Proposal of UAS Strategic Conflict Detection concept with a centralised service in multi-USSP environment using an octree data structure

Sandra Amarillo, Álex Sanchis, Brando Fraiz Freire and Juan V. Balbastre
Air Navigation System Lab, ITACA Institute
Universitat Politècnica de València
Valencia, Spain
{sanamva, asanmar4, bfrafre, jbalbast}@upv.edu.es

***Abstract*—U-space regulation require the use of specific U-space services (USSs) to ensure safe UAS operations within the so-called U-space airspaces. One of the key services is the Flight Authorisation Service (FAS). The FAS is responsible for granting authorisation for UAS flights by reviewing flight authorisation requests (FARs), also known as U-plans. To obtain acceptance of the request, an efficient strategic conflict detection is needed. This paper presents an approach to detect strategic conflicts in U-space airspaces in multi USSP environments, facilitating the resolution of potential conflicts between UAS 4D trajectories included in the U-plans submitted to the FAS. The proposal is based on regulatory requirements and inspired by the discretised airspace set out in ASTM F3548-21. The services developed in this proposal encompass the Flight Authorisation Service, as defined in the U-space regulation, and two additional services: the U-Plan Preparation Service (UPPS) and the Search For Overlap Service (SFOS). The UPPS assists operators in creating their U-plan by defining the intended trajectory waypoints and is responsible for the creation of realistic 4D operational intents considering flight uncertainties by using the PX4 autopilot software-in-the-loop (SITL) environment with gazebo as the simulation platform. The SFOS implements an octree data structure to build and manage in an efficient way the airspace representation used by ASTM F3548-21 to enable the operation of multiple USSPs in the same U-space airspace. The proposed solution has been evaluated through simulations with varying level of UAS traffic density as well as different trajectory types (linear and area-based). The results demonstrate the algorithm's ability to detect conflicts and the time required for detection in different density scenarios, and show the impact of the area-based trajectories.**

***Index Terms*—Flight authorisation, strategic conflict detection**

I. Introduction

Unmanned Aerial Systems (UAS) were initially developed for military purposes. However, since 2010 the envisaged positive impact on economy and society of using UAS for commercial applications has raised thrilling expectations among people and industry.

The anticipated expansion of UAS applications implies the need for innovative systems and procedures to facilitate their safe and efficient integration in the airspace, as it was previously implemented for manned aircraft with the Air Traffic Management (ATM). UAS are expected to operate within Very Low-Level (VLL) airspace and over densely populated urban environments, where traditional ATM may encounter challenges in effectively managing such air traffic. This results in the formulation of a new airspace concept. The emerging concept is referred to as UAS Traffic Management (UTM), known as U-space within the European Economic Area (EEA)[1]. This initiative aims to implement new services and regulations for the management of UAS flights while ensuring safety, security, and operational efficiency.

The European Union Aviation Safety Agency (EASA) has laid down the regulatory framework for U-space. Regulation 664/2021 [1] establishes the basis for U-space, defining what constitutes U-space airspace and setting the requirement to utilise a minimum of four 'U-space services' (USSs) for executing UAS operations within it. The fundamental bundle of U-space services includes a Network Identification Service, a Geo-Awareness Service, a Flight Authorisation Service, and a Traffic Information Service. Additionally, the competent authorities may request to incorporate two other services: Weather Information and Conformance Monitoring, depending on the outcomes of an Airspace Risk Assessment (ARA) also set out in [1]. The FAS is defined as the service responsible for granting authorisation for each UAS flight and establishing the terms and conditions governing the operation. To this end, UAS operators shall submit flight authorisation requests (FAR) to the FAS. The authorisation must encompass the intended flight trajectory represented as a series of 4D volumes in addition to other relevant information. Prior to the approval of the FAR, the FAS shall ensure that the intended trajectory of the UAS is not in conflict with previously approved FARs, i.e., it does not overlap with them in either time or space.

A number of research projects are currently receiving funding from the SESAR 3 Joint Undertaking, which serves as the technological foundation of the European Union's Single European Sky initiative. These projects are aimed at the

This research was conducted within the framework of the SPIRIT-UCS project, which has received funding from the MCIN/AEI/10.13039/501100011033 under the reference PID2022-141829OB-I00.

[1]The EEA includes the European Union, Switzerland, Norway, and Iceland

development and implementation of U-space. In accordance with the integrated framework of the SPATIO [2] project, we employ the term 'strategic conflict' to characterise pre-flight conflicts. This terminology differentiates these conflicts from 'tactical conflicts', which refer to scenarios involving the loss of separation between aircraft in flight. Moreover, the International Civil Aviation Organization (ICAO) UTM framework [3] introduces a range of services, among which is the Strategic Deconfliction Service, aimed at enhancing safety by minimising the potential for mid-air collisions.

In this paper, we present an approach to a Strategic Conflict Detection concept with a centralised service enabling the efficient detection of strategic conflicts by more than one U-space Service Providers (USSP) in U-space airspaces with medium to high density of operations, enabling the subsequent resolution of any potential strategic conflicts between flights. To achieve this, we have developed a strategic conflict detection algorithm that ensures safe and coordinated operations within the airspace, thereby minimising the risk of mid-air collisions. We have based the concept presented herein on the existing regulatory requirements and their guidance material AMCs to Reg. 664 [4] and drew inspiration from existing standards referred to as guidance material therein, such as ASTM F3548-21 [5], which sets out new services and protocols for UAS Traffic Management (UTM). One of the key elements in the ASTM F3548-21 standard relevant to our research is the Discovery and Synchronization Service (DSS). The DSS enables U-space Service Providers (USSPs) to identify other providers within a given U-space they are interested in (e.g. because they have information relevant to their operations). According to ASTM F3548-21, *the DSS encapsulates an airspace representation into which entity references are mapped.* Examples of entity references are the 4D volumes defining UAS's intended trajectories. According to the same standard, *such airspace representation can be conceptualised as a grid, and mapping an entity reference into the airspace representation determines what grid cells it intersects.*

Drawing inspiration from that airspace representation set out in the ASTM Standard and meeting the regulatory requirements, we have defined a new service named Search For Overlap Service (SFOS). The SFOS is a supporting service that utilises the octree as a data structure to build the airspace representation for our U-space Service implementation. The octree divides a three-dimensional space by repeatedly splitting it into eight smaller regions called 'octants'. This structure was chosen among other possible structures due to its efficient search capabilities and adaptable data handling [6].

We have proven the concept outlined in this paper by means of simulations in several scenarios. Such scenarios feature progressively increasing number of UAS flights and different types of trajectories in order to evaluate the algorithm's performance under varying traffic density conditions.

The rest of the paper is structured as follows. In Section II, we introduce the foundations of our concept, emphasising Flight Authorisation Services, trajectory definition, and the Search For Overlap Service. In Section III, we outline the key assumptions that guide our approach and a comprehensive analysis of the methodology used to instantiate our Strategic Conflict Detection concept, offering an in-depth description of its implementation. In Section IV, we present the simulation results, while in Section V we present the final remarks and a prospect on future directions.

II. Background

In this section, we provide a comprehensive review of the existing regulatory frameworks and standards relevant to U-space within the context of Strategic Conflict. Additionally, we address the initiatives pertaining to Unmanned Aircraft System Traffic Management (UTM) and review ongoing efforts regarding the topic.

A. Flight Authorisation Service

The Flight Authorisation Service (FAS) set out in Reg. 664 [1] processes authorisation requests on a *first come, first serve*[2] basis when flights have the same priority. The service also includes verifying that requests are complete, correct, and conflict-free. If the request is incomplete or incorrect, it is rejected, and the operator is notified.

The CORUS-XUAM ConOps [7] proposes a structure of the FAS, which includes the utilisation of Geo-awareness service and Weather Information service as services used in the granting authorisation process. It also accounts for detecting conflicts, which leads to a Strategic Conflict Prediction Service that can be invoked by the FAS. To check if the flight is in conflict, the trajectories of the flights already accepted are compared to the new or updated one. Hence, a definition of trajectory is needed as it plays an essential role in deconfliction.

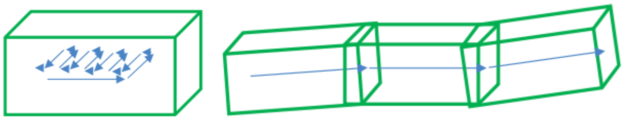
Project SPATIO [2] defines U-space services for the pre-flight phase, including Strategic Conflict Detection as part of the Strategic Conflict resolution service. The service is directly related to the Flight Authorisation Service defined in Reg. 664. The Strategic Conflict Resolution aims to prevent loss of separation by detecting a conflict through Strategic Conflict Detection and offering alternative operations once detected, which relates to the resolution aspect. SPATIO also assumes that authorisation requests contain the trajectory planned specified as 4D linear segments and/or volumes.

B. Trajectory

In Reg. 664 [1], the UAS' trajectory is defined as a series of 4D volumes, including space and time, as both are needed for the deconfliction of the authorisation. In the ConOps from SESAR CORUS-XUAM Project [7], there are two types of trajectories defined, shown in Fig 1.

The linear 4D trajectory is defined by prisms associated with an entry and exit time, while the area-based defines a single prism with an entry and exit time. To increase the

[2]The first to request authorisation for a flight of the same priority is the first to obtain authorisation if there is no conflict with an already accepted flight plan



(a) Area-based (b) Linear

Figure 1. 4D trajectories described in ConOps [7]

level of safety achieved through the strategic deconfliction of flight plans, the Regulation establishes a deviation threshold additional to the 4D volume [4]. The deviation threshold defines the volumes used to detect conflicts (i.e., a strategic conflict will consist of the overlap in space and time of the 4D volumes enlarged by the deviation thresholds). Despite its relevance for detecting strategic conflicts, the deviation threshold value has not yet been defined.

On the other hand, ASTM 3548-21 [5] defines an operational intent as the representation of airspace and time intended to contain the flight. This operational intent is defined by the horizontal and vertical bounds of the corresponding volume and its entry and exit times, as in the ConOps [7]. These volumes must account for the uncertainties due to the UAS performance and operational conditions. Considering this, the ASTM defines this volume based on the Total System Error (TSE) from the outer boundary of the operational intent for area-based trajectories or from the nominal path in the case of linear trajectories, whereas a time buffer is applied to the entry and exit times for the time component, reflecting errors that will result in time uncertainty.

According to these definitions, a conflict occurs when a volume in a given operational intent overlaps with another volume comprised of an already accepted one.

C. Discovery and Synchronization Service

ASTM 3548-21 [5] assumes a multi-USSP environment, raining the need to enable interoperability among different service providers. The ASTM Standard uses the Discovery and Synchronization Service and data exchange protocols (APIs) to achieve the required interoperability.

The DSS is a standardised discovery mechanism that allows USSPs to identify other USSPs for necessary data exchange while ensuring USSPs use current and consistent entity data. This service encompasses an airspace representation into which ‘entity references’ are mapped. ‘Entity’ refers to data types that must be shared between the USSPs, e.g., volumes in which the USS is interested. The clients do not have access to the implementation details of this airspace representation. The concept is that the airspace is a 3D grid, and once the USS sends the entity reference, the DSS determines what grid cells are intersected by the entity.

An example of the operation of the DSS is when a given USSP (e.g., USSP1) is interested in a 4D space for its UA operation (operational intent), so it sends the entity with the data to the DSS. The DSS identifies if the desired area is already occupied by other USSPs. Under those circumstances,

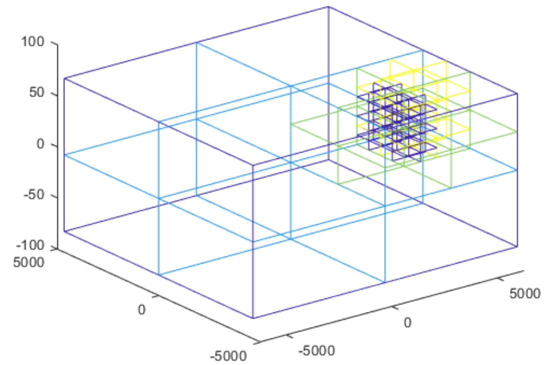


Figure 2. Octree example

the DSS provides the USS1 with the ID of the other USS to contact them so they can negotiate to solve the conflict.

The data exchange protocol utilised for the communication between USS-DSS in the ASTM 3548-21 uses an API (OpenAPI) described in one of the annexes to the standard. The specification defined in that annex contains the main methods for data exchange; POST, PUT, GET, DELETE. These methods will be used to create, update, retrieve or delete data such as the operational intent the USSP is planning or the creation of the subscriptions to the cells of interest. Those data are stored in a database named *DSS pool*, in order to access to them via the OpenAPI.

The implementation of ASTM is not aligned with U-space regulation, as it does not contemplate negotiations between different USSPs. Therefore, we have developed a solution drawing inspiration from the DSS that enables each USSP to declare a conflict based on the information provided by our SFOS. The SFOS constructs an airspace representation of the airspace using an octree data structure, which enables it to identify any overlapping volumes in time and space.

D. Octree

In Fig. 2 an octree data structure division is shown, in order of clarification for the explanation. The octree is made up of nodes that recursively subdivide into octants. This means each parent node is divided into 8 children nodes, with each octant representing a portion of a three-dimensional space in a cube-shaped form. The root is the first cube created and it represents the whole 3D space occupied by the octree.

These octants can, in turn, be further divided into eight smaller children octants if necessary. The division goes on until the minimum cell size (which is a parameter that needs to be fixed when the octree is created) is achieved and it is only made when required. That meas that when we search for cells within an area of interest, if a parent cell is not in that area it is not divided into children cells.

III. Formulation and implementation of the concept

In this section, we set out the proposed concept of a U-space Strategic Conflict Detection with a centralised service and instantiate it by means of a prototype developed in C++. Although the concept mainly addresses the strategic conflict detection requested by the flight authorisation service, the U-plan Preparation Service [8] is used to create the FARs used for the detection of conflicts.

A. *Assumptions*

As U-space is still under development, there is no common agreement for the formulation of critical data such as trajectories, volumes, flight requests, and other relevant information. Moreover, the FAS set out in Regulation 664 shall perform a number of functions beyond the detection of strategic conflicts. Therefore, we had to make some assumptions to keep the formulation of our ConOps for strategic conflict detection as simple as possible for the sake of clarity. We state and justify those assumptions in the paragraphs below.

The scope of the FAS set forth by Reg. 664, as well as the one of the SCD service set out in ASTM F3548-21, include both conflicts between UAS operations and between airspace constraints (e.g. UAS geographical zones designated within a particular U-space airspace, or the portions of a U-space airspace deactivated because of a dynamic airspace reconfiguration as per Reg. 664). Although the extension to detect infringements of airspace constraints is straightforward, for the sake of simplicity, we have limited the scope of our ConOps to strategic conflicts between UAS operations. Reg. 664 and its AMC&GM describe UAS in terms of trajectories defined by 4D volumes, whereas ASTM F3548-21 uses the expression ‘operational intent’, also defined by the aggregation of 4D volumes. In the context of the work presented in this paper, we have considered that 4D trajectories and operational intents are equivalent and used interchangeably.

For the sake of simplicity, we consider that all the flight authorisation requests have the same priority.

Contrary to what describes the standard ASTM F3548-21 and in line with Reg. 664, the FAS does not enable communication between USSP so that they can agree on how the conflict can be solved but simply indicates the 4D volumes overlap with previously approved requests. The Strategic Conflict Detection service, functioning as an internal microservice within the FAS, determines whether the overlap results in a conflict.

Although there is nothing in the ConOps that we present in this paper precluding the use of a distributed service in charge of the detection of conflicts, we assume a centralised one for the sake of simplicity. This approach is meant to reduce the synchronisation problems that may arise from communication between the USSPs and other agents.

Moreover, our airspace representation for the SFOS will be built on the octree structure because of its potential to speed up the deconfliction process that we propose and implement using parallel computing.

B. *Concept formulation*

The main steps of the process are (as depicted in Fig. 3): (1) UAS operators define the trajectory they intend to fly by means of a set of waypoints and their time of overfly; (2) the UPP service, taking into account the UAS performance and some local conditions, builds 4D trajectories composed of a series of 4D volumes as small as possible but commensurate with the flight uncertainty; (4) the UPP service creates U-plans including the 4D trajectory; (5) The FAS receives U-plans from the UPP and checks that they are complete and correct; (6) if the FAS detects any issue in a U-plan that might

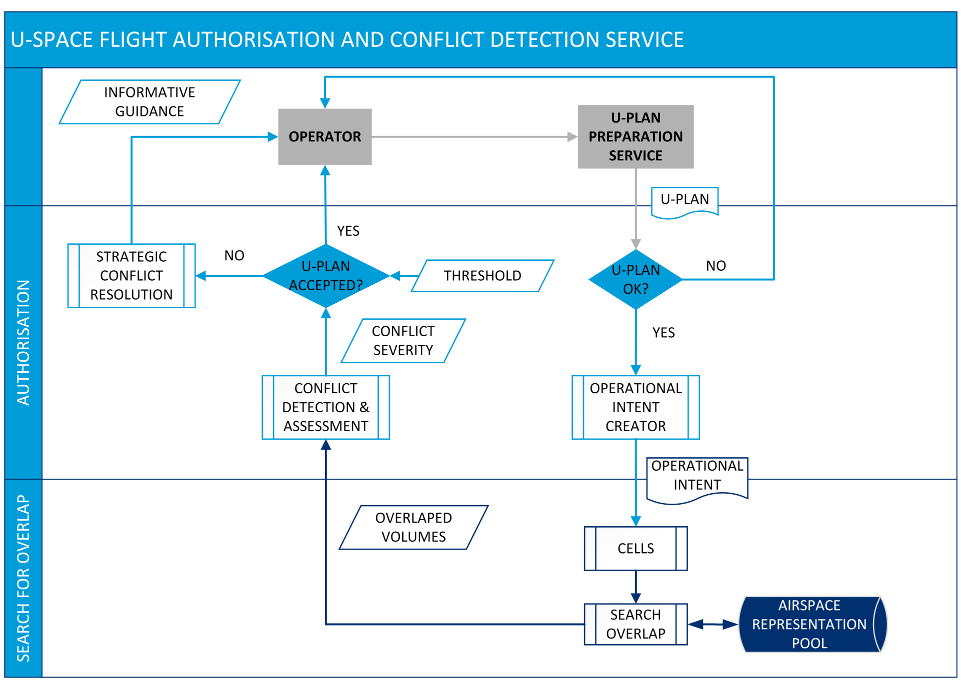


Figure 3. Operational view of the Strategic Conflict Detection

provoke errors in the subsequent processes, it rejects the FAR and notifies the UAS operator, including in the notification the reason of the rejection; (7) if the U-plan is correct, the FAS creates Operational Intents (OI) by enlarging the volumes in the 4D trajectory to achieve a given level of safety and assigns a unique ID to them; (8) the FAS sends the corresponding OI to the SFOS; (9) The SFOS stores the OI in the its database (Airspace Representation pool) and searches for the cells of the octree structure that intersect it; (10) the SFOS checks whether the octree cells that intersect the OI have already been assigned to another OI (i.e., whether the U-plan that is being processed overlaps in space and time previously approved U-plans); (11) if there is no overlap, the SFOS notifies the FAS that that the submitted OI is free of overlap; (12) if there is any overlap, the SFOS notifies the FAS that the submitted OI is in conflict and provides the identification of the volumes overlapping with previously accepted operations. The following paragraphs provide insight into the main steps of the process.

1) Creation of FARs: The operator sends the desired waypoints and the information the FAS requests through the UPP interface. The relevant information for the purposes of our deconfliction algorithm (besides the trajectory waypoints and associated times) encompass the UAS type (multirotor/fixed wing), the operation category (according to Reg. 947 [9]) and class (according to Reg. 945 [10], when applicable).

The UPP service transforms these waypoints into a 4D trajectory with associated entry and exit times. The computation of these waypoints into trajectories is explained in detail in [8]. However, it uses PX4 as the trajectory simulator and creates the 4D volumes for these trajectories with associated entry and exit times. The basic 4D volume is the ‘flight geometry’ (according to ASTM F3548-21), whose size depends on the flight uncertainties. Also according to ASTM F3548-21, such uncertainties can be characterised by the so-called Total System Error (TSE), which is defined both in the vertical (TSE_V) and horizontal (TSE_H) dimension (no distinction is made between along-track and across-track directions). The TSE defines the volume which contains the UA with 95% confidence and is driven by the Flight Technical Error (FTE), which depends on the ability of the remote pilot/autopilot to stick to the intended trajectory, and by the System Navigation Error (SNE), which represents the error in the position delivered by the UAS navigation system to the remote pilot/autopilot. The FTE depends on the UAS type (multirotor/fixed wind) and the local weather conditions (especially wind gusts), whereas the SNE is mainly driven by the Global Navigation Satellite System (GNSS) receiver used by the navigation system. We have taken the values for the FTE from EUROCAE ED-301 [11], while the NSE can be regularly updated using reports from the GNSS managers (e.g., Galileo Quarterly Performance [12]). We assume that the applicable FTE and NSE values for fixed-wing and multirotor UAS, both in nominal and non-nominal (e.g., strong wind gusts or degraded GNSS performance) conditions are part of the U-space definition made by the Competent Authority when the U-space airspace is designated and are available to the UPP from the Common Information Services (CIS). Therefore, when the UPP service receives a new U-plan, it gets the applicable TSE value and computes evenly spaced points along each trajectory segment. Then the FAS builds the 4D trajectory by creating flight geometry volumes centred at each of those points.

2) Detection of conflicts between trajectories: Flight authorisation requests (or U-plans) submitted by the UPP service must adhere to specific formatting rules[3]. Hence, before any conflict detection occurs, the FAS reviews the submitted U-plans to check whether all required fields are filled out correctly and the data are in the correct format. This validation step is crucial to prevent errors during the subsequent processing stages.

The next step in the deconfliction process is the creation of OI. While 4D trajectories are composed of flight geometry volumes built in light of the flight uncertainties, we use larger volumes defined around them (called ‘protection volumes’) for the detection of strategic conflicts, pursuing that the strategic conflict detection is compelling enough to achieve the applicable TLS (e.g., the deviation thresholds defined in [4]). We have defined protection volumes with higher containment confidence levels considering an initial value of 99% of containment confidence to define the protection volumes, although lower (or higher) figures might be needed depending on the local implementation of the U-space (e.g., the use of tactical deconfliction services or the expected demand). We consider that the size of the protection volume relative to the flight geometries is a property of the U-space airspace made available to the USSPs through the CIS. After the preliminary checks for completeness and correctness performed by the FAS on the U-plans, the FAS builds the protection volumes around the flight geometry volumes, transforming the 4D trajectory into an operational intent (Fig. 4 depicts an example of 4D trajectory and the corresponding OI).

Once the OI, corresponding to the 4D trajectory in the received U-plan, is created, the FAS sends the OI to the SFOS. The SFOS first checks the OI for correctness and accuracy. After those checks, the SFOS stores the information in the Airspace Representation pool and searches for overlaps in time and space. To this end, the SFOS first determines which cells of the SFOS airspace representation grid intersect the 4D volumes in the OI. For this purpose, we use the octree data structure. When the octree is initialised, its root represents the entire airspace where the service is being implemented. When the SFOS receives an OI, it splits the octree cells that intersect the OI protection volumes until the minimum size is achieved unless those cells have already been created when processing a previous OI. Then, the SFOS checks whether these cells are already occupied in time by other flight plans submitted earlier.

If the SFOS finds no overlap, it notifies the FAS that there are no 4D volumes with overlap. However, if an overlap

[3]The data format selected is JSON with the fields specified in the [4]. It includes the volumes defined as GeoJSON format [13].

is detected, the SFOS reports to the FAS the IDs of the protection volumes affected by the conflict, and the SCD service determines if the conflict exists. Due to data protection policies, the SFOS does not disclose the identity of the USSP responsible for the conflicting plan.

C. Concept implementation

In order to prove the concept outlined in the previous subsection, we have used different software tools. As we have described above, our strategic conflict detection process has three elements: the UPP, the FAS and the SFOS.

We have built the UPP by adapting trajectories obtained in PX4 by applying the TSE in MATLAB for its simplicity in managing data. We have also created a JSON file in the format of the U-plans generated by the tool, which is the format needed by our FAS.

The FAS (including the SCD service) and the SFOS (including the octree generation and search algorithms) were initially developed in MATLAB because of its simplicity. However, MATLAB functions were not compiled software, and its computational performance (in terms of execution times) was relatively poor. For this reason, we decided to migrate the code to C++ to optimise the execution times. Additionally, we implemented parallelisation for the function responsible for identifying the octree cells occupied by the volumes.

On the other hand, we have implemented the interfaces between the different entities using APIs developed in Python. For this purpose, the main method implemented is the POST method, which consists of the creation of the request or the OI, respectively. The APIs also perform checks to ensure that the U-plan or OI is formatted correctly and the data are accurate before accepting the transmission from the operator or FAS.

IV. Results

To test our proposed Strategic Conflict Detection concept, we computed our trajectories and created volumes around them to develop our 4D trajectories. We constructed both linear and area-based trajectories to evaluate the influence of these two types within our tool. Finally, we obtained execution time results to assess the efficiency of the tool. The results obtained in this paper are presented in this section.

A. Trajectories

In this section, we present the UAS trajectory defined by volumes that represent the possible deviation during flight. Figure 4 illustrates the final operational intent trajectory along with the associated volumes. This figure highlights the defined flight geometry and the protection volume that has been applied.

B. Scenarios

Several scenarios have been defined to test the capabilities of the strategic conflict detection algorithm. An integration area for the service has been established over the port of Valencia, covering an area of approximately 22.5km^2.

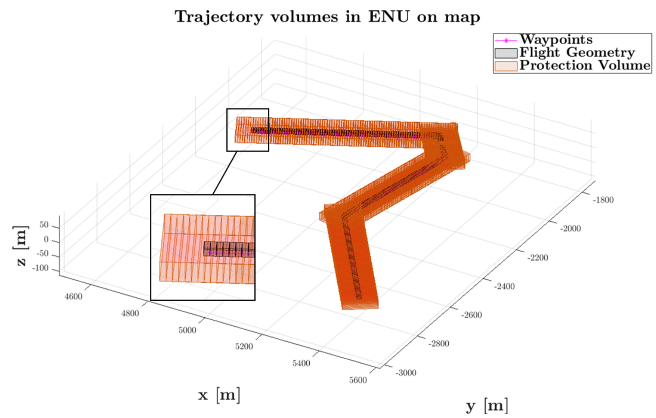


Figure 4. 4D trajectory and protection volumes example

In this area, we simulated various scenarios involving an increasing number of flight authorisation requests. Table I outlines the different scenarios. The flights are defined to take off within a one-hour time frame and the waypoints have been generated randomly.

TABLE I. CASE SCENARIOS DEFINED

Case	Number of flights
1	10
2	25
3	50
4	100

We have also included three area-based trajectories to evaluate their impact on the tool. These trajectories were incorporated randomly during the execution of the scenarios described earlier. These trajectories are also to take off within a one-hour time frame, and their duration can vary between 15 and 30 minutes.

Fig. 5 presents scenario 3 with the area of the port in red and including the three area-based trajectories in different colours, comprising a total of 53 flights.

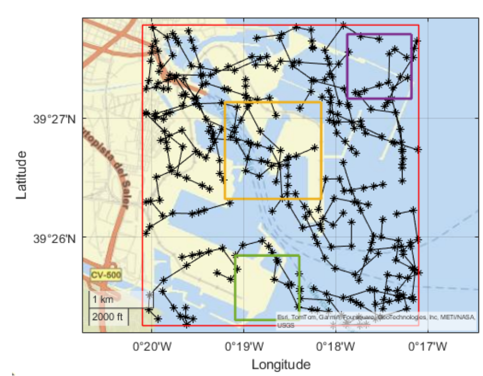


Figure 5. Scenario 3 with three area-based trajectories represented on map

C. Detection of conflicts

We conducted measurements of execution time across the proposed scenarios to evaluate the time required for conflict detection. In Table II, we provide the time taken to identify overlaps in each scenario, the overall time for the scenarios, the average time per flight along with its deviation for each scenario, and the total number of conflicts identified.

TABLE II. RESULTS FOR CONFLICT DETECTION WITH LINEAR TRAJECTORIES

	Overall Time [min.]	Average Time [min.]	Conflicts
Scenario 1	8.86	0.89 [0.26]	0
Scenario 2	25.26	1.01 [0.32]	0
Scenario 3	53.02	1.10 [0.53]	1
Scenario 4	173.98	1.76 [0.79]	3

In Table III, we present the same results, including the area-based trajectories.

TABLE III. RESULTS FOR CONFLICT DETECTION WITH LINEAR AND AREA-BASED TRAJECTORIES

	Overall Time [min.]	Average Time [min.]	Conflicts
Scenario 1	8.64	0.66 [0.40]	1
Scenario 2	24.88	0.89 [0.45]	1
Scenario 3	53.71	1.03 [0.60]	4
Scenario 4	163.84	1.62 [0.77]	7

The results of our conflict detection analysis provide valuable insights into the performance of our algorithm across different scenarios. For linear trajectories, the overall times increase with scenario complexity. The deviation in average flight times also increases with scenario complexity, indicating more significant variability in detection times as the scenarios become more intricate. Notably, no conflicts were detected in the simpler scenarios, while more complex scenarios identified up to three conflicts in the scenarios with more traffic.

In contrast, when we consider area-based trajectories, the overall and average flight times are lower due to the reduced number of associated volumes. While area-based trajectories involve only a single, albeit large, volume, linear trajectories can vary from 200 to 400 volumes. The deviation in average flight times for area-based trajectories also increases with complexity but remains generally lower than linear trajectories.

This comparison underscores the robustness of our algorithm and its ability to adapt to different conditions. These findings not only validate the current performance of our algorithm but also offer valuable insights for future enhancements, aiming to improve efficiency and scalability further.

To better understand the results obtained, we have detailed the average of the execution times on the different phases of our SFOS. The term “Find” refers to the time to locate the cells occupied by the 4D volumes within the airspace representation. “Times” indicates the processing time to associate these cells with their entry and exit times. “Check” signifies the time required to verify whether the detected cells are already occupied, while “Insert” denotes the time needed to insert the newly identified cells when no conflicts arise. Table IV shows each scenario’s results.

In Table V, we show the results when including the area-based trajectories.

The “Find” and “Times” steps for linear trajectories remain relatively consistent across scenarios. In contrast, the “Check” step time increases significantly with scenario complexity,

TABLE IV. TIME ANALYSIS FOR SCENARIOS WITH LINEAR TRAJECTORIES

	Find [s]	Times [s]	Check [s]	Insert [s]
Scenario 1	11.31	7.23	4.47	30.14
Scenario 2	11.08	7.22	12.82	29.50
Scenario 3	9.96	6.45	23.46	26.40
Scenario 4	11.04	6.80	58.86	28.75

TABLE V. TIME ANALYSIS FOR SCENARIOS WITH LINEAR AND AREA-BASED TRAJECTORIES

	Find [s]	Times [s]	Check [s]	Insert [s]
Scenario 1	8.39	4.80	4.53	22.15
Scenario 2	9.75	5.77	12.50	25.29
Scenario 3	9.55	6.23	22.01	24.18
Scenario 4	9.97	5.61	55.49	26.26

reflecting the added effort required to check with trajectories already accepted. The “Insert” step consistently takes the longest time.

Meanwhile, area-based trajectories show lower times for each step. The “Check” step time also increases with complexity but remains generally lower than that of linear trajectories and the “Insert” step continues to be the most time-consuming.

Despite the “Check” and “Insert” steps taking the longest time in the conflict detection process, the “Find” and “Times” steps demonstrate excellent performance with the implementation of the octree. The octree structure significantly enhances the efficiency of these steps by enabling faster data retrieval and processing. This improvement is evident in both linear and area-based trajectories, where the times for the “Find” and “Times” steps remain relatively low and consistent across different scenarios.

To study the difference between the linear and area-based trajectories times in the “Find” and “Times” steps, we have normalised the execution time with the number of volumes for Scenario 1. In Table VI we present the average time for each time of trajectory and the normalised with its associated number of volumes.

TABLE VI. AVERAGE TIME FOR “FIND” AND “TIMES” STEP IN SCENARIO 1

	Find [s]	Normalised [s]	Times [s]	Normalised [s]
Linear	10.885	0.033	6.243	0.018
Area	0.054	0.054	0.001	0.001

For the linear trajectories, the “Find” step takes less time per volume than the area-based as the volumes are smaller than the area-based so the number of cells funded per volume are lower. In contrast, for the “Find” step the area-based trajectories consume less time as all the cells identified will have the same entry and exit times. This analysis highlights the efficiency achieved by our SFOS through our implementation with the octree and its parallelisation.

The results of our conflict detection analysis demonstrate that the algorithm is both robust and adaptable across various scenarios. we observe that both overall and average flight times increase with complexity, along with greater variability in detection times. As we detailed in the execution times for each step, this increase can be attributed to the "Check" step, where having more previously approved trajectories results in more cells to check in our database. Additionally, the breakdown of execution times highlights the efficiency of the "Find" and "Times" steps, particularly due to the implementation of the octree and its parallelisation. These findings validate the current performance of our algorithm and provide valuable insights for future enhancements, aiming to improve efficiency and scalability further.

V. Conclusions & future works

In this paper we present the formulation, implementation and proof of an algorithm for Strategic Conflict Detection within U-space, addressing the need for effective deconfliction and ensuring safe operations. Through simulations that increase operational demand in a predefined airspace, we have demonstrated that our algorithm can effectively detect strategic conflicts while complying with regulatory requirements. By incorporating the concept of an octree as our 'SFOS airspace representation,' our algorithm features a robust search mechanism for identifying relevant cells.

Nonetheless, we have identified several improvements that we intend to developed in the near future.

First, the insertion of data into the database may be parallelised to reduce the execution time. For the "check" step time, a parallelization could also be included to minimise the increase of execution time as traffic grows.

Another important step forward is the incorporation of the Common Information Service (CIS) to provide information about potential airspace constraints. This will enable us to detect not only conflicts between UAS trajectories but also conflicts related to geozones with specific constraints or protected areas.

Furthermore, the FAS could be improved by establishing a Strategic Conflict Resolution service. This service would use the information from the Strategic Conflict Detection service to provide suggestions for alternative flight routes or timing adjustments, helping UAS operators obtain the necessary flight authorisation.

Lastly, stochastic deconfliction mechanism, in which a U-plan can be accepted even when there is a given level of overlapping pursuing a better use of the airspace, especially when tactical conflict resolution methods are employed.

References

[1] EASA, "Commission Implementing Regulation (EU) 2021/664," 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0664

[2] SESAR Project, "SPATIO - U-space Separation Management," 2023. [Online]. Available: https://www.sesarju.eu/projects/SPATIO

[3] ICAO, "Unmanned Aircraft Systems Traffic Management (UTM) –A Common Framework with Core Principles for Global Harmonization," 2022. [Online]. Available: https://www.icao.int/safety/UTMframework/Pages/default.aspx

[4] EASA, "Acceptable Means of Compliance and Guidance Material to Regulation (EU) 2021/664 on a regulatory framework for the U-space," 2022.

[5] ASTM, "F3548–21, Standard Specification for UAS Traffic Management (UTM) UAS Service Supplier (USS) Interoperability," 2021.

[6] D. P. Mehta and S. Sahni, Eds., *Handbook of Data Structures and Applications*, 1st ed. Chapman and Hall/CRC, Oct. 2004.

[7] EUROCONTROL, "U-space ConOps and architecture (edition 4)," SESAR Project CORUS-XUAM D4.2 Ed: 01.00.02, 2023. [Online]. Available: https://www.sesarju.eu/node/4544

[8] Alex Sanchis, Sandra Amarillo & Juan V. Balbastre, "Realistic trajectory generation in simulated environments for U-space systems assessment," ICNS 25, 2025, to be published.

[9] EASA, "Commission Implementing Regulation (EU) 2019/947," 2019. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R0947

[10] EASA, "Commission Delegated Regulation (EU) 2019/945," 2019. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R0945

[11] EUROCAE, "ED-301 - Guidelines for the Use of Multi-GNSS Solutions for UAS Specific Category - Low Risk Operations SAIL I and II," 2022.

[12] EUROPEAN UNION, "EUROPEAN GNSS (GALILEO) SERVICES –QUARTERLY PERFORMANCE REPORT. JANUARY–MARCH 2024," 2024.

[13] Butler et al., "The GeoJSON Format," 2016. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7946