


## Article

# Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph

Luciano Blasi <sup>1</sup>, Egidio D'Amato <sup>2,\*</sup>, Massimiliano Mattei <sup>1</sup> and Immacolata Notaro <sup>1</sup>

<sup>1</sup> Department of Engineering, University of Campania Luigi Vanvitelli, 81031 Aversa (CE), Italy; luciano.blasi@unicampania.it (L.B.); massimiliano.mattei@unicampania.it (M.M.); immacolata.notaro@unicampania.it (I.N.)

<sup>2</sup> Department of Science and Technology, University of Naples Parthenope, 80143 Naples, Italy

\* Correspondence: egidio.damato@uniparthenope.it

Received: 10 July 2020; Accepted: 10 August 2020; Published: 13 August 2020



**Abstract:** This paper deals with a novel procedure to generate optimum flight paths for multiple unmanned aircraft in the presence of obstacles and/or no-fly zones. A real-time collision avoidance algorithm solving the optimization problem as a minimum cost piecewise linear path search within the so-called Essential Visibility Graph (EVG) is first developed. Then, a re-planning procedure updating the EVG over a selected prediction time interval is proposed, accounting for the presence of multiple flying vehicles or movable obstacles. The use of Dubins curves allows obtaining smooth paths, compliant with flight mechanics constraints. In view of possible future applications in hybrid scenarios where both manned and unmanned aircraft share the airspace, visual flight rules compliant with International Civil Aviation Organization (ICAO) Annex II Right of Way were implemented. An extensive campaign of numerical simulations was carried out to test the effectiveness of the proposed technique by setting different operational scenarios of increasing complexity. Results show that the algorithm is always able to identify trajectories compliant with ICAO rules for avoiding collisions and assuring a minimum safety distance as well. Furthermore, the low computational burden suggests that the proposed procedure can be considered a promising approach for real-time applications.

**Keywords:** path planning; collision avoidance; visibility graph; UAV; Dubins curve

## 1. Introduction

In the last few years, improvements in technology have led to an increasing use of unmanned aerial vehicles, especially for the reduction of costs and human risks they can assure. Many researchers are currently focusing on the perspective of hybrid airspaces, in which both manned and unmanned aircraft share the environment and information, to prepare scenarios implementing effective and general purpose unmanned missions [1–5].

Unmanned Traffic Management (UTM) depends on the ability to ensure safe and efficient paths to every aircraft in the airspace [6]. In this context, the research on path planning becomes central, with the need for ever faster and more effective algorithms to be used in real-time applications.

Path planning has been studied using several approaches in the literature [7–9]. One of the most effective categories of path planning techniques is based on geometrical methods, which can be implemented in real time due to their low computational burden. The most famous approach is probably the one based on Dubins' car [10–14], in which planned paths are made up as a sequence of straight lines and circular arcs.

However, pure geometrical approaches can be hardly used in the presence of obstacles without any amendment. In some cases, they are used to smooth trajectories generated with other methods,

including graph based paths. Graphs are widely used in trajectory optimization thanks to their capability of discretizing and modeling the continuum by means of a certain number of nodes and arcs: probabilistic roadmaps [15,16], Voronoi diagrams [17–20], cell decomposition [21–25], and polynomial and spline parameterizations [26–28] are examples of methods based on graphs.

The Visibility Graph (VG) [29–31] can be a very effective approach in the case of polygonal obstacles. Several approaches based on VGs have been developed to reduce the computational burden, as the Reduced Visibility Graph (RVG) [32–34] and the Essential Visibility Graph (EVG) [34–36].

Other popular techniques in path planning make use of potential fields for both offline nominal trajectory optimization [37,38] and online collision avoidance [39–41]. More sophisticated approaches are specifically aircraft oriented, integrating flight dynamics constraints into the problem [42–44]. Trajectory optimization with constraints can be also based on nature inspired optimization techniques [45], but they are usually too burdensome to be implemented in real-time applications.

The available approaches for conflict resolution can be grouped into three different categories. Prescribed methods [46–48] are based on rules and protocols, and they are very effective in the case of integration in a preexisting airspace, but hardly capable of facing unexpected conditions. Optimization based methods [49–55] inherit several techniques from offline path planning and obstacle avoidance. Optimization makes these methods more versatile, but it increases the computational burden; for this reason, they are difficult to implement in real-time applications. Potential field based techniques [56–61] are based on the concept of considering aircraft as charged particles repelled by any other vehicle or obstacle. While the computational burden of these approaches are very low, sometimes they can result in non-feasible paths and possible deadlocks, limiting the capability to make optimal decisions independently. In the literature [62,63], some works exist that try to solve the problem of escaping from deadlocks and local minimums.

In order to optimize flight paths in real-time collision avoidance problems, in the presence of other aircraft or newly detected obstacles, this paper deals with the design of a reactive distributed guidance system. The proposed algorithm considers independent aircraft, sharing information only about position, speed, and direction within their communication/sensor range.

One of the original contributions of this work is the use of the Essential Visibility Graph (EVG). While the EVG was previously described in [34,36], in this paper, a complete proof of the optimality of the resulting path under some assumptions is presented. The EVG concept is extended in the presence of unknown or newly detected obstacles in both static and dynamic scenarios. Although the application of the velocity obstacles concept is shared with other papers [64–68], the applicability with obstacles of a general shape and the efficient use of the visibility graph concept coupled with Dubins' paradigm in finding the optimum flyable path place the proposed approach a further step ahead in the field of aircraft collision avoidance algorithms; finally, the further implementation of the visual flight rules, compliant with the International Civil Aviation Organization (ICAO) Annex II Right of Way rules, makes our proposed collision avoidance strategy suitable for future applications in hybrid scenarios where both manned and unmanned aircraft share the airspace.

The article is organized as follows: In Section 2, the geometrical approach based on Dubins' car is recalled, aimed at providing smooth trajectories compliant with the aircraft minimum turn radius constraint; in Section 3, a graph based approach based on the Essential Visibility Graph (EVG) is presented, with the proof of the optimality of the resulting path. The extension of such a method to dynamic scenarios with the implementation of the ICAO Right of Way rules is presented in Section 4. Finally, numerical results are shown and discussed in Section 5.

## 2. Dubins' Path Generation in the Absence of Obstacles

Consider two points  $A = (x_A; y_A)$  and  $B = (x_B; y_B)$ , namely the starting and target point, and two heading angles  $\psi_A$  and  $\psi_B$  used to define the current starting point and the desired final direction, respectively.

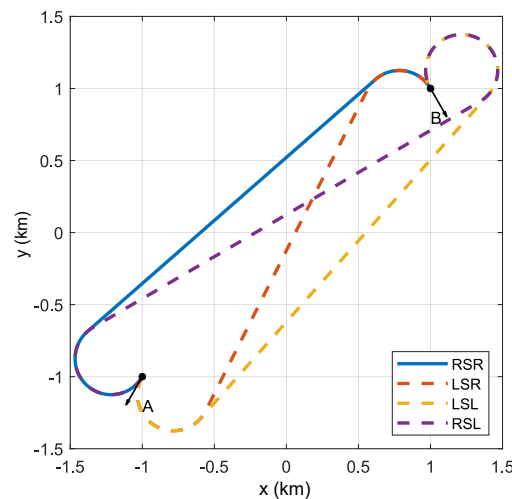
To obtain a path compliant with aircraft performance limits, connecting  $A$  with  $B$  with given heading angles, the trajectory must be smooth and continuous with bounds on curvature to account for the minimum flight turning radius  $R_{min}$ . The quality of the path is measured in terms of path length, in order to rate the fuel consumption.

In 2D unconstrained environments, Dubins' algorithm [10] is able to provide optimal and smooth paths, which are composed of straight lines and circular arcs only. Optimality, in the absence of obstacles, is guaranteed under the assumption that the distance between  $A$  and  $B$  is more than  $2R_{min}$ . Consequently the path is composed of three pieces: an arc of circumference with the radius equal to  $R_{min}$ , a straight line, and another circular arc with the same radius.

The optimal path is obtained with the following geometrical construction.

- Consider the circumferences,  $\mathcal{C}_{A_{Left}}$  and  $\mathcal{C}_{A_{Right}}$  ( $\mathcal{C}_{B_{Left}}$  and  $\mathcal{C}_{B_{Right}}$ ), with radius  $R_{min}$  to the left and right of the starting (final) point  $A$  ( $B$ ) and having at this point a tangent described by  $\psi_A$  ( $\psi_B$ ).
- For each pair of circumferences ( $\mathcal{C}_{A_{Right}}, \mathcal{C}_{B_{Right}}$ ), ( $\mathcal{C}_{A_{Right}}, \mathcal{C}_{B_{Left}}$ ), ( $\mathcal{C}_{A_{Left}}, \mathcal{C}_{B_{Left}}$ ), and ( $\mathcal{C}_{A_{Left}}, \mathcal{C}_{B_{Right}}$ ), four common tangents can be used to connect one circumference to the other, but only one is compatible with the starting direction  $\psi_A$  and the final direction  $\psi_B$ .
- Along the above tangents compatible with  $\psi_A$  and  $\psi_B$ , four feasible paths can be computed, namely RSR (Right Straight Right) for ( $\mathcal{C}_{A_{Right}}, \mathcal{C}_{B_{Right}}$ ), RSL (Right Straight Left) for ( $\mathcal{C}_{A_{Right}}, \mathcal{C}_{B_{Left}}$ ), LSL (Left Straight left) for ( $\mathcal{C}_{A_{Left}}, \mathcal{C}_{B_{Left}}$ ), and LSR (Left Straight Right) for ( $\mathcal{C}_{A_{Left}}, \mathcal{C}_{B_{Right}}$ ).
- The optimal path is the shortest among RSR, RSL, LSL, and LSR.

Figure 1 provides an example of how Dubins' concept is applied. Assuming  $A = (-1 \text{ km}; -1 \text{ km})$  and  $B = (1 \text{ km}; 1 \text{ km})$  with  $\psi_A = -2\pi/3$  rad and  $\psi_B = -\pi/3$  rad and a turning radius  $R_{min} = 250 \text{ m}$  it shows the four feasible paths, namely RSR, RSL, LSL, and LSR. In this case, the shortest one is the RSR path.



**Figure 1.** 2D Dubins paths from the point  $A = (-1 \text{ km}; -1 \text{ km})$  to  $B = (1 \text{ km}; 1 \text{ km})$  with initial heading angle  $\psi_A = -2\pi/3$  rad, final heading angle  $\psi_B = -\pi/3$  rad, and turning radius  $R_{min} = 250 \text{ m}$ .

### 3. Obstacle Avoidance Using EVG

Consider the path planning problem between  $A$  and  $B$  described in the previous section. In the presence of obstacles, it is possible to convert the trajectory planning problem into a minimum cost path search within a so-called Essential Visibility Graph (EVG) as presented in [35], under the following main assumptions:

**Hypothesis 1 (H1).** *Obstacles can be approximated by polygons.*

**Hypothesis 2 (H2).** Minimum turn radius  $R_{\min}$  is much shorter than both the distance between any two obstacle vertices and the distance between any obstacle vertex and the A/B points.

**Hypothesis 3 (H3).** Desired starting and ending heading  $\psi_A$  and  $\psi_B$  can be locally achieved by using Dubins' circles with a negligible increase of the total path length.

The flight path optimization problem is then formulated as follows.

**Problem 1.** Given starting and target points and directions, say A and B, and  $\psi_A$  and  $\psi_B$ , respectively, and  $N_o$  polygonal obstacles, find the shortest piecewise linear path connecting A to B.

Let us denote with  $\mathcal{I}_l$  the set of vertices of obstacle  $\mathfrak{P}_l$  ( $l = 1, \dots, N_o$ ) and build a graph whose node set is  $\Omega = \{A, B, \bigcup_{l=1}^{N_o} \mathcal{I}_l\}$ .

**Definition 1.** The set of visible vertices from a node  $V_i$  is defined as:

$$\mathcal{C}_{V_i} = \{V_j \in \Omega : V_i \rightarrow V_j\} \quad (1)$$

where  $V_i \rightarrow V_j$  means that there exists a segment  $r_{V_j, V_i}$  connecting  $V_i$  to  $V_j$ , which does not intersect any obstacle.

**Definition 2.** The set of obstacles reachable from  $V_i$  is:

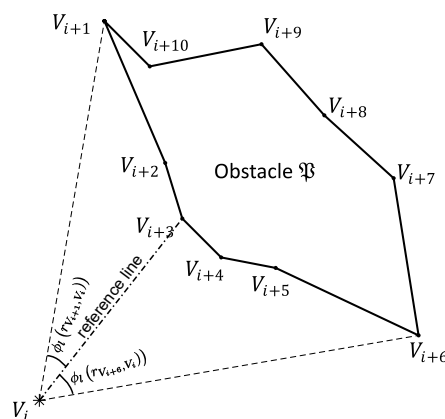
$$\mathcal{O}_{V_i} = \{\mathfrak{P}_l, l \in \{1, \dots, N_o\} : \mathcal{M}_{\mathfrak{P}_l, V_i} \neq \{0\}\} \quad (2)$$

where  $\mathcal{M}_{\mathfrak{P}_l, V_i} = \mathcal{I}_l \cap \mathcal{C}_{V_i}$  is the set of the  $l$ -th obstacle vertices reachable from  $V_i$ .

**Definition 3.** The set of transition nodes through the obstacle  $\mathfrak{P}_l$  boundary, from a point  $V_i$ , is:

$$\mathcal{T}_{\mathfrak{P}_l, V_i} = \left\{ \underset{V_j \in \mathcal{I}_l}{\operatorname{argmin}} \mathcal{E}_l(r_{V_j, V_i}); \underset{V_j \in \mathcal{I}_l}{\operatorname{argmax}} \mathcal{E}_l(r_{V_j, V_i}) \right\} \cap \mathcal{M}_{\mathfrak{P}_l, V_i} \quad (3)$$

$\mathcal{E}_l(r_{V_j, V_i})$  being the phase angle of vector  $r_{V_j, V_i}$  assuming a reference null angle in correspondence with any obstacle node belonging to  $\mathcal{M}_{\mathfrak{P}_l, V_i}$  (see Figure 2).

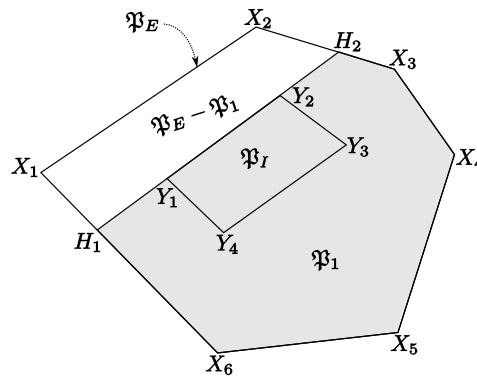


**Figure 2.** The set of nodes of the  $l$ -th obstacle reachable from  $V_i$  is  $\mathcal{M}_{\mathfrak{P}_l, V_i} = \{V_{i+1}, V_{i+2}, V_{i+3}, V_{i+4}, V_{i+5}, V_{i+6}\}$ , and the set of transition nodes is  $\mathcal{T}_{\mathfrak{P}_l, V_i} = \{V_{i+1}, V_{i+6}\}$ .

**Lemma 1.** Given two polygons  $\mathfrak{P}_E$  and  $\mathfrak{P}_I$ , with  $\mathfrak{P}_I$  convex and  $\mathfrak{P}_I \subset \mathfrak{P}_E$ , then  $\Lambda(\mathfrak{P}_I) < \Lambda(\mathfrak{P}_E)$ ,  $\Lambda(\mathfrak{P}_k)$  being the perimeter length of polygon  $\mathfrak{P}_k$ .

**Proof.** Assume that  $\mathfrak{P}_E$  and  $\mathfrak{P}_I$  have  $m$  and  $n$  sides respectively and vertices  $X_i$  (with  $i = 1, 2, \dots, m$ ) and  $Y_j$  (with  $j = 1, 2, \dots, n$ ) respectively. By prolonging in sequence each side of  $\mathfrak{P}_I$ , we can build a sequence of  $n$  polygons  $\mathfrak{P}_1, \mathfrak{P}_2, \dots, \mathfrak{P}_{n-1}, \mathfrak{P}_n$  with  $\mathfrak{P}_I \equiv \mathfrak{P}_n \subseteq \mathfrak{P}_{n-1} \subseteq \dots \subseteq \mathfrak{P}_2 \subseteq \mathfrak{P}_1 \subset \mathfrak{P}_E$  characterized by a decreasing perimeter length,  $\mathfrak{P}_k$  having  $k$  sides in common with  $\mathfrak{P}_I$ .

If  $\mathfrak{P}_I \subset \mathfrak{P}_E$ ,  $\mathfrak{P}_I$  has at least one side that is not coincident with any side of  $\mathfrak{P}_E$ . Without loss of generality, we start prolonging this side, to intersect two sides of  $\mathfrak{P}_E$ . With reference to Figure 3, let us consider the prolongation of sides  $Y_1Y_2$  that intersect the sides  $X_1X_6$  and  $X_2X_3$  of  $\mathfrak{P}_E$  at points  $H_1$  and  $H_2$ , respectively. Polygon  $\mathfrak{P}_1$ , defined by vertices  $\{H_1H_2X_3X_4X_5X_6\}$  in a clockwise sense, is such that  $\mathfrak{P}_I \subseteq \mathfrak{P}_1 \subset \mathfrak{P}_E$ .



**Figure 3.** Proof of Lemma 1: the definition of polygons  $\mathfrak{P}_E$ ,  $\mathfrak{P}_I$ , and  $\mathfrak{P}_1$ .

Since  $d_{H_1H_2} < d_{H_1X_1} + d_{X_1X_2} + d_{X_2H_2}$  ( $d_{X_jX_k}$  is the length of segment  $\overline{X_jX_k}$ ), then  $\Lambda(\mathfrak{P}_1) < \Lambda(\mathfrak{P}_E)$ . With the same procedure, we can generate polygon  $\mathfrak{P}_{k+1} \subseteq \mathfrak{P}_k$  prolonging the other sides of  $\mathfrak{P}_I$  to intersect the sides of  $\mathfrak{P}_k$  and demonstrate that  $\Lambda(\mathfrak{P}_k) \geq \Lambda(\mathfrak{P}_{k+1})$ .

Therefore, we can build a sequence of  $n$  polygons such that  $\Lambda(\mathfrak{P}_1) \geq \Lambda(\mathfrak{P}_2) \geq \dots \geq \Lambda(\mathfrak{P}_{n-1}) \geq \Lambda(\mathfrak{P}_n)$  with  $\mathfrak{P}_I \equiv \mathfrak{P}_n$ .  $\square$

**Theorem 1.** Shortest path connecting two points in the presence of one obstacle: Given two points  $A$  and  $B$ ,  $B$  being not reachable from  $A$ , and one single obstacle  $\mathfrak{P}_I$ , whose vertices are also nodes of the graph (set of nodes  $\mathcal{I}_I$ ), then the optimal path connecting node  $\bar{V}_0 = A$  to  $\bar{V}_{n+1} = B$  is a piecewise linear path defined by the sequence of  $n + 2$  nodes  $\bar{V}_0, \bar{V}_1, \dots, \bar{V}_{n+1}$  with  $\bar{V}_i \in \mathcal{T}_{\mathfrak{P}_I, V_{i-1}} \cap \mathcal{T}_{\mathfrak{P}_I, V_{i+1}}, \forall i = 1, \dots, n$ .

**Proof.** Let us prove by contradiction, assuming that there exists an optimal piecewise linear path defined by the sequence of nodes  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i^*, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$  with  $\bar{V}_i^* \notin \mathcal{T}_{\mathfrak{P}_I, V_{i-1}} \cap \mathcal{T}_{\mathfrak{P}_I, V_{i+1}}$ .

Case 1.  $\bar{V}_i^* \notin \mathcal{I}_I$ .

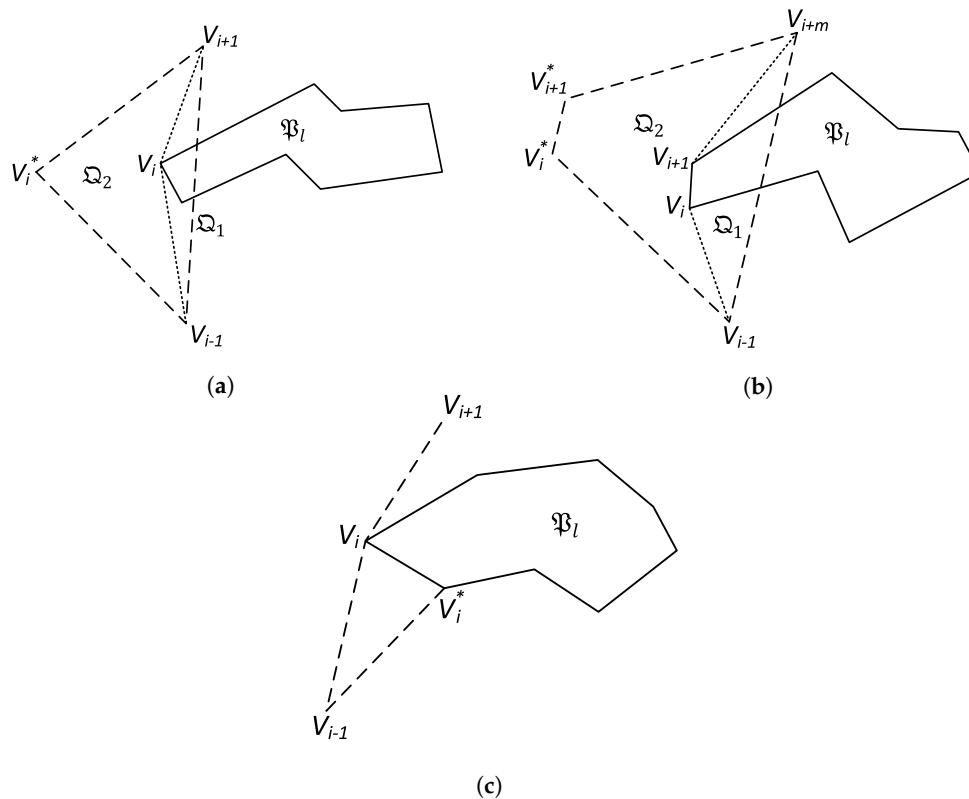
Case 1a.  $\bar{V}_i^*$  is reachable from nodes  $\bar{V}_{i-1} \notin \mathcal{I}_I$  and  $\bar{V}_{i+1} \notin \mathcal{I}_I$  (Figure 4a).

Let us define triangular polygon  $\Omega_1$  with vertices  $\{\bar{V}_{i-1}, \bar{V}_i, \bar{V}_{i+1}\}$  and triangular polygon  $\Omega_2$  with vertices  $\{\bar{V}_{i-1}, \bar{V}_i^*, \bar{V}_{i+1}\}$ . Segment  $\bar{V}_{i-1}\bar{V}_{i+1}$  necessarily cuts the obstacle, otherwise the optimal sequence of nodes would have been  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_{i+1}, \dots, \bar{V}_n, \bar{V}_{n+1}$ , then  $\Omega_1 \subset \Omega_2$ , and by virtue of Lemma 1,  $\Lambda(\Omega_1) < \Lambda(\Omega_2)$ . Since  $\Omega_1$  and  $\Omega_2$  share one side, we have:

$$d_{\bar{V}_{i-1}, \bar{V}_i} + d_{\bar{V}_i, \bar{V}_{i+1}} \leq d_{\bar{V}_{i-1}, \bar{V}_i^*} + d_{\bar{V}_i^*, \bar{V}_{i+1}} \quad (4)$$

which implies that the length of the piecewise linear path through the sequence of nodes  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i^*, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$  is not shorter than the length of the piecewise linear path through nodes  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$ , contradicting the optimality of the first sequence.

Case 1b.  $\bar{V}_i^*, \bar{V}_{i+1}^*, \dots, \bar{V}_{i+p}^*$  is a sequence of nodes  $\notin \mathcal{I}_l$  replacing  $\bar{V}_i, \bar{V}_{i+1}, \dots, \bar{V}_{i+m-1} \in \mathcal{I}_l, \bar{V}_{i-1}, \bar{V}_{i+m} \notin \mathcal{I}_l$  (Figure 4b).



**Figure 4.** Theorem 1. (a) Case 1a:  $\bar{V}_i^*$  reachable from  $\bar{V}_{i-1} \notin \mathcal{I}_l$  and  $\bar{V}_{i+1} \notin \mathcal{I}_l$ . (b) Case 1b:  $\bar{V}_i^*, \bar{V}_{i+1}^*, \dots, \bar{V}_{i+p}^* \notin \mathcal{I}_l$ . (c) Case 2:  $\bar{V}_i^* \in \mathcal{I}_l$ .

Let us define polygon  $\Omega_1$  with vertices  $\{\bar{V}_{i-1}, \bar{V}_i, \dots, \bar{V}_{i+m-1}, \bar{V}_{i+m}\}$  and polygon  $\Omega_2$  with vertices  $\{\bar{V}_{i-1}, \bar{V}_i^*, \dots, \bar{V}_{i+p}^*, \bar{V}_{i+m}\}$ . As for Case 1a,  $\Omega_1 \subset \Omega_2$ , and by virtue of Lemma 1,  $\Lambda(\Omega_1) < \Lambda(\Omega_2)$ . We have:

$$\begin{aligned} d_{\bar{V}_{i-1}, \bar{V}_i} + d_{\bar{V}_i, \bar{V}_{i+1}} + \dots \\ + d_{\bar{V}_{i+m-1}, \bar{V}_{i+m}} < d_{\bar{V}_{i-1}, \bar{V}_i^*} + \\ + d_{\bar{V}_i^*, \bar{V}_{i+1}^*} + \dots + d_{\bar{V}_{i+p}^*, \bar{V}_{i+m}} \end{aligned} \quad (5)$$

contradicting the optimality of path  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i^*, \dots, \bar{V}_{i+p}^*, \bar{V}_{i+m}, \dots, V_n$ .

Case 2.  $\bar{V}_i^* \in \mathcal{I}_l$  (Figure 4c).

Assume that  $\bar{V}_i^* \in \mathcal{T}_{\mathcal{P}_l, \bar{V}_{i-1}}$  and  $\bar{V}_i^* \notin \mathcal{T}_{\mathcal{P}_l, \bar{V}_{i+1}}$  (or vice versa). Without loss of generality, assume that  $\bar{V}_i^*$  lies on the same side that  $\bar{V}_i$  belongs to and that  $\bar{V}_i$  is reachable from  $\bar{V}_i^*$ . Since  $\bar{V}_{i+1}$  is not reachable from  $\bar{V}_i^*$ , at least one additional node belonging to  $\mathcal{M}_{\mathcal{P}_l, \bar{V}_{i+1}} \cap \mathcal{M}_{\mathcal{P}_l, \bar{V}_i^*}$  must be added to the sequence  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i^*, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$ . This is by construction vertex  $\bar{V}_i$ . It turns out that the new sequence of nodes  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i^*, \bar{V}_i, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$  defines a trajectory that is longer than that one identified by  $\bar{V}_0, \dots, \bar{V}_{i-1}, \bar{V}_i, \bar{V}_{i+1}, \dots, \bar{V}_{n+1}$ .  $\square$

The following two straightforward corollaries of Theorem 1 hold.

**Corollary 1.** Given an optimal sequence of nodes  $\bar{V}_0, \dots, \bar{V}_{n+1}$ , with  $(\bar{V}_i \in \mathcal{I}_l, i = 1, \dots, n)$ , then this sequence must be a set of clockwise or counterclockwise ordered and adjacent vertices of the  $l$ -th obstacle's convex hull.

**Corollary 2.** Shortest path connecting two points in the presence of  $N_o$  obstacles: given two points  $A$  and  $B$ ,  $B$  being not reachable by  $A$ , and  $N_o$  obstacles, the optimal piecewise linear trajectory connecting  $A = \bar{V}_0$  to  $B = \bar{V}_{n+1}$  is defined by the sequence of  $n + 2$  nodes  $\bar{V}_0, \dots, \bar{V}_{n+1}$ , where each intermediate node  $\bar{V}_i$ , with  $i = 1, \dots, n$ , belongs to  $\mathcal{T}_{\mathfrak{P}_i, \bar{V}_{i-1}} \cap \mathcal{T}_{\mathfrak{P}_i, \bar{V}_{i+1}}$  with obstacle  $\mathfrak{P}_i \in \mathcal{O}_{\bar{V}_{i-1}} \cap \mathcal{O}_{\bar{V}_{i+1}}$ .

The Essential Visibility Graph (EVG)  $\mathcal{G} = \{\mathcal{W}, \mathcal{E}\}$  can be then computed using a branching algorithm reported in the form of pseudo-code in Algorithm 1.

---

**Algorithm 1** EVG building procedure.

---

```

1 Data: Starting point  $A$ , target point  $B$ , obstacles' vertices  $\mathcal{I}_1, \dots, \mathcal{I}_{N_o}$ 
2 Result: EVG  $\mathcal{G} = \{\mathcal{W}, \mathcal{E}\}$ 
3 Add  $A$  and  $B$  to  $\mathcal{W}$ ;
4  $n = 1$ ;
5  $\mathcal{K}_n = \{A\}$ ;
6 while  $\mathcal{K}_n \neq \emptyset$  do
7    $n = n + 1$ ;
8    $\mathcal{K}_n = \emptyset$ ;
9   foreach  $X \in \mathcal{K}_{n-1}$  do
10    if  $B$  is visible from  $X$  then
11      add arc  $(X, B)$  to  $\mathcal{E}$ ;
12    else
13      compute the set of obstacle reachable from  $X$ ,  $\mathcal{O}_X$ ;
14      foreach  $\mathfrak{P} \in \mathcal{O}_X$  do
15        compute the set of transition nodes  $\mathcal{T}_{\mathfrak{P}, X}$ ;
16        foreach  $Y \in \mathcal{T}_{\mathfrak{P}, X}$  do
17          if  $Y \notin \mathcal{W}$  then
18            add  $Y$  to  $\mathcal{W}$ ;
19            add  $(Y, X)$  to  $\mathcal{E}$ ;
20            add  $Y$  to  $\mathcal{K}_n$ ;
21          end
22        end
23      end
24    end
25  end
26 end
27 end

```

---

The shortest path connecting two points in the presence of polygonal obstacles can be then calculated as a minimum cost search [69] over the EVG.

Once a piecewise linear optimal flight path has been found, an optimal flyable path, i.e., compatible with the flight mechanics constraints like minimum turn radius, can be computed using Dubins' circles, as summarized in the following Procedure 1.

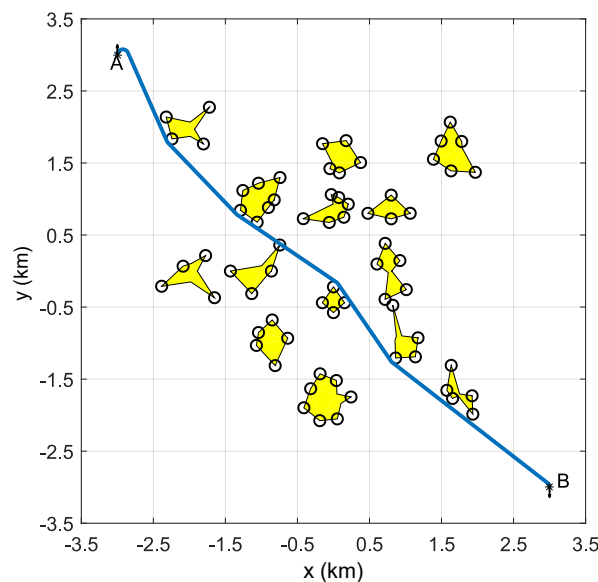
**Procedure 1.** Find a (sub-)optimal flyable path

- Step 1. Find an optimal piecewise linear path over the EVG described by  $n + 2$  nodes including  $A$  and  $B$ , called  $\bar{V} = \{\bar{V}_0, \dots, \bar{V}_{n+1}\}$ .
- Step 2. For each intermediate node  $h = 2, \dots, n$  of the optimal sequence, define a circle  $\mathcal{C}_h$ , centered in the corresponding polygon vertex.
- Step 3. Define Dubins' circles  $\mathcal{C}_{A\text{Right}}$  and  $\mathcal{C}_{A\text{Left}}$ ,  $\mathcal{C}_{B\text{Right}}$  and  $\mathcal{C}_{B\text{Left}}$ , for the starting and target points  $A$  and  $B$ .
- Step 4. Build 2D Dubins trajectories based on four possible sequences of circles, namely:

1.  $(\mathcal{C}_{A\text{Right}}, \mathcal{C}_2, \dots, \mathcal{C}_h, \dots, \mathcal{C}_{n+1}, \mathcal{C}_{B\text{Left}})$ ,
2.  $(\mathcal{C}_{A\text{Left}}, \mathcal{C}_2, \dots, \mathcal{C}_h, \dots, \mathcal{C}_{n+1}, \mathcal{C}_{B\text{Left}})$ ,
3.  $(\mathcal{C}_{A\text{Left}}, \mathcal{C}_2, \dots, \mathcal{C}_h, \dots, \mathcal{C}_{n+1}, \mathcal{C}_{B\text{Right}})$ ,
4.  $(\mathcal{C}_{A\text{Right}}, \mathcal{C}_2, \dots, \mathcal{C}_h, \dots, \mathcal{C}_{n+1}, \mathcal{C}_{B\text{Right}})$ ,



As a numerical example, Figure 5 shows the optimal flyable path from point  $A = (-3 \text{ km}; 3 \text{ km})$  to  $B = (3 \text{ km}; -3 \text{ km})$  with initial heading angle  $\psi_A = \pi/2$  rad, final heading angle  $\psi_B = -\pi/2$  rad, and turning radius  $R_{min} = 80$  m, in the presence of  $N_o = 14$  polygonal obstacles. The EVG is composed of 486 arcs and 76 vertices, whereas the standard visibility graph has 1322 arcs and 100 vertices. As expected, the construction of the weighted EVG provides an improvement with respect to the visibility graph in terms of the number of nodes and arcs, leading to a lower computational burden.



**Figure 5.** Optimal path between  $A = (-3 \text{ km}; 3 \text{ km})$  and  $B = (3 \text{ km}; -3 \text{ km})$  with initial heading angle  $\psi_A = \pi/2$  rad, final heading angle  $\psi_B = -\pi/2$  rad, and turn radius  $R_{min} = 80$  m.

Algorithm 1, combined with Procedure 1, allows computing a starting nominal path, called  $\Gamma_0$ . However, during the flight, if one or more unknown obstacles are detected, the vehicle must change its trajectory to avoid possible collisions. The new flight path can be computed online in an efficient way by updating the nominal EVG to include the newly detected obstacles. The updated EVG can be used to find the shortest path between the current position of the aircraft, becoming also a new additional node of the graph, and the target point.

If the current position of the aircraft, say  $P^*$ , belongs to the nominal path  $\Gamma_0$  and no new obstacles are detected, the optimal flyable path between the starting point  $P^*$  and the target point  $B$  is a sub-path of  $\Gamma_0$  according to the optimality principle.

If  $N_n$  new obstacles are detected, let be  $\mathcal{I}_l$  the subset of nodes being vertices of the new  $l$ -th obstacle, with  $l = N_o + 1, \dots, N_o + N_n$ . Algorithm 2 is run to update the nominal EVG with a lower computational cost compared to the cost required to compute the nominal starting EVG.



**Algorithm 2** EVG updating procedure.

---

```

1 Data: Starting point  $P^*$ , new obstacles' vertices  $\mathcal{I}_{N_0+1}, \dots, \mathcal{I}_{N_0+N_n}$ , nominal EVG  $G = \{\mathcal{W}, \mathcal{E}\}$ 
2 Result: Updated EVG graph  $G_{new} = \{\mathcal{W}_{new}, \mathcal{E}_{new}\}$ 
3 Add  $P^*$  to  $\mathcal{W}$ ;
4  $\mathcal{W}_{new} = \mathcal{W}; \mathcal{E}_{new} = \mathcal{E};$ 
5  $n = 1;$ 
6 if  $B$  is not visible from  $P^*$  then
7    $\mathcal{K}_n = \{P^*\};$ 
8   foreach arc  $(X, Y)$  in  $\mathcal{E}$  do
9     if  $Y$  is not reachable from  $X$  then
10       remove arc  $(X, Y)$  from  $\mathcal{E}_{new}$ ;
11       add  $X$  and  $Y$  to  $\mathcal{K}_n$ ;
12     end
13   end
14   if  $B \in \mathcal{K}_n$  then
15     remove  $B$  from  $\mathcal{K}_n$ 
16   end
17   while  $\mathcal{K}_n \neq \emptyset$  do
18      $n = n + 1;$ 
19      $\mathcal{K}_n = \emptyset;$ 
20     foreach  $X \in \mathcal{K}_{n-1}$  do
21       if  $X \in \mathcal{I}_p$  with  $p = 1, \dots, N_0$  then
22         compute the set  $\mathcal{O}_X$  of new obstacle  $\mathfrak{P}_l$  (with  $l = N_0 + 1, \dots, N_n$ ) reachable from  $X$ ;
23         foreach  $\mathfrak{P} \in \mathcal{O}_X$  do
24           compute the set of transition nodes  $\mathcal{T}_{\mathfrak{P}, X}$ ;
25           foreach  $Y \in \mathcal{T}_{\mathfrak{P}, X}$  do
26             if  $Y \notin \mathcal{W}_{new}$  then
27               add  $Y$  to  $\mathcal{W}_{new}$ ;
28               add  $Y$  to  $\mathcal{K}_n$ ;
29             end
30             add  $(Y, X)$  to  $\mathcal{E}_{new}$ ;
31           end
32         end
33       else
34         if  $B$  is visible from  $X$  then
35           add arc  $(X, B)$  to  $\mathcal{E}$ ;
36         else
37           compute the set of all obstacle  $l$  (with  $l = 1, \dots, N_n$ ) reachable from  $X$ ,  $\mathcal{O}_X$ ;
38           foreach  $\mathfrak{P} \in \mathcal{O}_X$  do
39             compute the set of transition nodes  $\mathcal{T}_{\mathfrak{P}, X}$ ;
40             foreach  $Y \in \mathcal{T}_{\mathfrak{P}, X}$  do
41               if  $Y \notin \mathcal{W}_{new}$  then
42                 add  $Y$  to  $\mathcal{W}_{new}$ ;
43                 add  $Y$  to  $\mathcal{K}_n$ ;
44               end
45               add  $(Y, X)$  to  $\mathcal{E}_{new}$ ;
46             end
47           end
48         end
49       end
50     end
51   end
52 end

```

---

**4. Collision Avoidance with EVG**

In a dynamic environment, the UAV must prevent collisions also with moving obstacles and other vehicles called intruders. The proposed collision avoidance algorithm consists of a path re-planning

procedure based on the EVG and Dubins' paths, to avoid UAVs that are on a collision course. The algorithm is computationally efficient as it makes use of the procedures previously described.

When an aircraft predicts multiple future collisions, it selects the nearest one in accordance with a typical reactive scheme [39], and it calls the EVG updating procedure assuming a fictitious additional obstacle centered at the predicted collision position. To avoid ambiguity (i.e., it is possible to overcome the obstacle on the left or on the right), the EVG is cut according to the Right of Way rules prescribed in the Annex II of International Civil Aviation Organization (ICAO).

#### 4.1. Collisions Prediction

In our work, we assume that UAVs know the position, speed, and heading of neighboring aircraft, within a given sensing range given by ADS-B, TCAS, or other sensing devices.

Consider the  $i$ -th controlled aircraft and the set  $\mathcal{S}_i$  of vehicles in the sensor range. Let  $P_i(t) = (x_i(t), y_i(t))$  and  $P_j(t) = (x_j(t), y_j(t)) \in \mathcal{S}_i$  be the position of the  $i$ -th and  $j$ -th UAV, respectively. The line of sight vector between aircraft  $i$  and  $j$  is defined as  $LOS_{i,j} = P_j - P_i$  (see Figure 6).

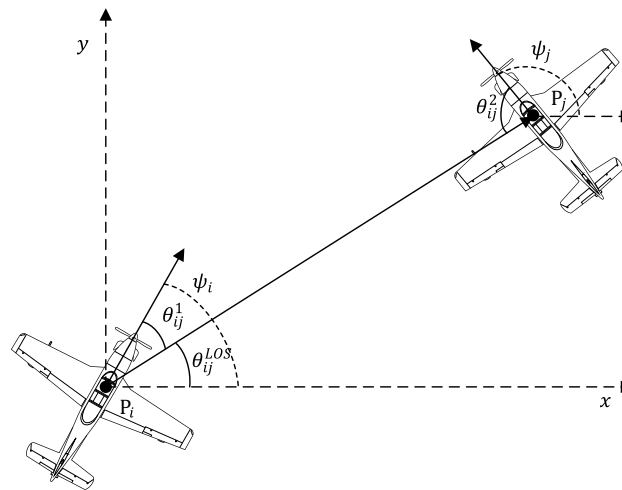


Figure 6. Line Of Sight (LOS) definition.

$\theta_{i,j}^1 \in [0, 2\pi[$  is the angle between the velocity vector of aircraft  $i$  and  $LOS_{i,j}$ , and  $\theta_{i,j}^2 \in [0, 2\pi[$  is the angle between the velocity vector of the vehicle located in  $P_j$  and  $LOS_{i,j}$  (see Figure 6).

The ICAO Right of Way rules can be formalized by defining an operator  $\Phi_{i,j}$ , such that  $\Phi_{i,j} = 1$ , if the  $j$ -th vehicles has the right of way, and  $\Phi_{i,j} = 0$  in other cases:

$$\begin{cases} \Phi_{i,j} = 1 & \forall \theta_{i,j}^2 \cap \frac{3}{2}\pi \leq \theta_{i,j}^1 < 2\pi \\ \Phi_{i,j} = 1 & 0 \leq \theta_{i,j}^2 \leq \frac{\pi}{2} \cap 0 \leq \theta_{i,j}^1 \leq \frac{\pi}{2} \\ \Phi_{i,j} = 1 & \pi \leq \theta_{i,j}^2 < 2\pi \cap 0 \leq \theta_{i,j}^1 \leq \frac{\pi}{2} \\ \Phi_{i,j} = 0 & \text{otherwise.} \end{cases} \quad (6)$$

At any time  $t$ , by evaluating the operator  $\Phi_{i,j}(t)$ ,  $\forall j \in \mathcal{S}_i(t)$ , it is possible to determine the list  $\overline{\mathcal{S}}_i(t)$  of vehicles taking priority in accordance with the ICAO rules:

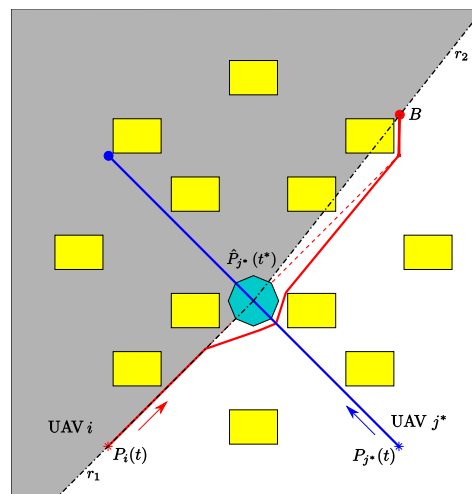
$$\overline{\mathcal{S}}_i(t) = \{j \in \mathcal{S}_i(t) : \Phi_{i,j}(t) = 1\} \quad (7)$$

In the prediction time horizon  $[t, t + T_d]$ , the  $i$ -th aircraft control system computes the predicted distance  $\hat{d}_{ij}(\tau|t)$ , with  $\tau \in [t, t + T_d]$  for each vehicle  $j \in \overline{\mathcal{S}}_i(t)$ , assuming that the intruders fly on a straight path at constant speed  $v_j$ . If  $\hat{d}_{ij}(t)$  is less than the minimum separation distance  $d_{min}$ , a collision

can occur, and vehicle  $j$  is added to the set of colliding aircraft, namely  $\mathcal{S}_i^* \subseteq \overline{\mathcal{S}}_i \subseteq \mathcal{S}_i$ . Assuming that  $t_{i,j}^*$  is the time of collision between  $i$  and  $j$ , predicted at time  $t$ , the  $i$ -th aircraft control system selects the nearest collision, i.e., aircraft  $j^*$  such that  $t_{i,j^*}^*$  is minimum, say  $t^*$  this time.

#### 4.2. Anti-Collision Flight Path Update

Consider that the  $i$ -th aircraft is following its nominal path  $\Gamma_0$ , and at time  $t$ , it predicts a collision with aircraft  $j^*$  at time  $t^*$ . Let  $\hat{P}_{j^*}(t^*)$  be the position of aircraft  $j^*$  at collision time  $t^*$ . Assume a fictitious obstacle of  $m$  vertices circumscribed to a circumference with the center at  $\hat{P}_{j^*}(t^*)$  and radius  $d_{min}$  (see Figure 7). If the obstacle overlaps with other existing obstacles in the scenario, consider the convex hull of the vertices of the overlapped obstacles plus the fictitious obstacle. The nominal EVG is then updated using Algorithm 2.



**Figure 7.** Anti-collision path update: All graph edges in the gray area are deleted in order to force UAV  $i$  to pass behind UAV  $j^*$ , i.e., on the right of the fictitious obstacle centered at  $\hat{P}_{j^*}(t^*)$ . Dashed lines indicate nominal paths; dashed-dotted lines indicate  $r_1$  and  $r_2$  straight lines; solid lines indicate the actual UAVs' trajectories.

The Right of Way rules prescribe that “An aircraft that is obliged by the following rules to keep out of the way of another shall avoid passing over, under or in front of the other, unless it passes well clear and takes into account the effect of aircraft wake turbulence” (Section 3.2.2.1 in [70]). For this reason, a cut of the EVG is needed to plan a path compliant with the rules.

Consider a straight line  $r_1$  passing through  $P_i(t)$  and  $\hat{P}_{j^*}(t^*)$ :

$$r_1: \quad \beta \cdot (x - x_i(t)) - \alpha \cdot (y - y_i(t)) = 0 \quad (8)$$

where  $\alpha$  and  $\beta$  are the components of the vector  $\hat{P}_{j^*}(t^*) - P_i(t) = [\alpha, \beta]^T = [\hat{x}_{j^*}(t^*) - x_i(t), \hat{y}_{j^*}(t^*) - y_i(t)]^T$ .

Consider a straight line  $r_2$  passing through  $\hat{P}_{j^*}(t^*)$  and the target point  $B = (x_B, y_B)$ :

$$r_2: \quad v \cdot (x - \hat{x}_{j^*}(t^*)) - \mu \cdot (y - \hat{y}_{j^*}(t^*)) = 0 \quad (9)$$

where  $\mu$  and  $v$  are the components of the vector  $B - \hat{P}_{j^*}(t^*) = [\mu, v]^T = [x_B - \hat{x}_{j^*}(t^*), y_B - \hat{y}_{j^*}(t^*)]^T$ .

To follow the Right of Way rules, the optimization algorithm over the EVG can choose only vertices in the semi-plane defined using  $r_1$  and  $r_2$  and the following conditions:

$$\begin{cases} -\sum_{i,j^*} [\beta \cdot (x - x_i(t)) - \alpha \cdot (y - y_i(t))] \leq 0 \\ -\sum_{i,j^*} [v \cdot (x - \hat{x}_{j^*}(t^*)) - \mu \cdot (y - \hat{y}_{j^*}(t^*))] \leq 0 \end{cases} \quad (10)$$

where the operator  $\Sigma_{i,j} = 1$  if the UAV  $i$  must turn to the left in accordance with Right Of Way rules, otherwise  $\Sigma_{i,j} = -1$  if  $i$  must turn to the right:

$$\begin{cases} \Sigma_{i,j} = 1 & 0 \leq \theta_{ij}^1 \leq \frac{\pi}{2} \quad \theta_{ij}^1 < \theta_{ij}^2 \leq \frac{\pi}{2} \\ \Sigma_{i,j} = 1 & \frac{3}{2}\pi < \theta_{ij}^1 < 2\pi \quad 0 \leq \theta_{ij}^2 \leq \theta_{ij}^1 - \pi \\ \Sigma_{i,j} = 1 & \frac{3}{2}\pi < \theta_{ij}^1 < 2\pi \quad \theta_{ij}^1 \leq \theta_{ij}^2 < 2\pi \\ \Sigma_{i,j} = -1 & \text{otherwise.} \end{cases} \quad (11)$$

Figure 7 shows the nominal path of two UAVs. UAV  $i$  predicts a collision with UAV  $j^*$  at time  $t^*$  in  $\hat{P}_{j^*}(t^*)$ . Since UAV  $i$  does not have the right of way, it must re-plan its trajectory. Before updating the graph, a fictitious obstacle is added to the environment, centered at the point  $\hat{P}_{j^*}(t^*)$  where the collision is expected. In order to force the UAV  $i$  to avoid the collision with the UAV  $j^*$ , passing on the right of fictitious obstacle, all arcs of the updated graph in the gray area are considered unfeasible and deleted from the graph.

## 5. Numerical Results

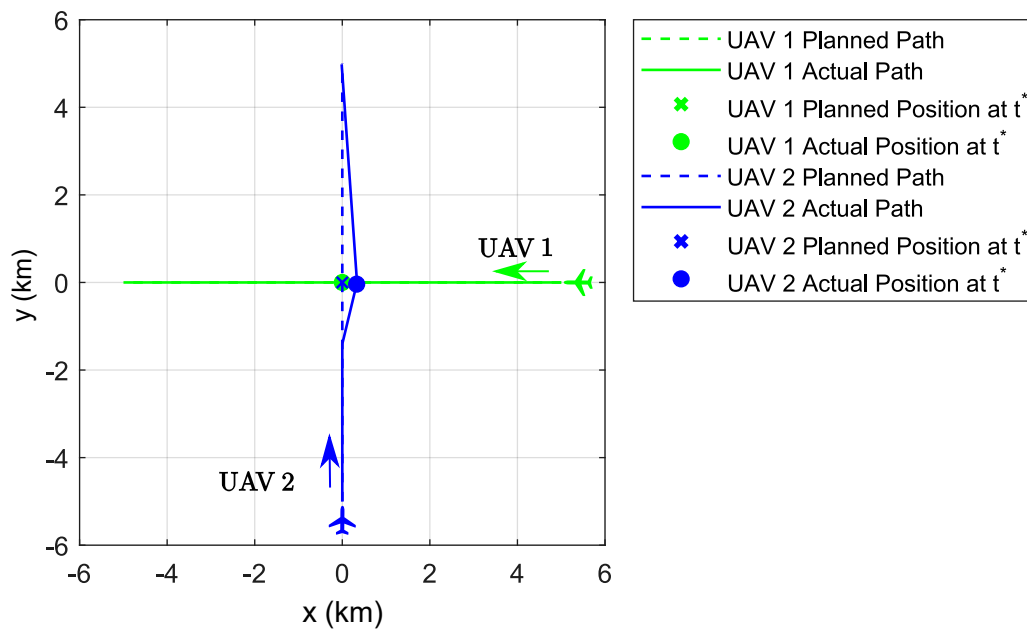
An extensive campaign of numerical simulations was carried out whose results are presented in this section. Several operational scenarios were set to verify the effectiveness of the proposed technique. The selected scenarios are among the most complex ones to suitably test our anti-collision algorithm. Both the planned paths and the actual trajectories, as modified by the collision avoidance system, are shown. Mutual distances between UAVs over time are also shown for scenarios with few aircraft. For the sake of clarity, by increasing the number of aircraft involved, only the minimum distances reached between UAVs are reported in tabular format. Finally, all the vehicles have the same cruise speed and minimum turn radius. Table 1 summarizes the simulation parameters used for Scenarios #1–#6.

**Table 1.** Simulation parameters of Scenarios #1–#6.

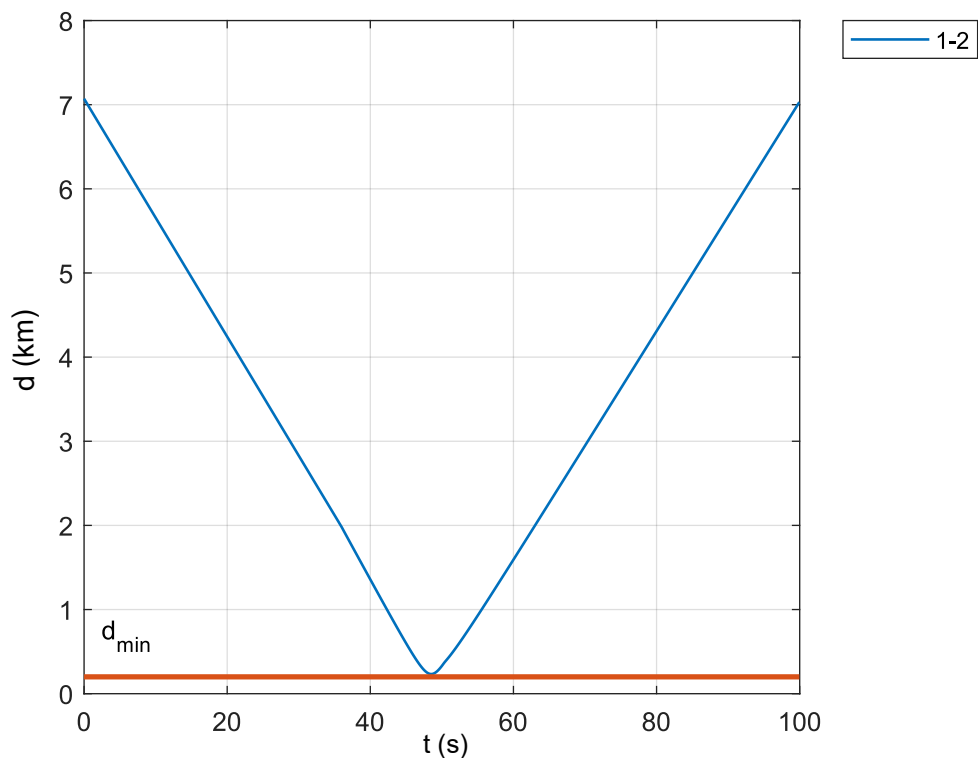
$d_{min}$	200 m
Turning Radius $R_{min}$	100 m
Cruise Speed $v_c$	10 m/s
Sample Time $\Delta t$	0.5 s
Prediction Time Horizon $T_d$	20 s

### 5.1. Scenario #1: Two Converging UAVs

In this scenario, two UAVs follow perpendicular flight paths intersecting at one point. When UAV 2 detects a possible collision at time  $t^*$ , in order to respect the right of way rules (see 3.2.2.3, in ICAO Annex II), it re-plans its path, giving the way to UAV 1, which does not change its planned trajectory. Figure 8 shows the planned paths (dotted lines) and the actual trajectories followed (solid lines) of both UAVs. As can be seen, UAV 2 passes on the right of the expected collision point, in order to pass behind UAV 1, according to the right of way rules. In Figure 9, the distance between the aircraft over time is reported. As we can see, the collision avoidance system is able to guarantee a mutual distance not below the prescribed minimum value of 200 m.



**Figure 8.** Scenario #1. Two converging UAVs. UAV 1's initial position  $\mathbf{A}_1(0) = (5 \text{ km}; 0 \text{ km})$ ,  $V_1(0) = V_c$ ,  $\psi_1(0) = \pi$  rad. UAV 2's initial position  $\mathbf{A}_2(0) = (0 \text{ km}; -5 \text{ km})$ ,  $V_2(0) = V_c$ ,  $\psi_2(0) = \pi/2$  rad.  $t^*$  represents the predicted collision time instant.

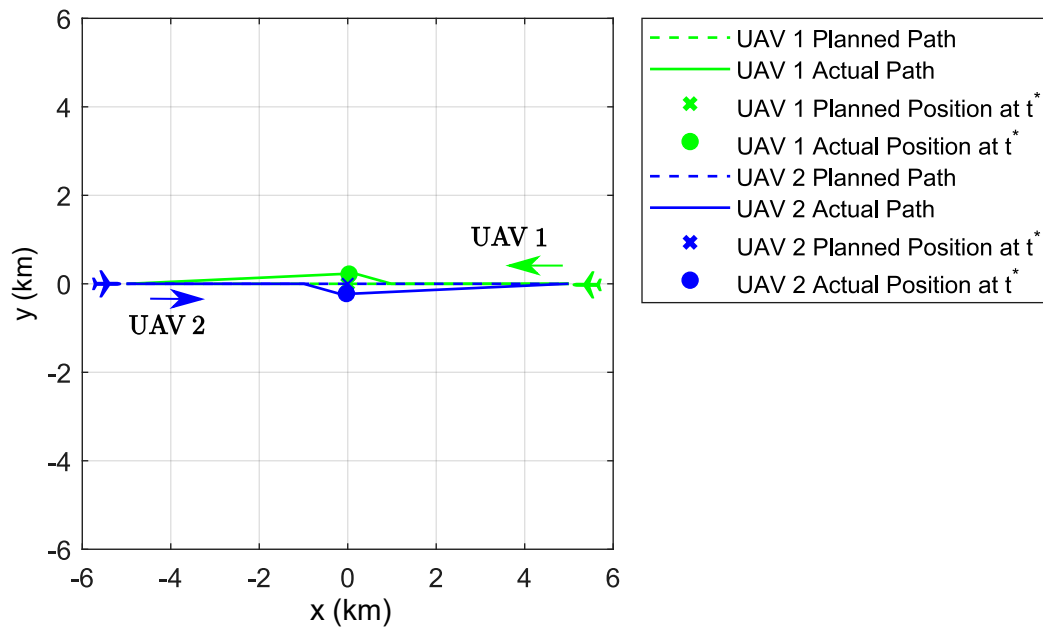


**Figure 9.** Scenario #1. Distance over time between UAV 1 and UAV 2.

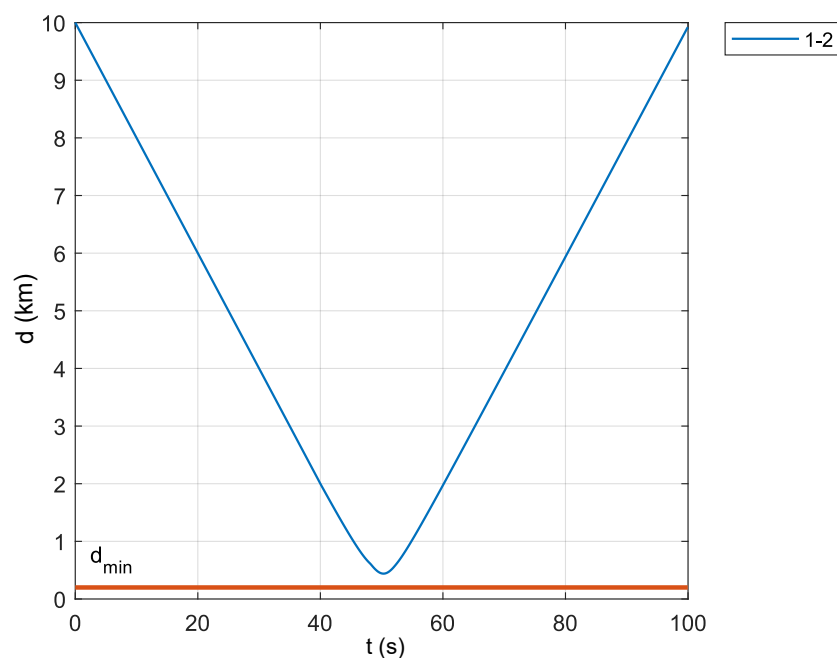
## 5.2. Scenario #2: Two Opposite UAVs

In this scenario, two aircraft are approaching from opposite directions following the same path. As prescribed by ICAO Rule 3.2.2.2, in Annex II [70], both UAVs must deviate, re-planning their trajectory in order to turn to the right. Figure 10 shows the vehicles' planned paths (dotted lines) and the actual trajectories followed (solid lines). As can be seen, both UAVs turn on the right near the

expected collision point according to the right of way rules. Figure 11 shows the distance between UAVs over time. As we can see, the collision avoidance system is able to guarantee a mutual distance not below the prescribed minimum value of 200 m.



**Figure 10.** Scenario #2. Two opposite UAVs. UAV 1's initial position  $\mathbf{A}_1(0) = (5 \text{ km}; 0 \text{ km})$ ,  $V_1(0) = V_c$ ,  $\psi_1 = \pi \text{ rad}$ . UAV 2's initial position  $\mathbf{A}_2(0) = (-5 \text{ km}; 0 \text{ km})$ ,  $V_2(0) = V_c$ ,  $\psi_2(0) = 0 \text{ rad}$ .  $t^*$  represents the predicted collision time instant.

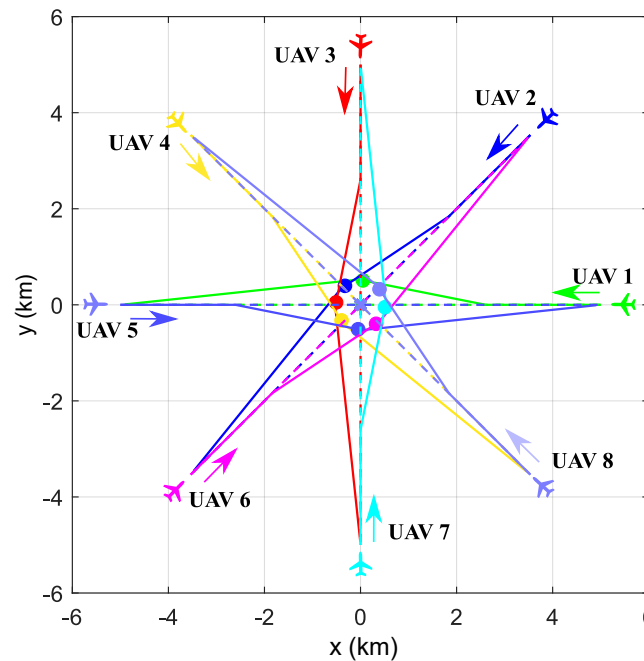


**Figure 11.** Scenario #2. Distance over time between UAV 1 and UAV 2.

### 5.3. Scenario #3: Eight Converging UAVs

In this scenario, eight aircraft are uniformly distributed in the airspace as a circular pattern and follow trajectories all converging towards the center, which is the potential collision point. Figure 12

shows the UAVs' planned paths (dotted lines) and the actual trajectories followed (solid lines). As can be seen, each aircraft avoids a collision by re-planning its own nominal path making a turn maneuver to the right near the center of the scenario and reaching the destination point on the opposite side. Circular markers indicate the UAVs' positions at the time the collision is predicted. Note that the aircraft are placed all around a sort of fictitious traffic circle with a radius that depends on the number of vehicles, centered at the point where the collision is expected to occur. As previously said, due to the higher number of aircraft involved in this scenario, only the minimum distances between UAVs are summarized in Table 2. As we can see, they are always greater than the required minimum value.



**Figure 12.** Scenario #3. Eight UAVs uniformly distributed as a circular pattern converging to the center of the airspace and reaching the destination points on the opposite side. UAVs' initial positions  $A_i = (5 \cos(\alpha_i); 5 \sin(\alpha_i))$  km,  $\alpha_i = (i - 1)(2\pi/8) \forall i = 1, \dots, 8$ ,  $V_i(0) = V_c$ ,  $\psi_i = \alpha_i + \pi$ . Dotted lines represent the UAVs' planned paths, whereas solid lines represent the actual trajectories. Circular markers indicate the UAVs' positions at the time the collision is predicted.

**Table 2.** Scenario #3: Minimum distances between UAVs (m).

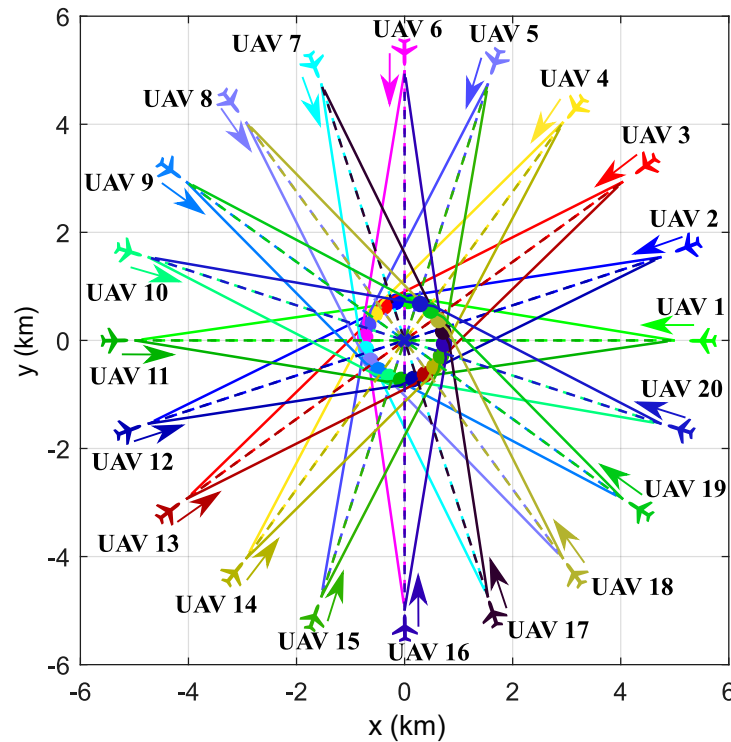
	1	2	3	4	5	6	7	8
1	-	389	719	939	1017	939	719	389
2		-	389	719	939	1017	939	719
3			-	389	719	939	1017	939
4				-	389	719	939	1017
5					-	389	719	939
6						-	389	719
7							-	389
8								-

#### 5.4. Scenario #4: Twenty Converging UAVs

In this scenario, twenty aircraft are uniformly distributed in the airspace as a circular pattern and follow trajectories all converging towards the center, which is the potential collision point. Figure 13 shows the UAVs' planned paths (dotted lines) and the actual trajectories followed (solid lines). Despite the even higher number of aircraft, also in this test case, the UAVs are able to avoid collisions. Once a potential collision point is detected, each aircraft re-plans its own path making a turn



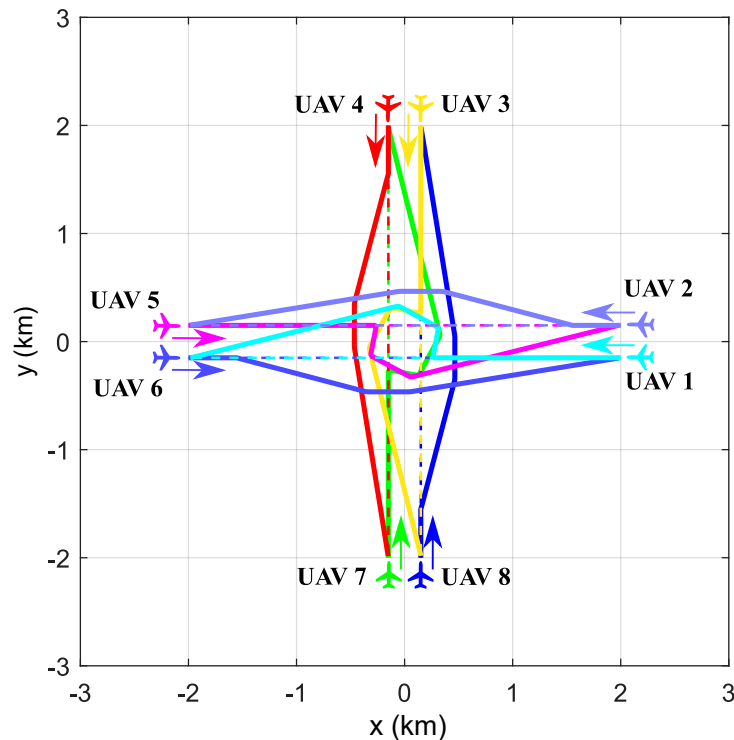
maneuver to the right compliant with the ICAO rules. Circular markers indicate the UAVs' positions at the time the collision is predicted. As in the previous scenario, the aircraft avoid a sort of fictitious traffic circle having a radius that depends on the number of vehicles, centered at the point where the collision is expected, before reaching their own destination points on the opposite side. For this test case, given the large number of aircraft, we preferred not to report the entire table with the minimum distances between aircraft, which, anyway, were never below 221 m.



**Figure 13.** Scenario #4. Twenty UAVs uniformly distributed as a circular pattern converging at the center of the airspace and reaching the destination points on the opposite side. UAVs' initial position  $A_i = (5 \cos(\alpha_i); 5 \sin(\alpha_i))$  km,  $\alpha_i = (i - 1)(2\pi/20) \quad \forall i = 1, \dots, 20$ ,  $V_i(0) = V_c$ ,  $\psi_i = \alpha_i + \pi$ . Dotted lines represent the UAVs' planned paths, whereas solid lines represent the actual trajectories. Circular markers indicate the UAVs' positions at the time the collision is predicted.

#### 5.5. Scenario #5: Four Converging Pairs of UAVs

In this scenario, eight vehicles grouped into four pairs are involved. All the aircraft follow parallel nominal paths pointing to their own destination points on the opposite side. The distance between parallel paths is 0.3 km. This scenario represents a challenging test case for our collision avoidance algorithm as any turn maneuver of one vehicle can affect the behavior of the others. The planned (dotted lines) and actual trajectories (solid lines) are shown in Figure 14. As can be seen, compliance with the ICAO right of way rules forces all the UAVs to turn right of the colliding aircraft near the center of the airspace. This makes the actual trajectories possibly deviate greatly from the planned ones. Again, as previously done, only the minimum distances between aircraft are resumed in Table 3. In this case as well, the collision avoidance algorithm is able to assure the required minimum value.



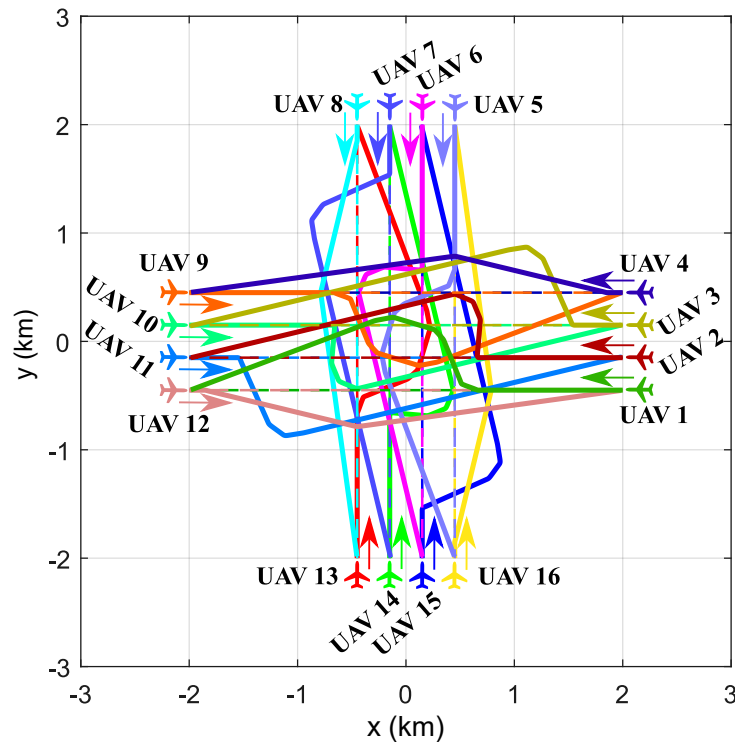
**Figure 14.** Scenario #5. Four pairs of UAVs following parallel paths pointing to the opposite side. Dotted lines represent the UAVs' planned paths, whereas solid lines represent the actual trajectories.

**Table 3.** Scenario #5: Minimum distances between UAVs (m).

	1	2	3	4	5	6	7	8
1	-	294	643	526	221	372	372	702
2		-	930	643	658	702	221	658
3			-	294	658	221	702	658
4				-	702	372	372	221
5					-	294	643	930
6						-	526	643
7							-	294
8								-

#### 5.6. Scenario #6: Sixteen Converging UAVs

In this scenario, to further test our anti-collision algorithm, sixteen aircraft, grouped into four formations of four UAVs each, are involved following parallel paths pointing to their own destination points on the opposite side. The distance between parallel paths is 0.3 km. Planned paths (dotted lines) and actual trajectories (solid lines) are shown in Figure 15. As we can see, with the exception of UAVs on the right side of the flight formations (UAVs 4, 8, 12, and 16), all the others are forced to deviate strongly from their planned trajectories to avoid a collision with other vehicles while complying with the ICAO rules. In this test case as well, given the large number of aircraft, we preferred not to report the entire table with the minimum distances between aircraft, which, anyway, were never below 213 m.

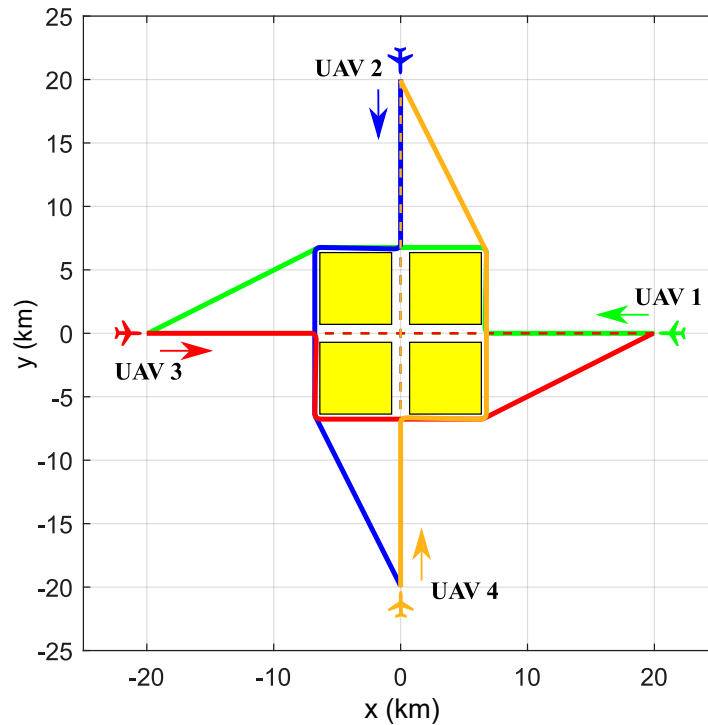


**Figure 15.** Scenario #6. Sixteen UAVs following parallel paths pointing to the opposite side. Dotted lines represent the UAVs' planned paths, whereas solid lines represent the actual trajectories.

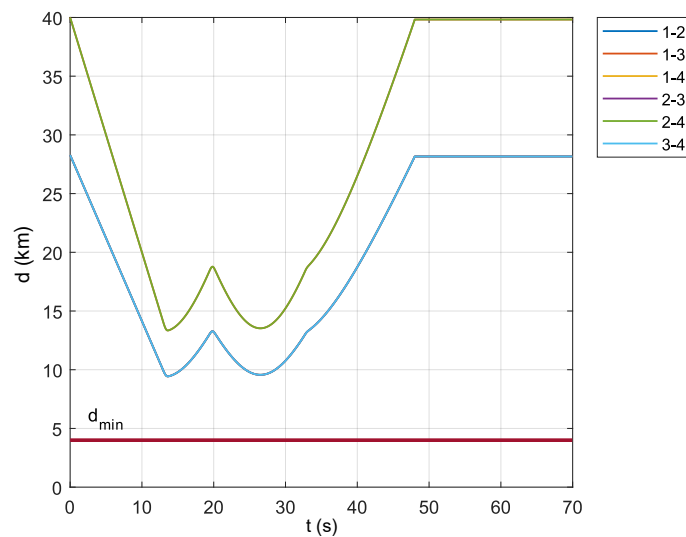
#### 5.7. Scenario #7: Four Converging UAVs in the Presence of Fixed Obstacles

In this scenario, four UAVs are involved following perpendicular flight paths in the presence of four fixed obstacles placed near the center point of the airspace where collision is expected. All the aircraft have to reach their own destination points placed on the opposite side. In this scenario, the desired minimum distance was chosen equal to  $d_{min} = 4$  km and the minimum turn radius  $R_{min} = 400$  m. All nominal paths (dotted lines) converge towards the center and pass through the two narrow perpendicular channels created by the four obstacles. Each vehicle must re-plan its trajectory to avoid a collision by performing a turn maneuver in order to fly out of the area occupied by the obstacles (see Figure 16). Each aircraft, in its prediction phase, considers a fictitious obstacle centered at the predicted collision point that overlaps with the existing real obstacles. Therefore, as stated in Section 4.2, a unique larger obstacle is considered by merging the existing ones. For this reason, the resulting path does not pass in the center of the scenario. The distances over time between UAVs are shown in Figure 17. As we can see, mutual distances never fall below the minimum required value. Due to the symmetry of the scenario, the distances between Aircraft 1–3 and 2–4 are superimposed, the same as the distances between Aircraft 1–2, 1–4, 2–3, and 3–4.

Finally, to update the aircraft trajectories, the proposed algorithm never took more than 0.01 s for the scenarios without obstacles and 0.03 s for the scenario with static obstacles by using an Intel i7 based laptop. This feature suggests that the proposed procedure can be considered a promising approach for real-time applications.



**Figure 16.** Scenario #7. Four UAVs converging to the center of the airspace and pointing to the opposite side in the presence of fixed obstacles.  $A_i = (20 \cos(\alpha_i); 20 \sin(\alpha_i))$  km,  $\alpha_i = (i - 1)(2\pi/4)$   $\forall i = 1, \dots, 4$ ,  $V_i(0) = V_c$ ,  $\psi_i = \alpha_i + \pi$ . Dotted lines represent the UAVs' planned paths, whereas solid lines represent the actual trajectories. Obstacles are represented as yellow boxes.



**Figure 17.** Scenario #7. Distances over time between UAVs. The distances between UAVs 1–3 and 2–4 are superimposed, the same as the distances between UAVs 1–2, 1–4, 2–3, and 3–4.

## 6. Conclusions

This paper describes a novel procedure to generate optimum flight paths in dynamic environments featuring the simultaneous presence of multiple flying vehicles and fixed or moving obstacles. A real-time collision avoidance algorithm was developed based on the Essential Visibility Graph (EVG) solving the optimization problem as a minimum cost piecewise linear path search. It was assumed that

all the air vehicles present in the airspace share information about their position, speed, and direction. This way, if a possible collision is detected within a fixed time horizon, a re-planning procedure is called that updates the EVG and changes the aircraft's current trajectory, avoiding UAVs that are on a collision course. The use of Dubins' curves allows obtaining smoothed paths compliant with the flight mechanics constraints. In view of possible future applications in hybrid scenarios where both manned and unmanned aircraft share the same airspace, visual flight rules compliant with the ICAO Annex II Right of Way were implemented. To test the effectiveness of the proposed algorithm, different operational scenarios of increasing complexity were defined. All the numerical simulations proved the algorithm's capability to avoid possible collisions by quickly changing the aircraft's current trajectories, assuring a minimum safety distance between air vehicles as well. Furthermore, the low computational burden suggests that the proposed procedure can be considered a promising approach for real-time applications. Future research activities will be directed toward the implementation of this new anti-collision algorithm in a guidance and navigation system taking into account possible atmospheric disturbances and model uncertainties. This will assure the identification of real flyable trajectories.

**Author Contributions:** Conceptualization, methodology, investigation, formal analysis, and writing, E.D., I.N., L.B., and M.M. Data curation, visualization, and software development, E.D. and I.N. All authors read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by REGIONE CAMPANIA, SCAVIR Project "Advanced Configurations Studies for an Innovative Regional Aircraft" CUP B43D18000210007.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- DeGarmo, M.; Nelson, G.M. Prospective unmanned aerial vehicle operations in the future national airspace system. In Proceedings of the AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum, Chicago, IL, USA, 20–22 September 2004; pp. 20–23.
- Mujumdar, A.; Padhi, R. Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles. *J. Aerosp. Comput. Inf. Commun.* **2011**, *8*, 17–41. [\[CrossRef\]](#)
- Barrado, C.; Boyero, M.; Brucculeri, L.; Ferrara, G.; Hately, A.; Hullah, P.; Martin-Marrero, D.; Pastor, E.; Rushton, A.P.; Volkert, A. U-Space Concept of Operations: A Key Enabler for Opening Airspace to Emerging Low-Altitude Operations. *Aerospace* **2020**, *7*, 24. [\[CrossRef\]](#)
- Wargo, C.A.; Church, G.C.; Glaneueski, J.; Strout, M. Unmanned Aircraft Systems (UAS) research and future analysis. In Proceedings of the 2014 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014; pp. 1–16.
- Pérez-Castán, J.A.; Rodríguez-Sanz, Á.; Gómez Comendador, V.F.; Arnaldo Valdés, R.M. ATC Separation Assurance for RPASs and Conventional Aircraft in En-Route Airspace. *Safety* **2019**, *5*, 41. [\[CrossRef\]](#)
- ICAO. Unmanned Aircraft Systems Traffic Management (UTM)—A Common Framework with Core Principles for Global Harmonization. 2019. Available online: <https://www.icao.int/safety/UA/Documents/Forms/AllItems.aspx> (accessed on 13 August 2020).
- Cabreira, T.M.; Brisolara, L.B.; Ferreira, P.R., Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **2019**, *3*, 4. [\[CrossRef\]](#)
- Zhang, H.Y.; Lin, W.M.; Chen, A.X. Path planning for the mobile robot: A review. *Symmetry* **2018**, *10*, 450. [\[CrossRef\]](#)
- Shin, H.; Chae, J. A Performance Review of Collision-Free Path Planning Algorithms. *Electronics* **2020**, *9*, 316. [\[CrossRef\]](#)
- Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [\[CrossRef\]](#)
- Isaacs, J.T.; Hespanha, J.P. Dubins traveling salesman problem with neighborhoods: A graph-based approach. *Algorithms* **2013**, *6*, 84–99. [\[CrossRef\]](#)
- Owen, M.; Beard, R.W.; McLain, T.W. Implementing Dubins airplane paths on fixed-wing uavs. In *Handbook of Unmanned Aerial Vehicles*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1677–1701.

13. Kikutis, R.; Stankūnas, J.; Rudinskas, D.; Masiulionis, T. Adaptation of Dubins paths for UAV ground obstacle avoidance when using a low cost on-board GNSS sensor. *Sensors* **2017**, *17*, 2223. [\[CrossRef\]](#)
14. Zhu, M.; Zhang, X.; Luo, H.; Wang, G.; Zhang, B. Optimization Dubins path of multiple UAVs for post-earthquake rapid-assessment. *Appl. Sci.* **2020**, *10*, 1388. [\[CrossRef\]](#)
15. Yan, F.; Liu, Y.S.; Xiao, J.Z. Path planning in complex 3D environments using a probabilistic roadmap method. *Int. J. Autom. Comput.* **2013**, *10*, 525–533. [\[CrossRef\]](#)
16. Madridano, Á.; Al-Kaff, A.; Martín, D. 3D trajectory planning method for uavs swarm in building emergencies. *Sensors* **2020**, *20*, 642. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Bortoff, S.A. Path planning for UAVs. In Proceedings of the American Control Conference, Chicago, IL, USA, 28–30 June 2000; Volume 1, pp. 364–368.
18. Huang, H.; Savkin, A.V.; Li, X. Reactive Autonomous Navigation of UAVs for Dynamic Sensing Coverage of Mobile Ground Targets. *Sensors* **2020**, *20*, 3720. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Davis, J.; Perhinschi, M.; Wilburn, B.; Karas, O. Development of a modified Voronoi algorithm for UAV path planning and obstacle avoidance. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012; p. 4904.
20. Tong, H. Path planning of UAV based on Voronoi diagram and DPSO. *Procedia Eng.* **2012**, *29*, 4198–4203. [\[CrossRef\]](#)
21. Scherer, S.; Singh, S.; Chamberlain, L.; Elgersma, M. Flying fast and low among obstacles: Methodology and experiments. *Int. J. Robot. Res.* **2008**, *27*, 549–574. [\[CrossRef\]](#)
22. Jun, M.; D’Andrea, R. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In *Cooperative Control: Models, Applications and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 95–110.
23. Li, Y.; Chen, H.; Er, M.J.; Wang, X. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics* **2011**, *21*, 876–885. [\[CrossRef\]](#)
24. Wahyunggoro, O.; Cahyadi, A.I. Quadrotor Path Planning Based On Modified Fuzzy Cell Decomposition Algorithm. *Telkomnika* **2016**, *14*, 655–664.
25. Samaniego, F.; Sanchis, J.; García-Nieto, S.; Simarro, R. Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs. *Electronics* **2019**, *8*, 306. [\[CrossRef\]](#)
26. Mattei, M.; Blasi, L. Smooth flight trajectory planning in the presence of no-fly zones and obstacles. *J. Guid. Control Dyn.* **2010**, *33*, 454–462. [\[CrossRef\]](#)
27. Shao, Z.; Yan, F.; Zhou, Z.; Zhu, X. Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization. *Appl. Sci.* **2019**, *9*, 2621. [\[CrossRef\]](#)
28. Samaniego, F.; Sanchis, J.; Garcia-Nieto, S.; Simarro, R. Smooth 3D Path Planning by Means of Multiobjective Optimization for Fixed-Wing UAVs. *Electronics* **2020**, *9*, 51. [\[CrossRef\]](#)
29. Scholer, F.; la Cour-Harbo, A.; Bisgaard, M. Configuration space and visibility graph generation from geometric workspaces for uavs. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011; p. 6416.
30. Maini, P.; Sujit, P. Path planning for a uav with kinematic constraints in the presence of polygonal obstacles. In Proceedings of the Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 62–67.
31. Majeed, A.; Lee, S. A fast global flight path planning algorithm based on space circumscription and sparse visibility graph for unmanned aerial vehicle. *Electronics* **2018**, *7*, 375. [\[CrossRef\]](#)
32. Latombe, J.C. *Robot Motion Planning*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 124.
33. Rohnert, H. Shortest paths in the plane with convex polygonal obstacles. *Inf. Process. Lett.* **1986**, *23*, 71–76. [\[CrossRef\]](#)
34. D’Amato, E.; Mattei, M.; Notaro, I. Bi-level Flight Path Planning of UAV Formations with Collision Avoidance. *J. Intell. Robot. Syst.* **2018**, *93*, 1–19. [\[CrossRef\]](#)
35. D’Amato, E.; Notaro, I.; Mattei, M. Optimal Flight Paths over Essential Visibility Graphs. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 708–714.
36. D’Amato, E.; Notaro, I.; Blasi, L.; Mattei, M. Smooth Path planning for Fixed-Wing Aircraft in 3D Environment Using a Layered Essential Visibility Graph. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 9–18.

37. Duleba, I.; Sasiadek, J.Z. Nonholonomic motion planning based on Newton algorithm with energy optimization. *IEEE Trans. Control. Syst. Technol.* **2003**, *11*, 355–363. [\[CrossRef\]](#)
38. Chen, Y.; Yu, J.; Su, X.; Luo, G. Path planning for multi-UAV formation. *J. Intell. Robot. Syst.* **2015**, *77*, 229–246. [\[CrossRef\]](#)
39. D’Amato, E.; Notaro, I.; Mattei, M. Distributed collision avoidance for unmanned aerial vehicles integration in the civil airspace. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA, 12–15 June 2018; pp. 94–102.
40. Wu, Q.; Chen, Z.; Wang, L.; Lin, H.; Jiang, Z.; Li, S.; Chen, D. Real-Time Dynamic Path Planning of Mobile Robots: A Novel Hybrid Heuristic Optimization Algorithm. *Sensors* **2020**, *20*, 188. [\[CrossRef\]](#)
41. Chen, Y.; Luo, G.; Mei, Y.; Yu, J.; Su, X. UAV path planning using artificial potential field method updated by optimal control theory. *Int. J. Syst. Sci.* **2016**, *47*, 1407–1420. [\[CrossRef\]](#)
42. Dever, C.; Mettler, B.; Feron, E.; Popovic, J.; McConley, M. Nonlinear trajectory generation for autonomous vehicles via parametrized maneuver classes. *J. Guid. Control Dyn.* **2006**, *29*, 289–302. [\[CrossRef\]](#)
43. Frazzoli, E.; Dahleh, M.A.; Feron, E. Real-time motion planning for agile autonomous vehicles. *J. Guid. Control Dyn.* **2002**, *25*, 116–129. [\[CrossRef\]](#)
44. Borrelli, F.; Subramanian, D.; Raghunathan, A.U.; Biegler, L.T. MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006.
45. Blasi, L.; Barbato, S.; D’Amato, E. A mixed probabilistic–geometric strategy for UAV optimum flight path identification based on bit-coded basic maneuvers. *Aerosp. Sci. Technol.* **2017**, *71*, 1–11. [\[CrossRef\]](#)
46. Bilimoria, K.; Sridhar, B.; Chatterji, G. Effects of conflict detection methods for air traffic management. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, USA, 29–31 July 1996.
47. Hwang, I.; Kim, J.; Tomlin, C.; McNally, D.; Gong, C.; Rantanen, E.; Naseri, A.; Neogi, N. Protocol-based conflict resolution for air traffic control. *Air Traffic Control Q.* **2007**, *15*, 1–34. [\[CrossRef\]](#)
48. Pallottino, L.; Scordio, V.G.; Bicchi, A.; Frazzoli, E. Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Trans. Robot.* **2007**, *23*, 1170–1183. [\[CrossRef\]](#)
49. Andrews, J. A relative motion analysis of horizontal collision avoidance. In Proceedings of the SAFE Association, Annual Symposium, Las Vegas, NV, USA, 5–8 December 1977; pp. 58–61.
50. Tomlin, C.; Pappas, G.J.; Sastry, S. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Trans. Autom. Control* **1998**, *43*, 509–521. [\[CrossRef\]](#)
51. Frazzoli, E.; Mao, Z.H.; Oh, J.H.; Feron, E. Resolution of conflicts involving many aircraft via semidefinite programming. *J. Guid. Control Dyn.* **2001**, *24*, 79–86. [\[CrossRef\]](#)
52. Hill, J.; Archibald, J.; Stirling, W.; Frost, R. A multi-agent system architecture for distributed air traffic control. In Proceedings of the AIAA Guidance, Navigation and Control Conference, San Francisco, CA, USA, 15–18 August 2005.
53. Ramasamy, S.; Sabatini, R.; Gardi, A. A unified approach to separation assurance and Collision Avoidance for UAS operations and traffic management. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 920–928. doi:10.1109/ICUAS.2017.7991523. [\[CrossRef\]](#)
54. D’Amato, E.; Notaro, I.; Mattei, M. Reactive Collision Avoidance using Essential Visibility Graphs. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; pp. 522–527.
55. D’Amato, E.; Mattei, M.; Notaro, I. Distributed Reactive Model Predictive Control for Collision Avoidance of Unmanned Aerial Vehicles in Civil Airspace. *J. Intell. Robot. Syst.* **2020**, *97*, 185–203. [\[CrossRef\]](#)
56. Eby, M.S.; Kelly, W.E. Free flight separation assurance using distributed algorithms. In Proceedings of the Aerospace Conference, Snowmass, Aspen, CO, USA, 7 March 1999; Volume 2, pp. 429–441.
57. Lalish, E.; Morgansen, K.A.; Tsukamaki, T. Formation tracking control using virtual structures and deconfliction. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 5699–5705.
58. Mastellone, S.; Stipanović, D.M.; Graunke, C.R.; Intlekofer, K.A.; Spong, M.W. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *Int. J. Robot. Res.* **2008**, *27*, 107–126. [\[CrossRef\]](#)



59. Roussos, G.P.; Dimarogonas, D.V.; Kyriakopoulos, K.J. 3D navigation and collision avoidance for a non-holonomic vehicle. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 3512–3517.
60. Tony, L.A.; Ghose, D.; Chakravarthy, A. Avoidance maps: A new concept in UAV collision avoidance. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1483–1492. doi:10.1109/ICUAS.2017.7991382. [[CrossRef](#)]
61. Ha, L.N.N.T.; Bui, D.H.P.; Hong, S.K. Nonlinear control for autonomous trajectory tracking while considering collision avoidance of UAVs based on geometric relations. *Energies* **2019**, *12*, 1551. [[CrossRef](#)]
62. Chen, T.; Shan, J. Continuous constrained attitude regulation of multiple spacecraft on SO (3). *Aerosp. Sci. Technol.* **2020**, *99*, 105769. [[CrossRef](#)]
63. Weerakoon, T.; Ishii, K.; Nassiraei, A.A.F. An artificial potential field based mobile robot navigation method to prevent from deadlock. *J. Artif. Intell. Soft Comput. Res.* **2015**, *5*, 189–203. [[CrossRef](#)]
64. Fiorini, P.; Shiller, Z. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [[CrossRef](#)]
65. Van Den Berg, J.; Snape, J.; Guy, S.J.; Manocha, D. Reciprocal collision avoidance with acceleration-velocity obstacles. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3475–3482.
66. Kufoalor, D.K.M.; Brekke, E.F.; Johansen, T.A. Proactive collision avoidance for ASVs using a dynamic reciprocal velocity obstacles method. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2402–2409.
67. Jyotish, P.N.; Goel, Y.; Kumar, A.S.B.; Krishna, K.M. IVO: Inverse Velocity Obstacles for Real Time Navigation. In Proceedings of the Advances in Robotics 2019, Chennai, India, 2–6 July 2019; pp. 1–6.
68. Wilkie, D.; Van Den Berg, J.; Manocha, D. Generalized velocity obstacles. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 5573–5578.
69. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
70. ICAO. Annex II to the convention on international civil aviation: Aerodromes. In *Rules of the Air ICAO*, 10th ed.; ICAO: Quebec, QC, Canada, July 2005.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).