A Framework for Operational Volume Generation for Urban Air Mobility Strategic Deconfliction

Ellis L. Thompson[1] and Yan Xu[2] and Peng Wei[3]

Abstract— Strategic pre-flight functions focus on the planning and deconfliction of routes for aircraft systems. The urban air mobility concept calls for higher levels of autonomy with onboard and en route functions but also strategic and pre-flight systems. Existing endeavours into strategic pre-flight functions focus on improving the route generation and strategic deconfliction of these routes. Introduced with the urban air mobility concept is the premise of operational volumes. These 4D regions of airspace, describe the intended operational region for an aircraft for finite time. Chaining these together forms a contract of finite operational volumes over a given route. It is no longer enough to only deconflict routes within the airspace, but to now consider these 4D operational volumes. To provide an effective all-in-one approach, we propose a novel framework for generating routes and accompanying contracts of operational volumes, along with deconfliction focused around 4D operational volumes. Experimental results show efficiency of operational volume generation utilising reachability analysis and demonstrate sufficient success in deconfliction of operational volumes.

I. INTRODUCTION

Urban Air Mobility (UAM) concept is generally described in [1]–[3] as: a safe and efficient, highly automated, aviation transportation infrastructure, for passengers and cargo, operating at low altitudes. The highly automated nature of this concept favours emerging technologies for aircraft, such as Unmanned Aircraft Systems (UASs) and electric Vertical Take-Off and Landing (eVTOL) vehicles, as well as Air Traffic Management (ATM) and route planning systems. The European Aviation Safety Authority (EASA) predicts UASs to become prevalent in European cities by 2025 [2], and the Federal Aviation Administration (FAA) estimates 828 thousand unmanned commercial aircraft by 2024 [4].

The prediction of high-density UAM airspace brings with it the requirement for both highly automized aircraft and ATM techniques. Introducing autonomy at the strategic pre-flight, level can provide efficient route planning and deconfliction for aircraft wishing to enter the airspace. Additionally, autonomous aircraft and ATM systems can provide tactical instructions to ensure the airspace remains safe with efficient traffic flow.

A UAM operator is introduced as an entity responsible for the management of UAM specific operations [5]. A UAM service provider therefore is any entity offering services to assist UAM operations. One such role that either an operator or service provider is responsible for is route planning and sufficient deconfliction for safe and efficient flight. The concept of four-dimensional (4D) regions of airspace, including time, is presented as Operational Volumes (OVs) [6], [7], with the intention that those operating aircraft remain within their designated OVs for the duration of the flight. This consequently becomes an agreed upon *contract* between the aircraft operator and airspace, of where the aircraft is allowed to operate and for what duration.

In this paper we propose a framework for generating deconflicted routes as well as 4Ds OVs over the routes. Motivated by the work presented in SkyTrakx [8] we introduce route planning and deconfliction to provide initial candidate routes for the OVs to be generated over. We additionally expand on their definition of an OV, resolving assumptions around OV boundary arrival times which could potentially cause inaccuracies in the temporal element of OVs. We further expand on the definition of an OV given in [8] by providing a definition that is sufficiently more adaptive to an uncertain airspace environment, by including the distribution of simulated aircraft in our modified definition. To build the OVs our approach leverages the reachability analysis framework identified in [9]. To generate the routes for OV generation we propose a rapidly-exploring random tree (RRT) based algorithm focusing on the cruise phase of flight.

The remainder of this document is formatted as following: Section II covers related works, we then formally define both a *contract* and *operational volume* and provide our implementation for generating OVs in Section III. The approach used for route generation is presented in Section IV. Experimental results are covered in Section V, followed by a discussion on the proposed framework in Section VI. Finally, concluding remarks are provided in Section VII.

II. RELATED WORKS

A. Concept of Operations

Stakeholders in the field of UAM, including the FAA, NASA and Europe's SESAR Joint undertaking, provide outlines for the conceptual UAM requirements. Provided in the technical documents [6], [7], [10], [11], are outlines into the safe operating standards expected to be imposed in UAM airspace. In addition to the U-Space operating regions and requirements, such as defining maximum altitudes, one key area referenced heavily is the imposition of operating volumes unique to each flight within the U-Space. Defined as reserved regions of 4D airspace and often depicted as

[1]Ellis L. Thompson is with the School of Engineering and Applied Science, George Washington University, Washington, DC 20052, USA thompson_e@gwu.edu

[2]Yan Xu is with the Centre for Autonomous and Cyberphysical Systems, Cranfield University, Bedford, MK43 0AL, UK yanxu@cranfield.ac.uk

[3]Peng Wei is with the School of Engineering and Applied Science, George Washington University, Washington, DC 20052, USA pwei@gwu.edu

either linear monotonically progressing tubes or polygons, there is a heavy implication that, within the conceptual UAM U-Space, operators will be required to define their expected operating area pre-flight.

B. Route Planning

Route planning algorithms provide potential candidate routes from an origin to a destination. Firstly, shortest path algorithms, such as Dijkstra's algorithm, A* and D* [12]–[14], generate the shortest or, in the case of A* and D*, the minimum cost path between two nodes. Used heavily in the field of Computer Science, A* and D*, work to optimise some heuristic function, commonly distance, always expanding the node with the lowest heuristic cost. The route can be considered to be guaranteed optimal if an admissible heuristic is found, that is a heuristic function that never overestimates the cost of the shortest path. The inverse of this would be a non-admissible heuristic, where the optimality of the generated path cannot be guaranteed but in many cases a near-optimal path can be found in less time.

A second class of route planning algorithms can be derived from motion control in robotics. Included in this classification are, RRT based approaches as well as Probabilistic Roadmaps (PRMs) and Fast Marching Trees (FMTs) [15]–[20]. These approaches are often not optimal and largely work on the principal of paths between randomly generated points, meaning the solution paths are rarely reproducible. However, they have the benefit of, in less complex environments, regularly finding a path in less time than A* and D*.

C. Strategic Deconfliction

Strategic deconfliction [21], is specifically concerned with pre-flight stratagem to provide efficient use of the airspace while reducing the likelihood of en route conflicts. Encompassed in this is the act of route planning and re-planning with a greater emphasis on re-planning for deconfliction. Most strategic deconfliction approaches [22]–[25] predict the aircraft's position at a given time and compares that with the predicted position of other airspace users. If a conflict, or conflicts, are detected then one of two approaches are usually employed: as in [23], [24] delays can be imposed to the schedule proposing the aircraft departs later than its previously scheduled time. Alternatively, as in [22], [23] re-planning can occur to change the original filed route proposing either changes to segments or the entire flight plan.

Intuitively, it is often favoured to propose a new departure time over re-planning the route due to its simpler nature and implementation for an already existing route. The latter however can be useful if no prior route exists, that is, the route is strategically deconflicted at the time of generation.

D. Reachability Analysis

As mentioned, our proposed framework uses reachability analysis to build the OVs from simulation data. Reachability analysis is the process of computing reachable sets given some initial conditions usually accompanied by either simulation data or real data. DryVR [9] provides data-driven approach for verification of automotive systems. In this approach reachable sets are generated by simulation and accompanying reach tubes, conservative volumes of reachable states over time, are generated. The reach tubes are generated through learning a discrepancy function and the related sensitivities of the simulation-generated data in relation to the initial given state.

An alternative methods, identified in [26]–[28], propose approaches utilising Lagrangian-based methods, with [27], [28] providing an analytical approach to reach tube generation. The benefit of these methods over the globally learned discrepancy function of DryVR is that, unlike DryVR, these methods are less susceptible to the wrapping effect [29], overall producing tighter volumes for the reach tubes.

E. Operational Volumes, Geofences and SkyTrakx

SkyTrakx [8] proposes an approach to generating a contract of OVs for a route and specific aircraft type by performing a reachability analysis over a finite time horizon. The SkyTrakx framework demonstrates the application of reachability-based OV generation, for light UASs, for both rotor and fixed-wing aircraft. They further allow for two generation types of OVs: conservative and aggressive, with the latter producing tighter bounded OVs. However, assumptions are made about the arrival time of aircraft at OV boundaries, potentially causing the temporal element of the OVs to become inaccurate.

Dynamic geofences are also proposed in [30] for coordination of low-altitude UASs. This approach uses knows of the aircraft dynamics to build a three-dimensional (3D) volumes of airspace, referred to as a geofences, that the aircraft must visit in sequence. Unlike [9] the volumes generated here are conservative to the extremes of the aircraft performance. It could be expected however, that this approach is susceptible to inaccuracies, significantly so as there is little provision added for the complexity introduced by wind. Yet, [30] has considered the possibility for delays producing a slightly less stringent requirement, or rather assumption, that the arrival time at volume boundaries can be guaranteed, creating a possibly more adaptable approach.

In [31], *keep-in* geofences are generated in tandem with a route generation. Monte Carlo simulations were used to propose volumisation strategies. An overlap between volumes was proposed as part of the strategy to account for the UAS flying faster or slower than expected, this effectively creates a buffer region where the aircraft can inhabit the most suitable volume for a given time. For the route planning and deconfliction, the flight path was optimised against flight time, distance and energy usage. Additionally, the approach maintained a minimum buffer, defined by the user, around both buildings and other aircraft through the use of obstacle-free visibility graphs.

III. CONTRACTS AND OPERATIONAL VOLUMES

A. Defining Contracts and Operational Volumes

Before we can define what is meant by the term *contract* we must first provide a definition of an OV. We can initially describe an OV as: A 4D region of airspace that describes both the positional operational area and the duration over which the volume is considered active. As such, if we represent the airspace as $\mathcal{X} \subseteq \mathbb{R}^3$, then a single OV is represented as tuple in Equation 1.

$$\mathcal{C} = (R_1, T_1), (R_2, T_2), ..., (R_k, T_k) \quad (1)$$

Where, $R_n \subseteq \mathcal{X}$ is a subset volume of the airspace and T_n is monotonically increasing time. This initial definition is derived in [8], and provides the foundation for the expansion we will apply. It should also be noted that this definition of an OV consists of several smaller volumes over discrete time intervals, the reason for this will be explained further in this section. At present though we are able to define the total area of this finite time operational volume as Equation 2 and the total duration as $\mathcal{C}^{dur} = T_k - T_1$.

$$\mathcal{C}^{vol} = \bigcup_{n=1}^{k} R_n \quad (2)$$

The definition of a contract (Σ) can then be determined as: The sequence of OVs over a given proposed trajectory for a single aircraft:

$$\Sigma = (\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_i)$$

Additionally, we can now impose two further requirements into the definition of an OV: (1) For any pair of OVs, $\forall (\mathcal{C}_i, \mathcal{C}_j) \in \Sigma$, then $\mathcal{C}_i^{dur} = \mathcal{C}_j^{dur}$; (2) By extension the interval at which the volumes in $\mathcal{C}_i$ are generated must be regular both within $\mathcal{C}_i$ and $\forall \mathcal{C}_n \in \Sigma$.

It is at this point our definition begins to deviate from that given in [8]. Where the assumption is made that, for two successive OVs, $\mathcal{C}_i$ and $\mathcal{C}_j$ the following is true: $\mathcal{C}_i(T_k) \equiv \mathcal{C}_j(T_1)$, that is there exists a hard boundary between OVs such that they do not overlap temporally. In our system we introduce the constant δ such that now $\mathcal{C}_j(T_1) = \mathcal{C}_i(T_{k-\delta})$ where $\delta \in [0, \mathcal{C}^{dur})$.

We are now able to make one more change to the definition of an OV. For deconfliction of OVs some distribution of aircraft position must be known at a given time. As such we can modify the definition given in Equation 1, like so:

$$\mathcal{C} = (R_1, T_1, D_1), (R_2, T_2, D_2), ..., (R_k, T_k, D_k) \quad (3)$$

Here, we have introduced the value D_n, representing the distribution of aircraft positional data at each discrete time interval. Defined more concretely later, this value now allows for the probabilistic evaluation of the aircraft position at any given time within the bounds of an OV or contract.

B. Generating Operational Volumes

For a given, predefined route, OVs are generated using an adaptation of the DryVR [9] framework. Included within the DryVR framework is: "a probabilistic algorithm for learning sensitivity of the continuous trajectories from simulation data" as well as an algorithm that then uses these sensitivities to perform a reachability analysis and generate a reach tube. We leverage these algorithms by utilising simulation data obtained with the BlueSky [32] ATM simulator to build a contract of 4D OVs over a predefined trajectory.

In the generation of the OVs two factors need to be considered: the first, as outlined previously, OVs exist for a predetermined duration $\mathcal{C}^{dur}$ but, for a OV $\mathcal{C}_n, n > 0$, the start time is offset from by some fixed amount δ from the previous OV. The second factor, not inherently obvious, is the inclusion of uncertainty. While we are able to simulate the aircraft's trajectory initial factors, as well as variations in speed, acceleration, climb rates and wind are not known. As such a level of reasonable uncertainty needs to be maintained when running these simulations.

The generation of the simulated trajectory data and accompanying OVs are performed in three stages: (1) Uncertainty is introduced into the simulation during the initialisation of aircraft starting states; (2) The simulation is run, for some duration ($t_d = \mathcal{C}^{dur}$) with aircraft states being stored are regular intervals, for example 1 second; (3) The DryVR framework is used on a sample of output trajectories and then verified on the remaining trajectories to build the OV bounds. This three-step approach is repeated until the aircraft have reached the terminal waypoint in the simulation. The resultant OVs are then combined into a contract.

1) Initialising aircraft states with uncertainty: To introduce variation and uncertainty into the simulation N aircraft, between 500 and 2000 are created. For the first initialisation, i.e., from the origin, the position, altitude, heading and speeds are chosen at random from within predetermined bounds.

After each simulation run to t_d, N new aircraft are initialised in the system. Unlike the initial run however, these positions are sampled from the previous batch of trajectories, more specifically, they are sampled from the normal distribution of states from the offset interval $\delta \pm c$, where c is some time constant in seconds. This approach favours generating states closer to the existing *mean state* but provides uncertainty and variation. Additionally, to avoid potentially infeasible states from occurring, the samples are bound to lie either within the previous OV or no further than some fixed distance from the previous OV.

2) Collecting trajectory data from the simulation: From the initialised states the BlueSky simulation in run for the duration t_d. The step-size and logging interval are independent of each other and in our environment are set as 0.1 seconds and 1 second respectively. Two aircraft were used in our simulations: an Amazon octocopter available through the OpenAP [33] suite and the Airbus Vahana eVTOL vehicle using performance data available in [34], [35]. These aircraft provide two different examples of UAM space users to run

evaluations on, although, they are not used together when generating OVs.

For the N aircraft in the environment, N states were generated each second with each state storing: record time, latitude, longitude, altitude, heading, vertical speed (m/s), true airspeed (m/s). Noise was also added to the values in the state to further inject uncertainty, mimicking the effects of wind and uncertainty in GPS position.

The result of this second stage is the set of trajectories S^N for N aircraft taken at 1 second intervals from the simulation such that the total duration of the trajectories is equal to t_d.

3) Generating OVs from trajectories: The resultant trajectories (S^N) from the simulation are then used in the DryVR framework to perform reachability analysis and obtain a resultant reach tube. These reach tubes, along with the aircraft distribution data to satisfy Equation 3, form the set of OVs over the prescribed trajectory which form a contract.

The reachability analysis is performed in 3D space over the current and next states for each time step. Also, for all trajectories only the: record time, latitude, longitude and altitude are provided to the framework. Additionally, to avoid over or under estimation for trajectories with a small geographical footprint, the latitude, longitude and altitude are normalised to values in the range $[0, 1]$.

The DryVR framework is provided with a subset of 15-20 trajectories $S^X \subset S^N$. Additionally, the centre trajectory (S^X_{centre}) is provided and is used by DryVR to generate the sensitivities. The remaining set ($S^N - S^X$) is used for verification of the OV to ensure accuracy, or that the generated volume includes a sufficient number of points from the larger trajectory sample. If the verification is not satisfied then the route and accompanying OVs are not released and a re-plan with new parameters is required.

The output of the framework is a reach tube which, combined with the duration t_d and initialisation time (t_0), create a single OV. When this process is repeated, the OVs can then be combined into a contract spanning the proposed flight path as seen in Figure 1.
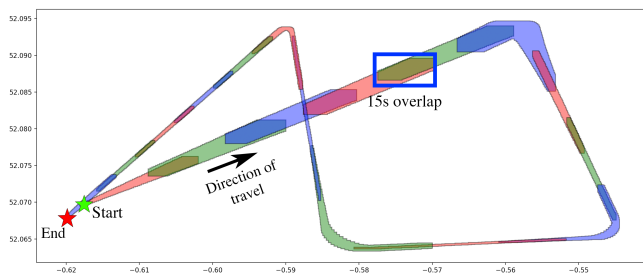

Fig. 1: A two-dimensional, top-down view of the 18 OVs in a contract. The duration (t_d) of each OV is 60 seconds and a 15 second offset has been used. Each OV has been given a color (red, green or blue) to easily distinguish them.

C. Distribution of Aircraft within OVs

As already emphasised, aircraft are expected to make constant progression along their route, derived from the finite time than an OV can exist. To ensure efficient use of the airspace when generating routes, the position of an aircraft within an OV can be represented as a distribution and then utilised in route planning to avoid regions of airspace with high likelihoods of aircraft existing.

Included in the definition of an OV in Equation 3, the distribution of aircraft at any time interval within the duration of an OV is given as D_n. Naturally, if we were to sample any given specific position $s \in \mathcal{C}(t)$ we would not expect to observe an aircraft presence due to the specific precision required. Instead, a grid is overlaid over the area of $\mathcal{C}$ creating a finite number of cells to sample the aircraft distribution over. We can now represent the cells of the grid over the OV as $\mathcal{C}_{i,j}$.

So for a specific query position s at time t within the duration of $\mathcal{C}$, and where the function $f(s) \mapsto (i,j)$ returns the relevant pair (i,j) corresponding to a cell coordinate in $\mathcal{C}$. We can now obtain the probability of an aircraft existing for the cell that s belongs to through using Equation 4, where N is the total number of aircraft in the simulation and a is a single aircraft.

$$P(s,t,\mathcal{C}) = \frac{1}{N}\left(\sum a \in \mathcal{C}_{f(s)}(t)\right) \quad (4)$$

IV. ROUTE GENERATION

A. An Overview of RRT*FND and RRT-Rope

For the route generation and adaptation of the RRT algorithm was chosen for the significantly faster execution speed in complex environments over algorithms like A*. The method leveraged uses a combination of: rapidly-exploring random tree fixed nodes - dynamic (RRT*FND) [17] for the initial candidate route generation, and its dynamic property allowing it to reconnect or rebuild disconnected trees, as well as RRT-Rope [18], which shortens and straightens out the winding routes generated by traditional RRT algorithms.

*1) RRT*FND:* Built on the *fixed node* concept of rapidly-exploring random tree fixed nodes (RRT*FN) [16], RRT*FND can generate routes between two points in an environment with obstacles that change over time.

RRT* firstly builds routes through randomly adding nodes into an environment and then re-wiring the tree such that the cost from the origin node to any other node is minimized, with respect to all the nodes in the tree. This is referred to as the `Grow` function and forms the base of RRT*FND. Operating identically to RRT*, RRT*FN introduces the concept of *fixed nodes* or, a maximum number of nodes (M) that can exist in the tree. The algorithm uses the `Grow` function until the number of nodes exceeds M where RRT*FN employs a force removal algorithm. The algorithm, in short, randomly selects a childless node to remove from the tree so that the number of nodes in the tree remains no greater than M. There is a possibility that the last node added exists on the solution path. To avoid removing this newly added node it is excluded from the random selection of childless nodes.

RRT*FND [17] improves RRT*FN for dynamic environments by utilizing existing information if the solution path is

severed. Introduced are the `Reconnect` and `Regrow` functions. If the existing solution path is severed, `Reconnect` attempts to reconnect the solution tree directly to the original parent tree. If this connection cannot be made, then `Regrow` attempts to connect the parent tree to any node in the solution tree by rerunning the `Grow` function.

2) RRT-Rope: RRT-Rope [18] shortens existing paths using a technique similar to that of pulling a rope. This was added to, in addition to shortening a path, provide a smoother path, easier for an aircraft to follow as RRT-Rope removes the frequent direction changes.

For each node in the solution tree, it looks at every subsequent node, starting with the furthest node first. It evaluates if a direct path has any collision with the obstacles in the environment. Providing no collision is found the two nodes will be directly connected moving intermediate nodes into a straight line between the two nodes.

B. Route Generation using RRT*FND and RRT-Rope

1) Pre-processing: To avoid errors over large distances when trying to use a constant interval with latitudes and longitudes, the airspace area was first converted to a local, flat Cartesian plane. This was performed using Equations 5-6 to obtain meters per degree of latitude/longitude at the latitude ϕ [36], [37].

$$111132.92 - 559.82\cos 2\phi + 1.175\cos 4\phi + 0.0023\cos 6\phi \tag{5}$$
$$111412.84\cos\phi - 93.5\cos 3\phi + 0.118\cos 5\phi \tag{6}$$

2) Building Routes: In this work we focus on the deconfliction against static No-Fly Zones (NFZs) and existing OVs in the airspace. Described in [38] as "*keep-out* geofences", the NFZs are regions of airspace that both the route and accompanying OVs must never overlap with. Unlike NFZs, OVs only exist for a finite duration (t_d) and so, from the perspective of the RRT*FND algorithm, appear as dynamic obstacles appearing and disappearing with the progression of time. It should be noted that in this implementation we specifically deconflict OV from the static NFZs and focus on route-OV for OVs belonging to other contracts.

As expected, we run the RRT*FND until a valid candidate solution is generated and then optimised for distance using RRT-Rope method. To improve computational efficiency, an optimisation factor is applied to the RRT-Rope algorithm defining the maximum number of nodes to consider during the optimisation, this is demonstrated in Figure 2.

This candidate solution then checks for conflicts against existing OVs by first checking for intersections between the route and OVs and then estimating the arrival time at the intersection with a user-defined departure time and either a provided cruise speed or upper and lower bound for the speed. If a conflict is found, i.e. a time aligned intersection exists with an existing OV, the system then checks the probability distribution of aircraft within the OV using Equation 4. If the probability lies above a safety threshold, the candidate solution route is severed and the

(a) Optimization Factor: 50
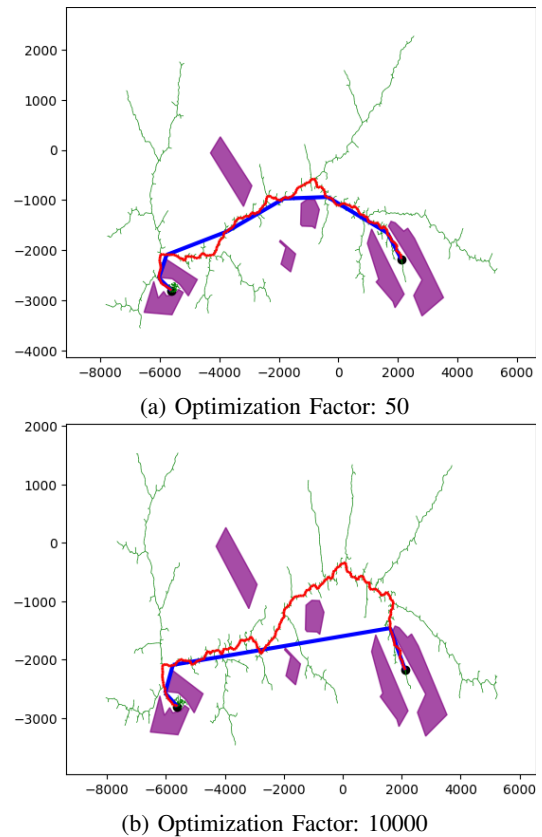

(b) Optimization Factor: 10000

Fig. 2: The RRT*FND algorithm generating a valid route (red). The route has been further optimised (blue) to decrease the overall distance while avoiding conflicts, this is done based on the RRT-Rope algorithm.

`Regrow` and `Reconnect` functions are used to repair the break with the addition of the OV which the conflict was with appearing as a static obstacle.

Once a candidate solution is generated the OV generation begins. After each OV has been generated the aircraft positions, boundaries and associated times are compared with the NFZs for violations. If a violation is found, violating nodes are removed from the graph and the `Reconnect` and `Regrow` functions of RRT*FND are again used to dynamically reconnect the graph.

The cycle is run until either a valid route is found or a timeout is reached, in which the system has not found a valid solution in a given time and so needs to be re-executed under different initialising parameters.

V. EXPERIMENTS

A. Evaluation of OV Generation

To evaluate OV generation three predefined routes were used. OVs were then generated for these routes using the method outlined in Section III. Two experiments were designed to evaluate the performance of this approach: The first to test the natural point inclusion of OVs for flights using autopilot; The second to test the sensitivity to initial conditions and the related size of the generated OVs.

1) Point inclusion: In this experiment simulations were run in BlueSky with 1000 aircraft following the routes and their state data being recorded every second. The recorded data was then compared, both spatially and temporally, with the generated 4D OVs recording the distribution of points that existed within a valid OV given their relative times.

The three test scenarios generated tested the inclusion of recorded points within the OVs. Beyond the autopilot logic implemented in BlueSky the aircraft were not controlled to specifically remain within the OVs. Three scenarios were created for the experiment testing: (1) relatively normal operating over a $\approx$ 15-minute route; (2) a significantly shorter circular route $\approx$ 8-minute flight time; (3) a complex route with frequent speed and altitude changes and aggressive turns ($> 120°$), this route was equivalent roughly to an approximate 25-minute flight.

The results can be seen in Table I, and demonstrate high performance within the first two scenarios. The poorer performance in Scenario 3, can be explained through the frequency and aggressiveness of speed and heading changes and the time it takes an aircraft to perform the manoeuvre.

TABLE I: OV point inclusion results

Scenario	Total recorded points	% of points in valid OV
1	732102	98.52%
2	553717	92.19%
3	1176679	79.55%

2) Sensitivity of OV generation to initial conditions: In this experiment the initial conditions of the aircraft were altered to provide larger uncertainty. The internal volume of the OVs, measured in kilometres^2 (km^2), was then recorded, along with the number of OVs generated. For this experiment we used the same three test routes as mentioned previously. We measured different distributions of initial speed, position and altitude.

Table II shows the results with variation in the initial speed. The initial position remained constant, and the altitude was between $[5m, 10m]$. In Case 1 speeds were generated randomly between $20m/s$ and $24m/s$ and in Case 2 between $16m/s$ and $26m/s$. The results show that, with a larger range of possible initial speeds, the solution generates more OVs and of a larger size. The increase in the number of OVs is likely a result of a larger distribution over the points used when initialising aircraft states from the normal distribution described in Section III.

TABLE II: OV sensitivity to variation in initial speeds

Scenario	Speed $[20m/s, 24m/s]$		Speed $[16m/s, 26m/s]$	
	Number of OVs	Mean Area	Number of OVs	Mean Area
1	18	0.44	18	1.3
2	12	0.1	13	0.1
3	28	0.03	29	0.13

In the next experiment, to test sensitivity to initial con-ditions, changes to the initial positional variation were implemented. Initial starting speed was fixed to $18m/s$ but, on the same three scenarios, the area aircraft could be generated in initially was altered. In the first test, from the initial starting points, a variation of $\pm10m$ was introduced, $\pm50m$ for the second and $\pm100m$ for the third in any direction. As is expected generated OVs are larger if there is a greater variation in initial starting positions, these results are demonstrated in Table III.

TABLE III: OV sensitivity to variation in starting position

Scenario	Starting Area $\pm10m$		Starting Area $\pm50m$		Starting Area $\pm100m$	
	Number of OVs	Mean Area	Number of OVs	Mean Area	Number of OVs	Mean Area
1	18	0.142	18	0.203	18	0.316
2	12	0.164	12	0.142	12	0.313
3	28	0.032	28	0.038	28	0.046

B. Evaluating Route Generation

To evaluate the performance of candidate route generation the generation time and overall route length were the concerning factors. Figure 3 shows the example environment that the routes were generated. The dark blue regions represent static NFZs. Additionally, the candidate route is denoted in cyan, optimised route is coloured magenta and the dashed orange line represents the direct distance, which is $7761m$.

Two experiments were run to evaluate performance: (1) evaluates the run-time of the route generation, this includes time to optimise the route as well; (2) evaluates the optimisation performance, examining how significantly the route can be shortened, using the maximum optimisation factor.
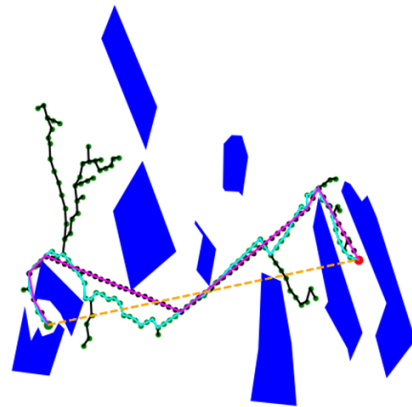

Fig. 3: Example routes generated with the cyan line being the original candidate route ($10113m$), the magenta line is the optimised route (9443), and the dashed orange line represents the direct route ($7761m$).

1) Route generation time evaluation: In this experiment the route generation was run with varying step sizes, maximum distance a node can be from any other node, to evaluate run time. Table IV shows the results of this experiment. Intuitively, the generation time increases with the resolution

of the step size. Pertaining to the *maximum nodes in graph* the $50m$ resolution step size was unable to generate a valid route with a maximum of 150 nodes in the graph, for this test the maximum number of nodes was increased to 300.

TABLE IV: Time to generate route

Resolution	Time (s)	Maximum Nodes in Graph
50 meters	108.82	300
100 meters	19.90	150
150 meters	6.84	150
200 meters	1.85	150

2) Route optimisation evaluation: To evaluate the performance of the RRT-Rope based optimisation, the lengths of the routes were compared. Again, the resolution of the step size was also considered. Table V shows there appears to be little correlation between the resolution of the step size and the difference in lengths. Each resolution was able to decrease the overall route length by more than $1000m$ where routes with step sizes $50m$ and $150m$ decreased the route length by more than $1500m$.

A significant improvement, demonstrated in Figure 2 and Figure 3, is that the optimisation significantly decreases the complexity of the route. While the RRT*FND candidate routes are valid they consist of many sharp or aggressive turns. In flight this could cause significant stress on the vehicle or, increase flight duration significantly. By reducing the path to straight line segments, the optimisation not only reduces the overall length but decreases the complexity of the route significantly.

TABLE V: Comparison of candidate and optimised route lengths

Resolution	Candidate Length (m)	Optimised Length (m)	Δ Length (m)
50 meters	11151.01	9369.89	1781.12
100 meters	10975.74	9525.31	1450.43
150 meters	11335.55	9658.40	1677.16
200 meters	11096.91	9836.05	1260.86

VI. DISCUSSIONS

A. Generation of Operational Volumes

In the approach presented in Section III, reachability analysis was performed using the DryVR [9] framework to generate reach tubes. These reach tubes provided the 3D space of the OVs. The reachability analysis performed in DryVR leverages a learned *Discrepancy Function* from the sensitivities obtained in the simulation traces. As such, with enough uncertainty or noise in the data, this approach begins to suffer from the *wrapping effect* [29], resulting in over conservative, non-descriptive OVs that grow in size exponentially. The results in Table II and Table III show the OVs growing in size with more introduced uncertainty. While uncertainty was presented manually, one source of

natural uncertainty can be the weather or, more commonly, wind. Including predicted winds into the simulation when generating the OVs will improve the overall model and provide more accurate simulation data to generate the OVs.

An improved method may look at using an analytical approach to reach tube generation. A Lagrangian approach, such as in [26]–[28] has shown to provide tighter bounded reach tubes and claim to not suffer from the *wrapping effect* [27]. While the DryVR-based approach did not necessarily suffer from the *wrapping effect* the test routes presented were relatively small, larger or more complex routes may result in additional problems.

B. Route Generation and Deconfliction

The RRT based algorithm used (RRT*FND) is not known to generate optimal shortest routes. While the approach is suited to a dynamic environment there is limited control on the optimality of the resultant route. As a result, improving this algorithm could be achieved through implementing stricter constraints in where valid nodes can be placed or using an algorithm such as D* or D* Lite [14], [39]. Both algorithms provide the dynamic requirement while yielding optimal, shortest-path trajectories. D* has also been proven to be adaptable to route planning for aircraft in [40], supporting the adoption further.

Additionally, refinements to either the aircraft positional probability or collision probability within a cost function could also improve the overall performance of the system. Presently, the approach only assigns probabilities to within a 10 meter $\times$ 10-meter region of airspace. This presents the problem of routes potentially approaching too closely to potential conflict areas. To improve this, extending regions of high values such that the values diminish in a gradient could discourage the path finding algorithm from approaching too closely to potential conflict regions.

VII. CONCLUSION

In this paper we presented a framework for generating and deconflicting routes in conjunction with the generation of 4D OVs to address airspace and operational uncertainties. We used a reachability analysis method built on the DryVR [9] framework to generate OVs given simulation data obtained through BlueSky [32]. An approach to route generation and deconfliction was given using an adaptation of the RRT algorithm: RRT*FND. This was then, optimized for shortened distance using the concept of rope-pulling described in [18]. Results demonstrated the ability to generate candidate routes, deconflict them with static NFZs and generate a contract of OVs based on the reachability analysis adapted from DryVR [9]. We also provide an oversight into a potential adaptation of this method that could remove the potential of OV generation to suffer from the wrapping effect by adapting the Lagrangian approach to reachability analysis in [28].

The approach presented in this paper, is intended to be performed pre-flight, however, due to the choice in dynamic path planning algorithm, (RRT*FND), there is prospect for an in-flight, re-planning approach that largely utilises the

same foundation. This is further reinforced by the path planning and OV generation occurring simultaneously. Additionally, proposed future work will improve both the route generation approach as well as OV generation, and introduce the strategic functionality of deconflicting against existing contracts of OVs accurately and efficiently.

Furthermore, we intend to develop a tactical, in-flight layer be added to ensure safe separation of en route aircraft. We intend to introduce such a tactical system, leveraging the constraints imposed by the aforementioned OVs. That is, a tactical separation assurance component will provide instructions to aircraft to ensure a minimum separation is maintained by aircraft, both laterally and vertically, and ensure the aircraft remain within their contracted OVs.

REFERENCES

[1] B. P. Hill, D. DeCarme, M. Mercalfe, C. Griffin, S. Wiggings, C. Metts, B. Bastedo, M. D. Patterson, and N. L. Medonca, "UAM vision concept of operations (ConOps) UAM maturity level (UML) 4," https://ntrs.nasa.gov/citations/20205011091, 12 2020, (accessed Oct. 07, 2022).

[2] EASA, "Study on the societal acceptance of urban air mobility in europe," https://www.easa.europa.eu/en/downloads/127760/en, 05 2021, (accessed Oct. 07, 2022).

[3] FAA, "Urban air mobility and advanced air mobility," https://www.faa.gov/uas/advanced_operations/urban_air_mobility, 06 2022, (accessed Oct. 07, 2022).

[4] ——, "FAA aerospace forecast fiscal years 2020-2040," https://www.easa.europa.eu/sites/default/files/dfu/uam-full-report.pdf, 2020, (accessed Oct. 24, 2022).

[5] ——, "Concept of operations V1.0: Foundation principals, roles and responsibilities, scenario and operational threads," https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf, 2020, (accessed Jan. 26, 2023).

[6] ——, "Concept of operations," https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf, 2020, (accessed Sep. 30, 2022).

[7] SESAR, "Concept of operations for European UTM systems," https://www.eurocontrol.int/project/concept-operations-european-utm-systems, 2019, (accessed Sep. 30, 2022).

[8] C. Hsieh, H. Sibai, H. Taylor, Y. Ni, and S. Mitra, "SkyTrakx: A toolkit for simulation and verification of unmanned air-traffic management systems," in 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 372–379.

[9] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, "DryVR: Data-driven verification and compositional reasoning for automotive systems," in Computer Aided Verification, R. Majumdar and V. Kunčak, Eds. Cham: Springer International Publishing, 2017, pp. 441–461.

[10] Catapult, "Implementing an open-access UTM framework for the UK," https://cp.catapult.org.uk/wp-content/uploads/2021/06/Implementing-an-Open-Access-UTM-Framework.pdf, 2021, accessed Jan. 11, 2023.

[11] J. L. Rios, J. Homola, N. Craven, P. Verma, and V. Baskaran, "Strategic deconfliction performance: Results and analysis from the NASA UTM technical capability level 4 demonstration," NASA, Tech. Rep., 2020.

[12] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

[13] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.

[14] A. Stentz, "Optimal and efficient path planning for partially-known environments," in Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1994, pp. 3310–3317 vol.4.

[15] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University (TR 98–11), Tech. Rep., 1998.

[16] B. Tong, Q. Liu, and C. Dai, "A RRT FN based path replanning algorithm," in 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol. 1, 2019, pp. 1435–1445.

[17] O. Adiyatov and H. A. Varol, "A novel RRT-based algorithm for motion planning in dynamic environments," in 2017 IEEE International Conference on Mechatronics and Automation (ICMA), 2017, pp. 1416–1421.

[18] L. Petit and A. L. Desbiens, "RRT-Rope: A deterministic shortening approach for fast near-optimal path planning in large-scale uncluttered 3D environments," in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 1111–1118.

[19] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[20] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions," 2015.

[21] D. Sacharny and T. Henderson, Strategic Deconfliction. Springer International Publishing, 2022, pp. 35–56.

[22] S. Chaimatanan, D. Delahaye, and M. Mongeau, "Strategic deconfliction of aircraft trajectories," in ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management, Toulouse, France, July 2013, p. p xxxx.

[23] X. Tang, Y. Zhang, P. Chen, B. Li, and S. Han, "Strategic deconfliction of 4D trajectory and perturbation analysis for air traffic control and automation system," Discrete Dynamics in Nature and Society, vol. 2016, p. 7028305, Sep 2016.

[24] J. Berling, A. Lau, and V. Gollnick, "European air traffic flow management with strategic deconfliction," in Operations Research Proceedings 2015, K. F. Dörner, I. Ljubic, G. Pflug, and G. Tragler, Eds. Cham: Springer International Publishing, 2017, pp. 279–286.

[25] J. R. Bertram, P. Wei, and J. Zambreno, "Scalable FastMDP for pre-departure airspace reservation and strategic de-conflict," 2020.

[26] J. Cyranka, M. A. Islam, S. A. Smolka, S. Gao, and R. Grosu, "Tight continuous-time reachtubes for lagrangian reachability," in 2018 IEEE Conference on Decision and Control (CDC), 2018, pp. 6854–6861.

[27] S. Gruenbacher, J. Cyranka, M. Lechner, M. A. Islam, S. A. Smolka, and R. Grosu, "Lagrangian reachtubes: The next generation," in 2020 59th IEEE Conference on Decision and Control (CDC). IEEE, dec 2020.

[28] S. Gruenbacher, M. Lechner, R. Hasani, D. Rus, T. A. Henzinger, S. Smolka, and R. Grosu, "GoTube: Scalable stochastic verification of continuous-depth models," https://arxiv.org/abs/2107.08467, 2021.

[29] A. Neumaier, The Wrapping Effect, Ellipsoid Arithmetic, Stability and Confidence Regions. Vienna: Springer Vienna, 1993, pp. 175–190.

[30] G. Zhu and P. Wei, "Low-altitude uas traffic coordination with dynamic geofencing," in 16th AIAA Aviation Technology, Integration, and Operations Conference, 2016.

[31] J. Kim and E. Atkins, "Airspace geofencing and flight planning for low-altitude, urban, small unmanned aircraft systems," Applied Sciences, vol. 12, no. 2, 2022.

[32] J. Hoekstra and J. Ellerbroek, "BlueSky ATC simulator project: an open data and open source approach," 2016.

[33] J. Sun, J. M. Hoekstra, and J. Ellerbroek, "OpenAP: An open-source aircraft performance model for air transportation studies and simulations," Aerospace, vol. 7, no. 8, p. 104, 2020.

[34] Airbus, "Vahana: Our single-seat eVTOL demonstrator," https://www.airbus.com/en/urbanairmobility/cityairbus-nextgen/vahana, (accessed Nov. 12, 2022).

[35] P. Pradeep, "Arrival management for eVTOL aircraft in on-demand urban air mobility," Ph.D. dissertation, Iowa State University, 05 2019.

[36] J. P. Snyder, "Map projections: A working manual," U.S. Department of the Interior, Washington, D.C., Tech. Rep., 1987, report.

[37] L. M. Bugayevskiy and J. Snyder, Map Projections - A Reference Manual. CRC Press, 1995.

[38] M. Stevens and E. Atkins, "Geofence definition and deconfliction for UAS traffic management," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 5880–5889, 2021.

[39] S. Koenig and M. Likhachev, "D* Lite," AAAI Conference of Artificial Intelligence (AAAI), pp. 476–483, 2002.

[40] S. Majumder and M. S. Prasad, "Three dimensional D* algorithm for incremental path planning in uncooperative environment," in 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), 2016, pp. 431–435.