Metropolis II: Centralised and strategical separation management of UAS in urban environment

Denis Bereziat, Sonia Cafieri, Andrija Vidosavljevic
ENAC, Université de Toulouse
Toulouse, France
denis.bereziat@enac.fr

***Abstract*—This paper presents a centralized and strategical approach for Unmanned aircraft systems Traffic Management (UTM) to design optimal 4D trajectories minimizing the total flight time of all vehicles over a given time window. Potential losses of pairwise separation between vehicles are modeled and solved. A 4D trajectory is modeled by choosing an horizontal path (with an associated nominal speed profile), a departure slot and a cruising flight level. The problem is formulated as a mixed-integer linear program. A two-step solution approach is proposed that takes into account operational requirements, such as late flight intention deposits, or static and dynamic geofences; and that is able to deal with very high traffic density (up to 6300 vehicles in an horizon of one hour). Experimental results show that it is possible, by delaying flights at the departure or modifying their 4D route (vertically or horizontally), to obtain Unmanned Aircraft Systems (UAS) flyable trajectories that avoid losses of separation and minimize the total flown time.**

***Keywords*—Centralized flight management, Unmanned aircraft systems, Strategic separation, UTM, U-space, Trajectory design**

I. Introduction

Unmanned Aircraft Systems (UAS) technology has seen many breakthroughs over the past decades. This, together with a wider accessibility of equipment, opens the door for new services in the urban environment, such as surveillance, deliveries and even passenger transportation. Recent studies foresee a large increase in fleet size in the near future, with an estimation of up to 400 thousand of UAS in Europe for commercial and governmental activities by 2050 [1]. Expected high traffic densities in the urban environment, characterised by narrow routes, high buildings and dense population, puts pressure on the future UAS Traffic Management (UTM) system, that must achieve both efficiency and safety targets. The degree of structuring of the airspace that would best accommodate such demand has been studied in the Metropolis project [7]. Results show that over-structuring the airspace reduces capacity and efficiency of the UTM system, without a significant impact on safety compared to unstructured airspaces; on the contrary, structuring the airspace only reducing relative velocities between vehicles is beneficial while a limited deterioration in its efficiency is observed [8]. The Metropolis II project [2] aims to study the impact of a level of centralization of the separation management on the performance of the UTM system. To achieve this goal, three UTM concepts, characterized by an increasing level of centralisation, have been foreseen. This paper presents the so called *centralised* concept, in which vehicle separation and flight management are handled by a single centralised authority.

This paper studies a centralised and strategical approach for UAS trajectory planning in the context of UTM. The aim is to reduce the number of *losses of separation*[1] (LoS) between UAS through strategical planning. It is structured as follows. Section II presents the problem definition and related works. Section III presents the proposed UAS 4D trajectory modelling, and analyses potential losses of separation between pairs of vehicles. A mixed-integer linear programming formulation is presented in Section IV. In Section V a two-step solution approach is presented. Numerical results are discussed in Section VI. Section VII draws conclusions and shows future research directions.

II. Problem definition and related work

The problem studied in this paper consists of strategic design of 4D trajectories for a set of UAS vehicles whose flight intentions are given, where potential losses of separation are prevented and the total travel time over all the trajectories is minimized. Every flight intention is defined by a departure and an arrival vertiport and a preferred departure time.

This problem shares similarities with the classical Air Traffic Flow Management Problem in ATM, whose aim is to assign routes to a set of flights respecting system capacities (the airports departure/arrival, and sector capacities). A seminal work on this problem is [10], where the authors present a Mixed-Integer Programming model based on assigning ground holdings (delays) and speed adjustments to the flights (flight routes are considered as given). This model is further improved in [11] with the addition of a rerouting option (a choice of ATC sectors through which a flight can pass). These works, however, consider capacity limitations only, not taking into account the prevention of LoS, which requires a higher accuracy in the trajectory definition (defined as list of ATC sectors in the above mentioned works). In [12], Pelegrín et al. introduce a tactical conflict resolution model relying on speed adjustments to vehicle trajectories. Although the authors focus on conflict solving (i.e., prevention of LoS), they consider a tactical resolution where trajectories are assumed already generated and affected to flights. Dai et al. introduce

[1]Loss of separation, usually referred as conflict in ATM, represents a situation where a physical separation between two aerial vehicles is less than given separation norms, i.e., minimal horizontal and vertical separations.

in [13] an conflict-free A*-based algorithm for 4D trajectory assignment, that relies on a First-Come First-Served paradigm. The flight trajectories are designed iteratively in the order of flight plan deposits, considering each time the already generated trajectories as constraints. The FCFS paradigm for strategic trajectory planning is likely to provide sub-optimal solution when the minimization of the total flight time is addressed. We present in the next sections our approach to the design of strategical 4D trajectories, handling potential losses of separation and optimizing the total flight time.

III. 4D Trajectory design

This section presents the proposed 4D trajectory modelling. The following hypothesis are considered:

- a UAS climbs/descends vertically over its departure/arrival vertiport;
- a UAS is cleared to fly a given cruising flight level and no additional level changes are allowed;
- during cruising, a UAS follows a predefined route structure modelled as a graph;
- a UAS flies at its nominal cruising or turning (when turn angles require to slow down) speed, as well as at its nominal climbing/descending speed called nominal speed profile.

The first three hypothesis represent a set of UTM flight rules aimed at limiting the possible intersections of UAS trajectories at predefined crossing points (i.e., nodes of the graph representing the route structure). The last is a modelling assumption (vehicle speeds are not decision variables, thus avoiding increasing the size of the problem).

A UAS trajectory is modelled as consisting of three parts: a vertical take-off and climb until the flight level is reached, the cruising phase where the UAS follows the horizontal path, and the vertical descent and landing phase. Therefore, the trajectory modelling relies on the choice of:

- Cruising flight level,
- Ground holding, i.e. a departure delay,
- Horizontal path, given as a list of vertices of the route structure graph.

This completely defines a 4D trajectory. For each such trajectory, one can compute the time of passage at any point p of the trajectory f, $t_{p,f}$ as:

$$t_{p,f} = t^*_{p,f} + d_f + \Delta_{p,f}$$

where

- $t^*_{p,f}$ the time of passage at point p of trajectory f on its assigned horizontal path, computed using nominal speed profile and UAS' dynamics.
- d_f the departure delay assigned to f.
- $\Delta_{p,f}$ the time required to climb or to climb and descend to the flight level of point p that depends on the phase of the flight (climbing, cruising, descending).
  In particular, for a point p in a cruising phase, it corresponds to the time needed to climb till the chosen flight level.

Since the aim is to obtain 4D trajectories preventing losses of separation, multiple alternative horizontal paths are pre-generated for each flight intention. Then, for each pair of such alternative horizontal paths, spatial intersections are detected and the required time to ensure separation is computed.

A. Alternative horizontal paths generation

In this study, the route structure is modelled as a graph, where arcs represent *streets* and vertices represent street intersections and vertiports. Hence, the horizontal path of UAS is a path on such a graph. An A* algorithm is used to find a path from the origin to the destination vertex (representing departure and arrival vertiports) that minimizes a given cost function. In this work, the objective is to minimize the total flight time. The choice of a path directly impacts the flight time of a UAS, since the UAS speed is decreased with respect to its nominal value in correspondence of "turning angle" between arcs, that in turn depend on the path shape. To take this into account, the dual graph is used instead.
In the dual graph, vertices represent arcs of the original graph, i.e., streets. Arcs of the dual graph link two streets (as by definition an arc links two vertices), that allows associating to each arc a weight which takes into account both the travel time on the first street and a penalty related, if any, to turning to the second street. Furthermore, the use of the dual graph allows to easily forbid certain turns (if necessary) by simply omitting an arc between two vertices (representing two connected streets.) Since the UAS dynamics (nominal speed, acceleration, etc.) depends on the specific UAS model, one dual graph is constructed for each UAS model.
Applying the A* algorithm to the dual graph, for each UAS the horizontal path for a given origin-destination pair is generated, minimizing the flight time-based objective function. Alternative path haven't been defined, as one could imagine, as second, third, etc. fastest path that would be very similar to the fastest one. Instead, to generate for each UAS alternative paths, in this work additional intermediate vertices are chosen such that the path to be generated must go through. This is justified by the need of obtaining alternative paths that avoid potentially congested areas, and that additionally are sufficiently different from the first (fastest) generated path.
Several alternative trajectories are generated for every flight intention. Once alternative horizontal paths are generated, spatial intersections between them can be detected and the required separations computed.

B. Detection of potential losses of separation

A *potential loss of separation* (PLoS) between two vehicle horizontal paths is a LoS that may occur between their 4D trajectories considering all possible flight levels and departure delay choices. Considering our modelling hypothesis, a PLoS can occur at a vertex of the graph[2], hence a necessary condition for a PLoS is that two paths share a common vertex. Different

[2]Furthermore, it is possible that PLoS occurs at a shared portion of the horizontal path as it will be explained in section III-C

categories of PLoS, depending on the phase of the involved flights, can be identified, as illustrated on Figure 1:

- Between two cruising UAS (**B** in Figure 1);
- Between a climbing and a cruising UAS (**A**) or between a descending and cruising drone (**C**);
- Between two departing or two landing UAS (not represented in the figure).

PLoS are possible only if certain conditions hold, for each category. As an example, for a PLoS between two cruising UAS they need to be associated to the same flight level. For a PLoS between a climbing/descending and a cruising UAS, the UAS which is evolving needs to be assigned to a higher flight level than the one that is cruising. Finally, two departing/landing UAS involved in a PLoS must share the same vertiport.

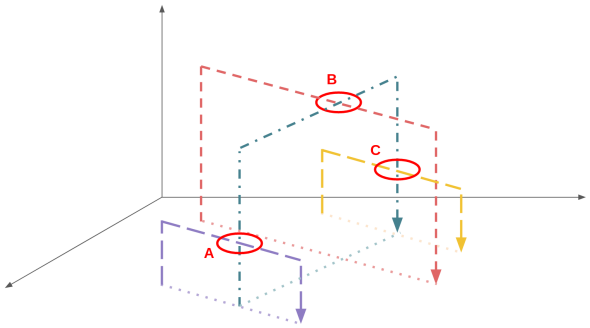

Figure 1. Potential LoS at intersections between trajectories

Even though sharing a common vertex is a necessary condition for PLoS between two path, an additional temporal condition needs to be satisfied too. For every horizontal path, the nominal time of passage over vertices of this path can be computed using the nominal (desired) departure time and the nominal speed profile. It is further possible to determine the earliest and the latest time of passage, considering the possible choices of flight level and departure delay. Hence, the earliest passage time is obtained considering the lowest flight level and no delay, while the latest passage time is obtained considering the highest flight level and maximum delay. If passage times are not separated by at least a given separation minimum, then a PLoS occurs, and a separation needs to be determined. Let us illustrate this on the example of a PLoS between two cruising UAS. Since such PLoS is possible only if both UAS are assigned to the same flight level, only the departure delay may influence the actual passage time over the intersecting vertex. Hence, the earliest passage time, for both UAS, is equal to their nominal passage time, while the latest is equal to the sum of the nominal passage time and the maximum delay.

C. Computation of separation time

Once a PLoS between two paths has been detected, it is necessary to compute the required separation time that ensures the physical separation between the trajectories of the two involved UAS, i.e. that prevents the PLoS resulting in a LoS. More specifically, the required separation time represents the time to be ensured between the passage of the two UAS over the vertex corresponding to the PLoS.

Figure 2 illustrates a generic PLoS situation between two intersecting paths noted $D1$ and $D2$. The points A_1 and B_1 (respectively A_2 and B_2) represent the points on path $D1$ (respectively $D2$) whose distance from the intersection point is the required minimum separation distance; A_1 (A_2) being the point before and B_1 (B_2) the point after the intersection. The minimum separation distance represents the separation norm to be satisfied and depends on the category of PLoS, e.g for cruising UAS a horizontal separation norm is considered and for climbing/cruising UAS a vertical separation norm. In this study we are using the horizontal and vertical minimum separation of 32 meters and 25 feet, respectively, established in [6] based on the ICAO Annex 10 GNSS requirements. As UAS flying on adjacent intersecting streets can be considered physically separated by buildings, in this study the computation of separation times to prevent LoS is focused on the case where UAS, along their flight, are on the same street. Note, however, that this is not a strong assumption, and that the same approach could be used to compute the required separation time for any given route intersection geometry as explained in [14].

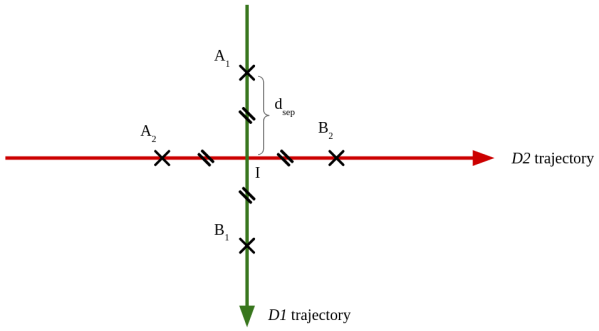

Figure 2. Illustration of a PLoS

Let us refer to Figure 2. If the UAS using path $D1$ passes before the one using $D2$, to ensure their separation the UAS on the $D2$ must not pass yet over point A_2 when the UAS on $D1$ reaches the intersection point; also, the UAS on $D1$ must already pass over point B_1 before the UAS on $D2$ reaches the intersection point. Hence, the required separation time between the time of passage of the UAS on $D1$ and the UAS on $D2$ is the maximum of the time needed to the UAS on $D2$ to fly from A_2 to the intersection point and the time needed to the UAS on $D1$ to fly from the intersection point to B_1. Similarly, in the case when the UAS using path $D2$ passes before the one using $D1$, the required separation time between the passage times is computed as the maximum of the time needed to the UAS on $D1$ to fly from A_1 to the intersection point and the time needed to the UAS on $D2$ to fly from the intersection point to B_2. It should be noted that in the computation of flight times constant speed is not assumed and a nominal speed profile is

used.

In some situations, two paths may share a common portion, i.e., a same subset of consecutive graph vertices, that can be in the same (*trailing PLoS*) or in the inverse order (*face-to-face PLoS*). Figure 3 illustrates such kind of situations, where I_1 and I_2 represent the extremal points of a shared path portion.

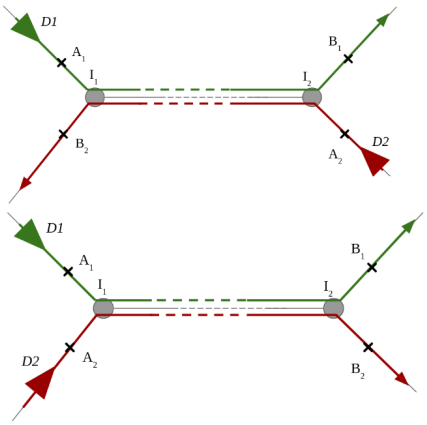

Figure 3. Trailing and face-to-face PLoS

Trailing PLoS. This is similar to the general intersection case presented above, where the two conditions that ensure separation are applied to the two points I_1 and I_2 (instead of one only intersection point as presented above), that corresponds to keeping the passage order on the shared path portion. This is illustrated on the time-distance diagram in Figure 4 for the case of the UAS on $D1$ passing before the one on $D2$. Hence, the required separation time between the time of passage of the UAS on $D1$ and the UAS on $D2$ at point I_1 is the maximum of two terms, the first one being the time needed to the UAS on $D2$ to fly from A_2 to I_1, and the second one being the sum of the time needed to the UAS on $D1$ to fly from I_2 to B_1 and the difference between flight times of UAS on $D1$ and $D2$ from I_1 to I_2. This guarantees the separation at I_1 and I_2 and that the order of passage is maintained. The separation time is computed similarly for the UAS on $D2$ passing before the UAS on $D1$.

Face-to-face PLoS. In this case, if the UAS on $D1$ passes before the UAS on $D2$ over the intersection point I_1, than it must also pass before at the intersection point I_2. Hence, the condition that ensure separation is that the UAS on $D1$ must have already passed over B_1 before the UAS on $D2$ reaches the intersection point I_2. This condition is illustrated on the time-distance diagram in Figure 5. Hence, the required separation time between the time of passage of the UAS on $D1$ and the UAS on $D2$ at point I_1 is equal to the sum of the time needed to the UAS on $D1$ to fly from I_2 to B_1 and flight times of UAS on $D1$ and $D2$ from I_1 to I_2. In the case when the UAS on $D2$ passes before the UAS on $D1$ over the intersection point I_1, the conditions that guarantee separation are equivalent to those for the general intersection.

A last particular case is that of a PLoS between departing

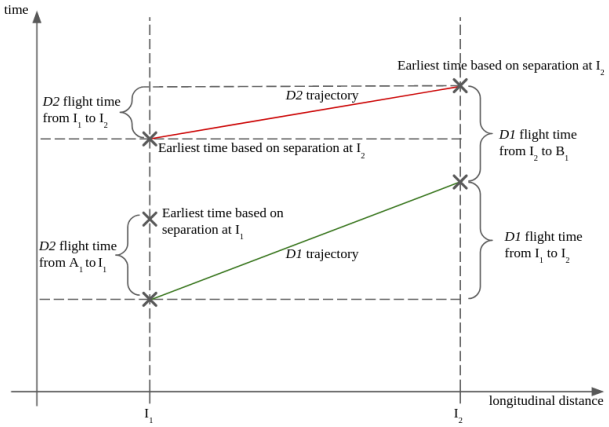

Figure 4. Trailing PLoS conditions, with $D1$ passing before $D2$

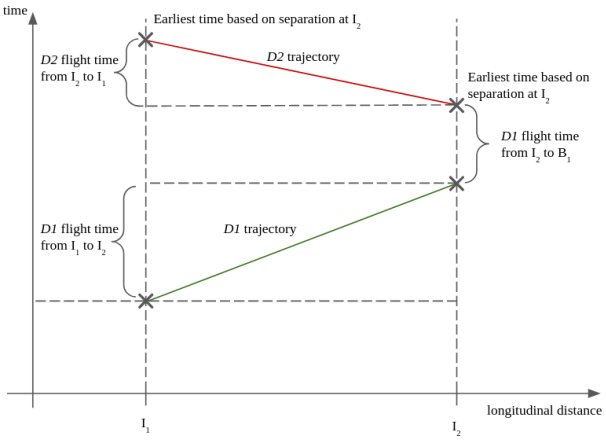

Figure 5. Face-to-face PLoS conditions, with $D1$ passing before $D2$

or landing UAS at the same vertiport. The required separation time is in this case a parameter of the problem.

The complete 4D trajectory of each UAS can now be defined while maintaining separation minima at all time. The above introduced separation times are considered as constraints in the Mixed Integer Linear Programming model presented in the following section.

IV. Mixed Integer Linear Optimization for 4D Trajectory Design

The Mixed Integer Linear Programming (MILP) model for 4D trajectories design is defined in this section. This model assigns a flight level, a ground holding (departure delay) and a horizontal path to each flight. These three parameters define 4D trajectories as presented in Section III. The model constraints guarantee that the separation times (see Section III-C) are maintained, that in return ensure physical separation between UAS i.e. prevent LoS. The objective is to minimize the total travel time (including ground holdings) of all flights.

A. Parameters

- F: set of flights
- K_f, $\forall f \in F$: set of all paths for f

- $K = \cup_{f \in F} K_f$: set of all paths
- $b_k,\ \forall k \in K$: travel time associated with path k
- Y: set of flight levels
- P: set of PLoS defined as in Section III-C
- $\delta(i)\ \forall i \in Y$: time to climb and descend to flight level i
- $D_f,\ \forall f$: maximum ground holding delay for f.

For each PLoS $p \in P$:

- $k_p^1, k_p^2 \in K$: the first and second path of PLoS p
- $f_p^1, f_p^2 \in F$: the first and second flight associated to k_p^1 and k_p^2
- t_p^1, t_p^2: nominal time of passage at point p of path k_p^1, and respectively of k_p^2
- s_p^{12}, s_p^{21}: separation times to be respected if f_p^1 is the first to pass over an intersection point, and respectively if f_p^2 is the first (as defined in Section III-C).

B. Decision variables

For each flight $f \in F$:

- $y_f \in Y$: flight level assigned to flight f
- $d_f \in [0, D_f]$: departure delay assigned to flight f.

For each $k \in K$:

- $x_k \in \{0, 1\}$: 1 if path k is assigned to its corresponding flight, 0 otherwise.

C. Objective function

The objective is to minimize the total duration of all flights:

$$\text{Minimize} \sum_{f \in F} \left(d_f + \delta(y_f) + \sum_{k \in K_f} b_k x_k \right) \tag{1}$$

D. Constraints

$$\sum_{k \in K_f} x_k = 1, \forall f \in F \tag{2}$$

$$(t_p^1 + d_{f_p^1}) - (t_p^2 + d_{f_p^2}) + \Delta_{p, y_{f_p^1}, y_{f_p^2}} + s_p^{12} \leq 0, \forall p \in P \tag{3}$$

OR

$$(t_p^2 + d_{f_p^2}) - (t_p^1 + d_{f_p^1}) + \Delta_{p, y_{f_p^1}, y_{f_p^2}} + s_p^{21} \leq 0, \forall p \in P \tag{4}$$

where $\Delta_{p, y_{f_p^1}, y_{f_p^2}}$ represents a time (offset) depending on the category of PLoS and on the flight level of f_p^1, f_p^2 at the intersection point.
Constraints (2) ensure that one path is assigned to each flight, and constraints (3) and (4) are disjunctive constraints ensuring LoS avoidance. The disjunction is due to the two alternative situations: the first flight passes before the second one at PLoS, or respectively the second one passes before. Disjunction constraints are linearized, as typically done in mathematical programming.

V. Solution approach

For problem instances with small to medium size, the model described in the previous section can be directly solved by a state of the art solver for MILP. However, UTM is usually associated to high traffic density instances. This implies large-scale problems, for which an appropriate solution approach needs to be defined.

The approach considered in this paper consists of two steps. The first one is an optimization of the flight level assignment to each UAS. This is done through a MILP model which assigns flight levels and minimizes the total number of LoS between flights. The second step of this approach consists in solving the model presented in Section IV. For this step, we consider a relaxation of the LoS constraints.

A. Flight level optimization

The number of flight levels available (16 in this study) has an important impact on the total combinatory of the MILP 4D trajectory design model presented in Section IV. On the other hand, flight levels can be assigned beforehand, thus reducing the complexity of the trajectory design.

A MILP model is solved to assign a flight level to each flight intention, while minimizing the total number of LoS considering that flights are assigned their shortest path and no ground holding. The flight levels computed as a solution of this first step are added as constraints in the second step. For each flight f, the flight level constraint is:

$$y_f = y_f^s, \quad \forall f \in F \tag{5}$$

With y_f^s the solution flight level of the first optimization step. This pre-optimization of flight levels reduces the number of variables in the next step of the optimization. While this reduces the computing time, solutions are not guaranteed to be optimal.

B. Relaxation of the LoS constraints

As a second step, we consider the solution of the trajectory design model of Section IV, where flight levels are fixed to the values computed in the first step, and LoS constraints are relaxed. As a consequence of relaxing such constraints, time separations are not guaranteed anymore. A violation variable is introduced for each separation constraint, this variable is proportional to the value of the violation (the time spent in LoS). The objective function is the sum of the total flight time and of the total violation of the constraints, to be minimized. Introducing variables:

- $v_p \in \mathbb{R}^+, \forall p \in P$: constraint violation variable for the separation constraint associated with p.

the relaxed constraints to prevent LoS are:

$$(t_p^1 + d_{f_p^1}) - (t_p^2 + d_{f_p^2}) + \Delta_{p, y_{f_p^1}, y_{f_p^2}} + s_p^{12} \leq v_p, \forall p \in P \tag{6}$$

$$(t_p^2 + d_{f_p^2}) - (t_p^1 + d_{f_p^1}) + \Delta_{p, y_{f_p^1}, y_{f_p^2}} + s_p^{21} \leq v_p, \forall p \in P \tag{7}$$

Note that the new constraints (6) and (7) are the same as (3) and (4) except for the addition of the v_p variable. The new objective function is:

$$\text{Minimize} : V \sum_{p \in P} v_p + \sum_{f \in F} \left(d_f + \delta(y_f) + \sum_{k \in K_f} b_k x_k \right) \quad (8)$$

with V representing the (user-defined) violation penalty factor.

VI. Experimental results

Problem instances considered for the 4D trajectory generation and simulations are described in the first subsection. The 4D trajectories generated using the two-step approach were simulated using the Bluesky traffic simulator [4]. Results of these simulations are discussed in the second subsection.

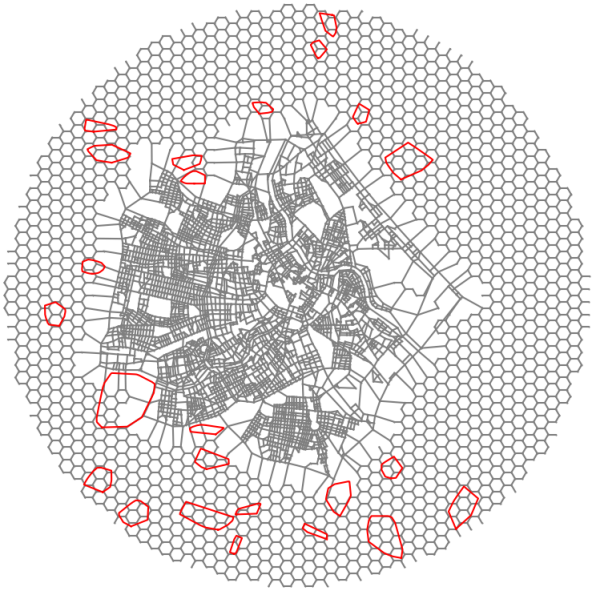

Figure 6. Graph representing the city of Vienna

A. Experimental framework

The problem instances considered in this paper are based on an estimated state of the city of Vienna in the near future. A study of the expected demand for parcel delivery in the city gives an estimation of 110 millions per year for the city [5]. The considered flight intentions are based on this study and stands for UAS parcel deliveries. The hourly number of flight intentions per instances ranges from 1,600 to 6,300 (the traffic density considered are named as in Table I). The departures and arrivals are distributed according to the population density and income of the different areas of the city. The area considered is a cylinder 500 feet high (divided in 16 levels) and 8 kilometers in radius, whose center is the middle of the city center. The city is modeled by a graph representing the streets, that the drones have to follow, shown in Figure 6. Red shapes in Figure 6 are geofenced areas that are forbidden for traffic.

TABLE I. Traffic densities and corresponding number of flights

Traffic density	Very Low	Low	Medium	High
Number of flights	1600	3300	5000	6600

B. Results & Discussion

We focus the analysis of experimental results on the impact of the solution on safety, and on its efficiency with respect to the total flight duration. Both MILP models of the two-step approach are solved using a state of the art commercial solver (Gurobi [3]). The 4D trajectories generated are then simulated using the Bluesky traffic simulator. The Bluesky simulator relies on a more realistic UAS dynamics than that considered in the model presented in Section IV. Thus, the time of passage observed at 4D waypoints in the simulation differs slightly from the one generated by the optimization model. Furthermore, let us recall that a relaxation of the LoS constraints is considered in the 4D trajectory design model, in order to reduce the model complexity and be able to handle high densities instances. This is responsible for some LoS that can be observed in the Bluesky simulations. Here a problem instance is defined as a set of flight intentions. Results for each traffic density are the average results over 20 instances.

For each instance, a baseline solution where no optimization is performed is simulated to give insights on the effects of the proposed strategic optimization. More precisely, a baseline solution is such that each flight takes its shortest route and no departure delay is applied. For the sake of fair comparison, the flights of baseline solutions are spread on flight levels following the same distribution as in the optimized solution.

TABLE II. Number of LoS solved at different traffic densities

Traffic density	Baseline number of LoS	Optimized number of LoS	LoS number reduction
Very Low	833	145	82 %
Low	1394	487	65 %
Medium	3842	1016	73 %
High	5252	1793	65 %

Table II shows the number of LoS (when 2 UAS come closer than the 32 meter limit, as defined in Section III-C) recorded during simulations using both baseline solutions and optimized solutions. Table III and Figure 7 show the cumulative time spent in LoS in both cases. The optimization reduces the number of LoS in simulation by 82% for the lowest traffic densities and by 65% for the highest. The cumulative time spent in LoS is reduced between 91% at low traffic density to 80% at higher densities. The percentage of LoS left after optimization in the simulation increases with the traffic density. The same can be observed with the cumulative time spent in LoS. Reductions in cumulative time spent in LoS are more significant than those in the number of LoS. This is explained by the use of constraint relaxations in the 4D trajectory design model, which aims to minimize the cumulative time spent in LoS instead of the number of LoS.

TABLE III. CUMULATIVE LOS TIME AT DIFFERENT DENSITIES

Traffic Density	Baseline cumulative LoS time (s)	Optimized cumulative LoS time (s)	LoS time reduction
Very Low	4444	392	91 %
Low	9881	1270	87 %
Medium	19962	2654	86 %
High	26486	5241	80 %

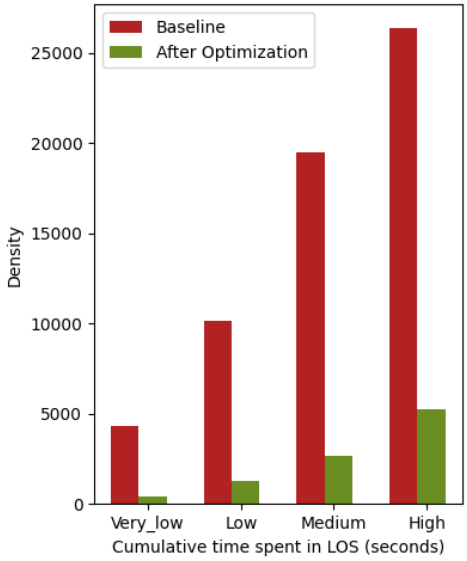

Figure 7. Cumulative time spent in LoS at different densities

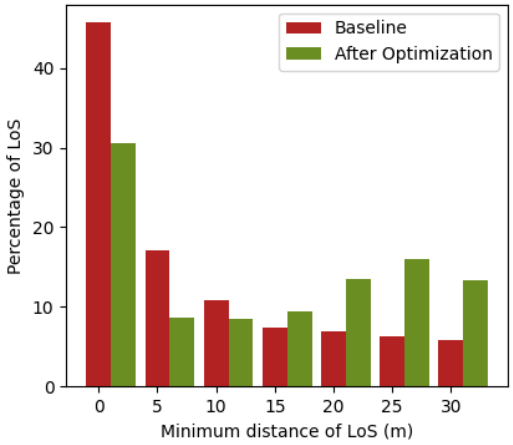

Figure 8. Distribution of minimum distance reached of LoS

During simulations, each time a LoS occurs, the minimum distance reached between the involved UAS is recorded. This distance gives insights on the severity of the LoS that occurred. The shorter the distance reached, the worse the safety violation is. In Figure 8 the distribution of the minimum distance recorded on average is illustrated. In both baseline and optimized solutions, residual LoS will imply some tactical intervention. However, it can be observed that LoS occurring in the baseline solution reach smaller minimum distances than in the optimized solution, where the severity of the residual LoS is reduced. Therefore the proposed traffic optimization reduces both the number and severity of residual LoS, making them easier to manage by tactical intervention.
Figure 9 displays the average increase in flight duration, due to departure delays decided during the optimization process. The average increase ranges from 68 seconds, at low densities, to 78, at the highest, for an average flight duration of almost 700 seconds. The increase of flight duration is then on average of 10%. Figure 10 shows the distribution of flights on the flight levels at different densities. We can observe that due to the minimization of the flight time, lower flight levels are filled more than higher ones. The distribution of the average flight duration due to the different considered decisions to perform trajectory optimization inducing delays is displayed in Figure 11. We observe that the choice of alternative paths causes delays having the strongest impact. On average, less than 5 % of the total delay comes from ground holdings, less than 20 % from the flight level assignment, while the rest is due to the choice of alternative paths. The very low impact of ground holdings on the total delay is likely due to the fact that, the higher the density of traffic is, the more trajectories interact with one another and generate more PLoS. This as a consequence, reduces the possible values of ground holding delay that are compatible with those PLoS. It is worth noting that since flights in the baseline solutions follow their shortest paths. Their flights are shorter on average and less time is spent in the air. This tends to reduce the probability of conflicts between UAS as seen in [15].

VII. CONCLUSION AND FUTURE RESEARCH

This paper presents a two step approach for optimal design of 4D trajectories minimizing the total flight time that prevents losses of separation between UAS. Results show that with strategical rerouting and ground holding assignation, the total time spent in LoS during operations can be reduced by 91% to 80% over high traffic densities (reaching up to 6300 UAS in an hour) for an average increase in flight duration of 10%. Hence, it can be concluded that by making use of the information available to a centralized authority, most tactical interventions can be avoided for a reasonable cost (additional flight time). Results show that while our approach allows for rerouting (vertically or horizontally) and ground holdings to be assigned to prevent LoS, at high traffic densities, it assigns rerouting in most cases. Therefore, future works may consider other options, like speed assignments. Furthermore, the presented approach relies on the assumption that UAS accurately follow their given 4D trajectories. This is unlikely to hold in a realistic context, for instance, due to meteorological events causing delays. Thus, future works will be mainly focused on the use of robust optimization approaches.

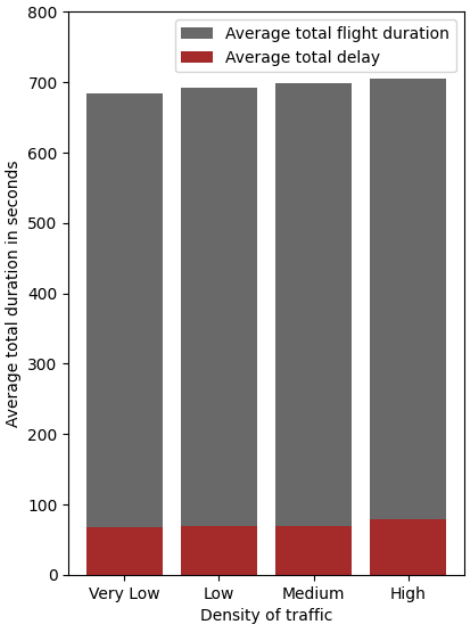

Figure 9. Average total travel time, and total delay at different densities after optimization

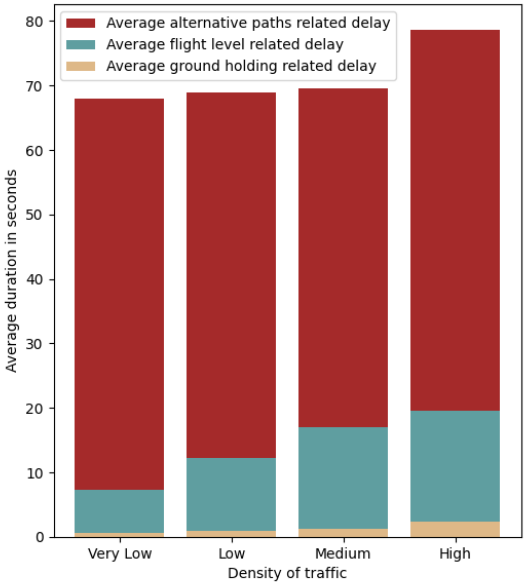

Figure 11. Causes of delay

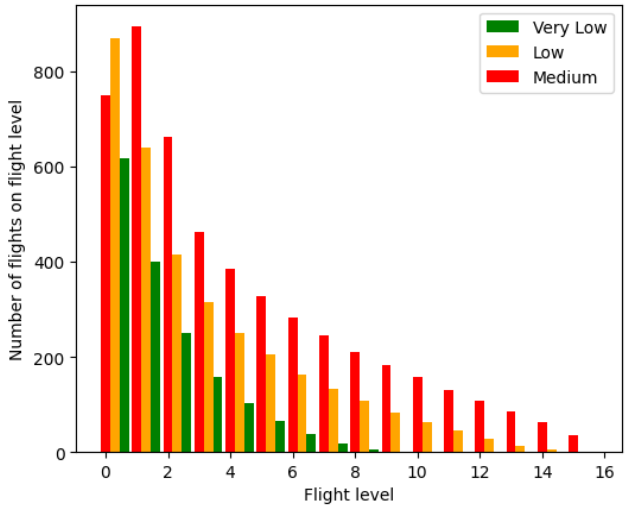

Figure 10. Number of flights on each flight level at different densities

ACKNOWLEDGMENT

This research has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 892928 (Metropolis II).

REFERENCES

[1] SESAR, *European Drones Outlook Study*. 2016.
[2] Metropolis II Consortium, Metropolis II, 2022.
[3] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2022, https://www.gurobi.com
[4] J. Hoekstra and J. Ellerbroek, *"Bluesky ATC simulator project: an open data and open source approach"*, in *Conference: International Conference on Research in Air Transportation*, 2016
[5] "Deliverables - Metropolis 2." [Online]. Available: https://metropolis2.eu/deliverables/
[6] "Deliverables - Bubbles D4.1 : Algorithm for analysing the collision risk" [Online]. Available: https://www.bubbles-project.eu/wp-content/themes/bubbles/Deliverables
[7] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, et al. *Metropolis: Relating Airspace Structure and Capacity for Extreme Traffic Densities* in *ATM seminar 2015, 11th USA/EUROPE Air Traffic Management R&D Seminar*, FAA & Eurocontrol, Jun 2015, Lisboa, Portugal. hal-01168662
[8] E. Sunil, J. Ellerbroek, J. Hoekstra, A. Vidosavljevic, M. Arntzen, et al. *An Analysis of Decentralized Airspace Structure and Capacity Using Fast-Time Simulations*. Journal of Guidance, Control, and Dynamics, American Institute of Aeronautics and Astronautics, 2017, 40 (1), pp. 38-51. doi 10.2514/1.G000528
[9] S. Mondoloni, N. Rozen. *Aircraft trajectory prediction and synchronization for air traffic management applications*, Progress in Aerospace Sciences, 2020, v. 119 doi 10.1016/2020.100640
[10] D. Bertsimas & S. Patterson. *The Air Traffic Flow Management Problem with Enroute Capacities*. Operations Research Vol. 46, No. 3, May-June 1998
[11] D. Bertsimas, G. Lulli, A. Odoni. *An Integer Optimization Approach to Large-Scale Air Traffic Flow Management*, Operations Research Vol. 59, No. 1, Jan-Feb 2011, pp. 211-227 doi 10.1287/opre.1100.0899
[12] Mercedes Pelegrín, Claudia dAmbrosio, Rémi Delmas, Youssef Hamadi. *Urban Air Mobility: From Complex Tactical Conflict Resolution to Network Design and Fairness Insights*. 2021. hal-03299573
[13] W. Dai, B. Pang, Kin Huat Low. *Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy*. Aerospace Science and Technology Vol. 119, 2021.
[14] W. Siddiqee. *A Mathematical Model for Predicting the Number of Potential Conflict Situations at Intersecting Air Routes*. Transportation Science , 1973, Vol. 7, No. 2, pp. 158-167
[15] E. Sunil, J. Ellerbroek, J. M. Hoekstra and J. Maas, *Three-dimensional conflict count models for unstructured and layered airspace designs*, Transportation Research Part C: Emerging Technologies, no. 95, pp. 295-319, 2018.