number of aerial vehicles (up to 10) in the indoor multi-UAV aerial testbed. Moreover, we plan to perform experiments with tactical UAVs in a segregated airspace of 30 × 35 km.

On the other hand, new models will be introduced to handle different sources of uncertainty in sensors, aerial vehicle model, wind, etc. This uncertainty should be considered to predict the aerial vehicle trajectory in the conflict detection.

**Acknowledgements** Their work is partially supported by the European Commission ICT Programme under the Network of Excellence CONET NoE (FP7-INFSO-ICT-224053), the project PLANET (FP7 257649), the project of excellence of the Junta de Andalucía "Sistema para el Despliegue y Operación Autónoma de Redes de Sensores empleando Vehículos aéreos no Tripulados" (P09-TEP-5120), as well as the Spanish R&D project "ROBAIR: Safety and Reliability in Aerial Robotics" (DPI2008–03847). The authors would like to thank Mr. Diego Hinojosa Puro, Mr. Miguel Ángel Trujillo Soto and Mr. Allan Anderson Bell for their unselfish help during the development of the experiments at CATEC's indoor testbed.

## References

1. Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., Veiga, I., Gutierrez, L.G., Viegas, D.X., Gonzalez, M.A., Mallet, A., Alami, R., Chatila, R., Hommel, G., Colmenero, F.J., Arrue, B., Ferruz, J., Martinez de Dios, R., Caballero, F.: Multiple eyes in the skies. Architecture and perception issues in the COMETS multi-UAV project. IEEE Robot. Autom. Mag. **12**(1), 46–57 (2005)
2. Maza, I., Caballero, F., Martinez de Dios, J.R., Ollero, A.: A Distributed Architecture for a Robotic Platform with Aerial Sensor Transportation and Self-Deployment Capabilities. Journal of Field Robotics, Wiley Blackwell (2011)
3. Bayraktar, S., Fainekos, G.E., Pappas, J.: Experimental cooperative control of fixed-wing unmanned aerial vehicles. In: 43rd IEEE Conference on Decision and Control, pp. 4292–4298, Bahamas (2004)
4. Cobano, J.A., Martínez-de Dios, J.R., Conde, R., Sánchez-Matamoros, J.M., Ollero, A.: Data retrieving from heterogeneous wireless sensor network nodes using UAVs. J. Intell. Robot. Syst. **60**(1), 133–151 (2010)
5. Kuchar, J., Yang, L.: A review of conflict detection and resolution modeling methods. IEEE Trans. Intell. Transp. Syst. **1**(4), 179–189 (2000)
6. Alejo, D., Conde, R., Cobano, J.A., Ollero, A.: Multi-UAV collision avoidance with separation assurance under uncertainties. In: Proc. of the IEEE International Conference on Mechatronics (2009)
7. Vela, A., Solak, S., Singhose, W., Clarke, J.P.: A Mixed integer program for flight-level assignment and speed control for conflict resolution. In: Proceedings 48th IEEE Conference on Decision and Control, pp. 5219–5226 (2009)
8. Pallottino, L., Feron, E.M., Bicchi, A.: Conflict resolution problems for air traffic management systems solved with mixed integer programming. IEEE Trans. Intell. Transp. Syst. **3**(1), 3–11 (2002)
9. Hwang, I., Tomlin, C.J.: Protocol-based Conflict Resolution for Air Traffic Control. Department of Aeronautics and Astronautics Stanford University, Stanford, CA (USA), SUDAAR-762 (2002)
10. Carbone, C., Ciniglio, U., Corraro, F., Luongo, S.: A novel 3D geometric algorithm for aircraft autonomous collision avoidance. In: Proceedings of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA (2006)
11. Masci, P., Tedeschi, A.: Modeling and evaluation of a game-theory approach for airbone conflict resolution in Omnet++. In: Second International Conference on Dependability (2009)
12. Durand, N., Alliot, J.M.: Ant colony optimization for air traffic conflict resolution. In: Eight USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), Napa (California, USA) (2009)
13. Vivona, R., Karr, D., Roscoe, D.: Pattern based genetic algorithm for airborne conflict resolution. In: AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado (2006)
14. Lamont, G.B., Slear, J.N., Melendez, K.: UAV swarm mission planning and routing using multi-objective evolutionary algorithms. In: Proc. of the IEEE Symposium on Computational – Intelligence in Multicriteria Decision Making, pp. 10–20 (2007)

15. Reif, J., Sharir, M.: Motion planning in the presence of moving obstacles. J. ACM **41**(4), 764–790 (1994)
16. Szczerba, R.J.: Threat netting for real-time, intelligent route planners. In: Proc. of the IEEE Symposium Information, Decision and Control, pp. 377–382, Adelaide, Australia (1999)
17. Lymperopoulos, I., Lecchini, A., Glover, W., Maciejowski, J., Lygeros, J.: A Stochastic Hybrid Model for Air Traffic Management Processes. Technical Report CUED/F-INFENG/TR.572, University of Cambridge, England (2007)
18. BADA website: http://www.eurocontrol.int/eec/public/standard_page/proj_BADA.html
19. Alpha test: http://parasol.tamu.edu/groups/amatogroup/benchmarks/mp/
20. Wall, M.: GAlib: a C++ library of genetic algorithm components. Available at http://lancet.mit.edu/ga/ (1996)