Gen4jectory algorithm – 4-D trajectory planning with minimised flight time for multiple rotary-wing UAVs

Ivan Panov*, Mouad Boumediene**, Henrik Skov Midtiby** and Kjeld Jensen**
* Department of Business Development and Technology, Aarhus University, Herning, Denmark 7400
** SDU Drone Center, University of Southern Denmark, Odense, Denmark 5230

ABSTRACT

The paper proposes an innovative Gen4jectory algorithm that allows planning multiple 4-D trajectories considering unique aircraft performance data of the uncrewed rotary-wing air traffic. The algorithm guarantees zero cases of Loss of Separation (LoS). We describe the physical model of drone (Unmanned Aerial Vehicle (UAV)) motion, map representation, pathfinding technique, 4-D trajectory planning, check of LoS, experimental setup and simulation results. Finally, we discuss the promising ways of further research.

1 INTRODUCTION

The highest level of flight safety in Europe sets a top expectation on unmanned aviation integration in airspace [1]. Simultaneously, U-space and Urban Air Mobility stakeholders expect a high business efficiency of the uncrewed operations where autonomous systems will be able to substitute an expensive labour force. People have limited capabilities in manual planning and managing uncrewed air traffic in congested airspace. These limitations highlight the important need for an autonomous guidance system [2] which will be able to manage U-space and Urban Air Mobility traffic in a safe and efficient way.

However, what is the efficiency of the uncrewed operations? For business applications, the efficiency can be defined as on-time delivery rate, cost per mile, fuel efficiency, vehicle utilisation rate, etc. The metric choice depends on the operation requirements. For example, surveillance operations may require maximisation of endurance; consumer goods delivery and human transportation may require flight time minimization; operation in the night time may require noise limitation; heavy cargo delivery may require energy consumption minimization, etc. Practically speaking, it implies very different approaches to UAV 4-D trajectory planning. For instance, noise limitation could require higher altitudes of operation and avoidance of the noise-sensitive areas, flight time minimisation could require high tilt angles with a high level of thrust during UAV operation.

*Email address: ivan.panov@btech.au.dk

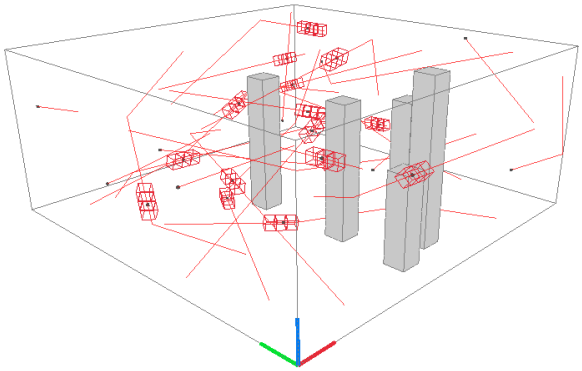

Figure 1: Visualization of the Gen4jectory algorithm in an urban environment

According to the forecast given by SESAR [3], delivery and transportation operations will cover about 45% of all UAV operations over Europe by 2050. We expect that delivery and transportation missions will require flight time minimisation in most cases. Bearing this in mind we decided to conduct a research project on uncrewed traffic 4-D trajectory planning focusing on the operations that require flight time minimisation.

4-D trajectory planning is a safety-critical aspect of autonomous UAV operations. It means that the proposed algorithm must exclude LoS for all airspace participants as a result of the operations planning.

The paper begins with an abstract and introduction. After that, we discuss our literature review and present the problem formulation. In the section “Algorithm design and methodology”, we discuss our physical model of a rotary-wing UAV, the choice of a map representation and pathfinding technique, and an approach for avoiding LoS. Then we discuss the simulation results, which is followed by a conclusion. A list of references finalizes the paper.

The main paper contribution is a novel Gen4jectory algorithm that allows planning 4-D trajectories for rotary-wing autonomous UAVs for autonomous uncrewed traffic management systems. (Figure 1. provides an example of 4-D trajectory planning in an urban environment (grey 3-D rectangles represent buildings, black points represent UAVs, and red 4-D rectangles indicate separation requirements).

Additionally, we make a comparison with a super sim-

plified model of flight, where flight time between waypoints is calculated by considering only maximum speed in level flight.

2 LITERATURE REVIEW

4-D trajectory planning for air traffic is a classical approach in the field of commercial aviation [4], [5]. However, this approach cannot be completely applied to the Unmanned aircraft system Traffic Management (UTM) as UAVs do flight with no flight corridors and flight levels. Additionally, 4-D trajectory planning for UTM requires a more complex approach as UAVs operate on a very low level facing ground obstacles and other on-surface objects and subjects.

4-D trajectory planning for commercial aviation considers aircraft performance which allows to pre-calculate with sufficient accuracy where and when the aircraft will be. In Europe, this issue is addressed via the Base of Aircraft Data (BADA) database, which incorporates essential aircraft performance data on certified aircraft [6]. The certification process of a commercial airliner is an expensive, well-documented process with an obligatory measuring of the aircraft performance data. There is no official demand for U-space air traffic participants to share aircraft performance data as a mandatory condition for operation approval. Therefore, 4-D trajectory planning for U-space is a complex task where alternative approaches must be identified or invented.

We expect that computational methods could help in obtaining quick and low-cost data extraction on aircraft performance based on the 3-D model of a UAV. However, we do not exclude machine learning algorithms that will be able to predict rough aircraft performance data based on a set of images of the UAV. We encourage the scientific community to contribute in this area to fill the gap.

While the issue of collecting aircraft performance data on UAVs remains unresolved, the researchers use assumptions or simply ignore aircraft dynamics constraints within their studies. For example, the authors of [7] proposed a solution for the multi-robot trajectory generation; however, the aerodynamic drag issue was not addressed. Similarly, the authors of [8] proposed the dynamic trajectory planning method missing an essential discussion on aircraft dynamics constraints. The authors of [9] invented a bio-inspired collision-free 4D trajectory generation method that incorporates a set of techniques to plan UAV trajectories like Tau-guidance, geometric, conflict detection, Particle Swarm Optimization, Conflict Detection, and Resolution. Again, the UAV dynamics model was not described in the paper. Similarly, the authors of the papers [10, 11, 12, 13] did not present a satisfactory physical model of the UAV to address 4-D trajectory planning.

In addition to the challenges in 4-D trajectory planning mentioned earlier, path planning in 3-D space for multiple UAVs also requires algorithms that can account for static obstacles such as buildings, and dynamic obstacles such as other UAVs sharing the same airspace. One such algorithm is theta-star [14] and its variants [15, 16], an extension of the A-star algorithm [17], which has been effectively used in 2-D path planning [18] and adapted for 3-D environments [19]. theta-star differs from A-star by allowing direct path connections (line of sight) between nodes, which reduces the path length and results in smoother routes [14].

Recent studies have explored the use of theta-star and its variants for UAV path planning in 3-D [20, 21], highlighting its ability to generate near-optimal paths while maintaining computational efficiency. For instance, theta-star has been applied in scenarios where UAVs must navigate urban environments with a high density of obstacles [22, 23]. This was done by considering the UAV's ability to move freely in three dimensions.

3 PROBLEM FORMULATION

A precise UAV flight dynamics model is a resolved issue [24]. However, precision requires precise aircraft performance data and costly computation time. For example, mass distribution on the UAV body is essential to estimate the corresponding moments. Obtaining such data for a large number of drone models is costly and inefficient. In this light, we decided to conduct a research project on a simplified flight dynamics model to allow 4-D trajectory planning for U-space needs. Additionally, we combined our flight dynamics model with the theta-star pathfinding algorithm [25], selected for its superior path quality. We implemented a LoS check that uses 4D collision detection to identify potential conflicts between the spaces reserved by two drones, accounting for both separation volumes in 3-D Euclidean space and their temporal overlap. To manage potential conflicts effectively, we also applied a priority-based resolution strategy.

The problem of modelling in physics normally implies finding a set of suitable instruments to deliver an acceptable level of precision. Having this in mind, we compared the results of the flight dynamics model of rotary-wing UAVs with a super simplified one where the trajectory is calculated based on the maximum velocity of the aircraft. If the deviation is not significant, then the problem of UAV 4-D trajectory planning in U-space could be solved without complex physical modelling during flight. Instead, pre-flight calculations could be done to find the maximum velocity of the UAV for the current case like aircraft weight, air temperature, air pressure, specific cargo transportation, and wind impact.

In this paper, we do not address wind impact, as it is one of the multiple essential constraints in the UAV autonomous guidance [2]. Instead, we focus our study on the fusion of our physical model with the theta-star-based algorithm to test LoS-free flight in our simulation.

The problem of 4-D trajectory planning is a complex multi-factor task with various interconnected constraints [2]. We acknowledge the complexity, and the essentiality of splitting the 4-D trajectory planning problem into the sub-issues. Specifically in this paper, we raise the following research

questions:

- How to plan 4-D trajectories for the uncrewed air traffic in order to minimise flight time for each consecutive UAV?
- What is the difference between 4-D trajectory planning based on the Gen4jectory physical model and a model that considers only the maximum velocity of the UAVs?

4 ALGORITHM DESIGN AND METHODOLOGY

To address the research questions, we invented an experimental setup based on the Gen4jectory algorithm [26] (Figure 2). The algorithm predefined:

- The 3-D map used in our experiment has dimensions of 350x350x150 metres, representing the airspace in which the UAVs operate. This airspace is defined as a 3-D rectangle, where the length and width span 350 metres each, and the height extends up to 150 metres above ground level, creating a bounded environment for path planning and navigation tasks.
- We created a lattice graph starting with a 3-D grid of points, spaced evenly at intervals of 25 metres. Each point on the grid became a node in the graph, and we connected these nodes to their neighbouring points with edges, forming a lattice structure.
- The start and goal positions of the UAVs were randomly generated and sampled from a normal distribution within the map's bounds. This allowed us to capture a broad spectrum of potential conflict scenarios, enhancing the robustness of our testing and analysis.
- Separation volumes are defined as 4D-directed rectangles (separation volumes) between two points. Each 4-D rectangle (separation volume) has a length of 30 metres, with the other two dimensions being 10x10 metres.

In the next stage, the simulation randomly generates departure and arrival waypoints for all UAVs involved. Also, it generates random values of aircraft performance for rotary-wing UAVs in the realistic range. Specifically, aircraft mass ranged from 2 kg to 500 kg, which covers a large variety of the U-space airspace participants including flying taxis. The drag coefficient is ranged from 0.6 to 1.2. We chose the range of cross-sectional areas in the range from 0.5 to 4 metres. Additionally we made cross-sectional areas linearly dependent on aircraft mass: The heavier the UAV, the greater its cross-sectional area. Similarly, maximum thrust depends on aircraft mass, ranging from 20% to 100% reserve over aircraft weight. The standard atmosphere is chosen as a typical baseline in aviation research. However, this variable can be adjusted to any value to reflect the real-life temperature and atmospheric pressure impact on flight dynamics.

Next, the algorithm sequentially calculates 4-D trajectories for each UAV. In the case of the LoS threat in a concrete place with concrete time, an alternative trajectory is being calculated. This process continues until a safe near-optimal trajectory with minimised flight time is found. We state "near-optimal" because map representation with predefined waypoints corresponded with deviations from optimality. The greater resolution gives the more optimal solution. However, it also implies a greater time complexity as the theta-star algorithm requires calculating more potential options to navigate. The optimal balance should be found as a compromise based on operational needs. Similarly, the greater the number of UAVs flying nearby, the more complicated check of LoS is required.

In the next stage, a supplementary algorithm makes a simplified calculation of flight time required to fly from one waypoint to another based on UAV maximum level flight speed. However, to make the maximum speed known, we do precalculations based on aircraft performance data and current air density. We cannot rely on the maximum speed given in the flight manual since UAVs can carry extra cargo which affects aerodynamic drag, the temperature can be far from the standard atmosphere, etc.

In the fifth stage, the simulation runs multiple flight experiments while checking: LoS event, flight time for reaching the destination point based on our main physical model in comparison with the super simplified one. Finally, the data is recorded and stored in a table which we use for analysis.

The methodology of this study includes a basic literature review, coding of the experimental setup, performing an analysis of the experiment (simulation) repeated 1,000 times, with random realistic flight characteristics for every UAV, a map 350x350x150 m, random initial positions and destination points of the air space participants, and 5 skyscrapers with random locations. The number of UAVs in flight is 25.

4.1 Physical model

4.1.1 Notation

a - acceleration, m/s^2.
C_D - drag coefficient.
$Chld$ – child waypoint in NEU coordinate system.
dp – dot product of $\boldsymbol{GP\_Par}$.
d_{plnd} - distance planned.
g – acceleration due to gravity, m/s2.
GP – grandparent waypoint in North East Up (NEU) coordinate system.
gp – coordinate of the grandparent waypoint in NEU.
m - UAV mass, kg.
$magGP_Par$ – magnitude of $\boldsymbol{GP\_Par}$.
$magPar\_Chld$ – magnitude of $\boldsymbol{Par\_Chld}$.
M_{end} - a point of the manoeuvre end.
M_{start} - a point of the manoeuvre start.

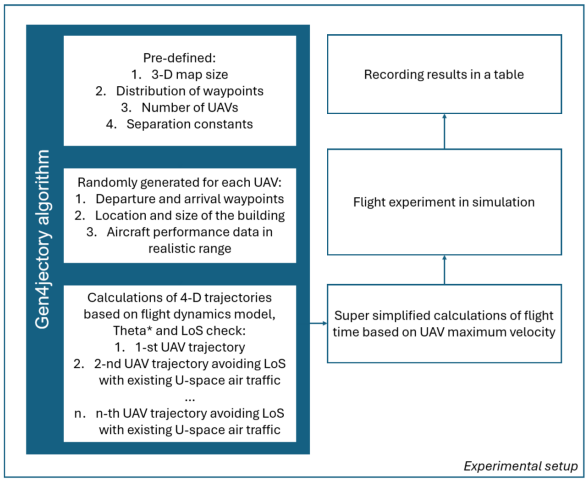

Figure 2: Process Diagram of the experimental setup

Par – parent waypoint in NEU coordinate system.
$prnt$ - coordinate of the parent waypoint in NEU.
S - cross sectional area.
t - flight time, s.
t_{new} - updated time, s.
v - final calculated velocity for a direct section of the flight path, m/s.
v_0 - initial velocity, m/s.
Δ - difference between a vertical component of thrust and UAV weight.
γ - flight path angle, degrees.
$\gamma_{\theta_{lim}}$ - flight path angle for flight with angle θ_{lim} and max thrust, degrees.
ε - angle of the flight path change.
θ - UAV tilt angle, degrees.
θ_{lim} - UAV tilt limit, degrees.
θ_{mh} - UAV tilt for flight in horizon with max thrust, degrees.
ρ - air density, kg/m^3.
$\boldsymbol{F}_{mh}$ - vector of maximum force in horizontal flight without drag, N.
$\boldsymbol{F}_{net}$ - vector of maximum net force, N.
$\boldsymbol{F}_{net_{mh}}$ - vector of maximum net force in horizontal flight, N.
$\boldsymbol{F}_t$ - vector of maximum thrust, N.
$\boldsymbol{F}_{t_{mh}}$ - vector of thrust in horizontal flight, N.
$\boldsymbol{F}_{t_{min}}$ - vector of minimum thrust for vertical descent, N.
$\boldsymbol{F}_{t_{\theta lim}}$ - vector of thrust with θ_{lim} angle, N.
$\boldsymbol{F}_{tr}$ - net force without drag impact, N.
$\boldsymbol{F}_{tr_{hc1}}$ - horizontal component of the net force without drag impact before manoeuvre, N.
$\boldsymbol{F}_{tr_{hc2}}$ - horizontal component of the net force without drag impact during manoeuvre, N.
$\boldsymbol{F}_x$- horizontal component of thrust vector, N.
$\boldsymbol{F}_z$ - vertical component of a corresponded vector, N.
$\boldsymbol{GP\_Par}$ - vector from waypoint GP to waypoint Par.
$\boldsymbol{Par\_Chld}$ - vector from waypoint Par to waypoint $Chld$.

Subscript N/E/U stands for North/East/Up coordinate accordingly. We name some values with "$_{red}$" in case the reduced thrust is discussed.

All symbols representing force vectors in this paper we denote by bold letters. The corresponding magnitudes, when referenced, we denote by italic letters without bold formatting.

4.1.2 Time-dependent UAV motion modelling

In the paper we consider the next conditions:
- NEU Body-fixed coordinate system.
- NEU origin at sea level at some point.
- Mass-point model of UAV.
- Earth curvature is ignored.
- Wind is zero.

To plan a 4-D trajectory safely, it is essential to reserve a certain volume of airspace at a certain time to avoid the risk of LoS. Vertical, horizontal, and longitudinal separation create a 4-D rectangle, whose sides are normally defined by national regulation. However, to reduce computation, the 4-D rectangle can be simplified as a set of safety bubbles placed on the aircraft's current, most recent past, and nearest future positions. If there is no presence of two aircraft in the same safety area, then the risk of potential collision is significantly mitigated. To make sure that there is no potential presence of two aircraft in the same volume of airspace, it is required to calculate where and when the aircraft will be. For that, it is essential to calculate flight time, initial and final velocity, and to address the inertia issue during manoeuvres. We created the code [26] that resolves this issue.

Following a common practice in aviation research, we assume that the flight occurs under standard atmospheric conditions with no wind. To calculate the flight time required to cover a distance between two waypoints, the physical model has three layers: inputs, calculations, and outputs. We assume that inputs are all given properties of the drone. Among them: mass (m), cross-sectional area (S) and drag coefficient (C_D), the maximum thrust (F_t), the maximal tilt angle of the UAV (θ_{lim}), initial velocity (v_0), air density (ρ), minimum percent of thrust for vertical descending, and delta time for Euler method calculations. The coordinates of the waypoints in the NEU reference system are also known.

Based on the inputs provided, the algorithm transforms Parent (departure) and Child (arrival) waypoints to a vertical plane where the Parent waypoint is placed in the origin. We switch to a body-fixed reference system where the z-axis is pointed up and the x-axis is directed to the Child (next) waypoint. In this case, it is essential to find the coordinates of the Parent and Child waypoint for each straight segment of the trajectory. Once the UAV has reached the area of the

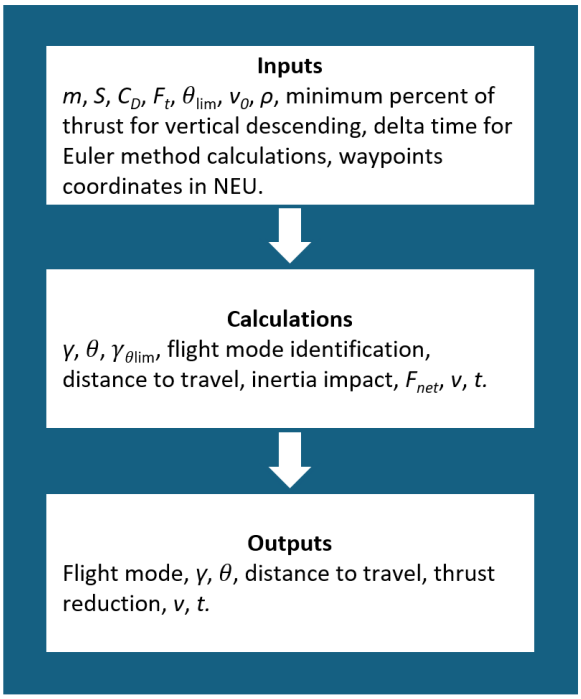

Figure 3: The high-level scheme of physical model calculations

arrival point, the algorithm switches through the loop to the next segment, and the Child point is now classified as the Parent point. It happens until the final destination waypoint has been reached.

There are a few steps to perform a transformation to the new coordinate system. Firstly, it is essential to select the departure and arrival waypoints that correspond to the segment required. Then we define two vectors corresponding to the departure and arrival waypoints.

$$\mathbf{W}_{\text{chld}} = (N_{\text{chld}},\, E_{\text{chld}},\, U_{\text{chld}}) \tag{1}$$

$$\mathbf{W}_{\text{par}} = (N_{\text{par}},\, E_{\text{par}},\, U_{\text{par}}) \tag{2}$$

A relative position vector can be found:

$$\mathbf{W}'_{\text{chld}} = \mathbf{W}_{\text{par}} - \mathbf{W}_{\text{chld}} \tag{3}$$

We ignore the vertical component and normalise the horizontal component of $\mathbf{W}_{\text{chld}}$', introducing new variables h_s, x_{axis}, Z_{axis}.

$$h_s = \begin{pmatrix} W'_{\text{chldN}} \\ W'_{\text{chldE}} \\ 0 \end{pmatrix} \tag{4}$$

$$x_{\text{axis}} = \frac{h_s}{\|h_s\|} \tag{5}$$

$$Z_{\text{axis}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{6}$$

Arrival waypoint x-coordinate and z-coordinate in body-fixed coordinate system:

$$w_x = (w_{\text{par}} - w_{\text{chld}}) \cdot x_{\text{axis}} \tag{7}$$

$$w_z = (w_{\text{par}} - w_{\text{chld}}) \cdot z_{\text{axis}} \tag{8}$$

Now the arrival point is transformed to the origin, and accordingly, the destination point has x and z coordinates.

The basic aerodynamic forces and maximum tilt angle θ have a significant impact on finding a flight mode that allows minimising flight time. With a high thrust-to-weight ratio, the aircraft will be ascending with angle $\gamma_{\theta_{lim}}$ if it has the maximum tilt angle with the maximum thrust.

However, if we significantly increase the weight of UAV, it could lead to a low thrust-to-weight ratio. In this case, the aircraft will be descending with angle $\gamma_{\theta_{lim}}$ at the maximum tilt angle with the maximum thrust. We mark flight modes for the high thrust-to-weight ratio with the capital letter ´A´ and for the low thrust-to-weight ratio with the capital letter ´B´.

Until the UAV has reached $\gamma_{\theta_{lim}}$ it shall fly at maximum thrust to minimise flight time - modes IA, IB, and IIB (Figures 4 and 5). Exceptions are possible for the aggressive style flight with multiple changes of flight directions. For U-space traffic, we expect mainly direct trajectories with limited manoeuvring.

Once $\gamma_{\theta_{lim}}$ is reached, the descent with γ in range (- π/2, $\gamma_{\theta_{lim}}$) is only possible if thrust is decreased - flight modes IIA, IIIA, and IIIB (Figures 4 and 5).

Table 1 identifies the significant differences between the flight modes. For example, vertical ascend and descent imply zero tilt angle. However, the maximum thrust is required for the fastest ascent. In order to maintain aircraft orientation, the minimum thrust can be applied for the vertical descending. Flight modes IA, IB, and IIB imply maximum thrust; however, UAV tilt angle requires calculations. Flight modes IIA, IIIA, and IIIB operate at the maximum tilt angle; however, the level of thrust in Newtons must be calculated.

Flight mode	**Angle γ**	**Thrust**	**Angle θ**
Vertical ascent	$\pi/2$	Max	0
IA	$[\gamma_{\theta_{\text{lim}}}, \pi/2)$	Max	To find
IB	$[0, \pi/2)$	Max	To find
IIB	$[\gamma_{\theta_{\text{lim}}}, 0)$	Max	To find
IIA	$[0, \gamma_{\theta_{\text{lim}}})$	To find	θ_{lim}
IIIA	$(-\pi/2, 0)$	To find	θ_{lim}
IIIB	$(-\pi/2, \gamma_{\theta_{\text{lim}}})$	To find	θ_{lim}
Vertical descent	$-\pi/2$	As input	0

Table 1: Flight modes

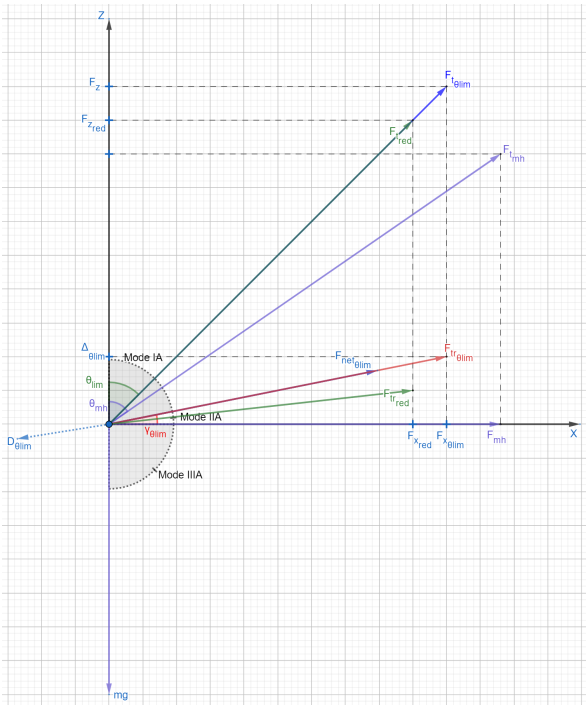

Figure 4: Flight modes A. For rotary-wing UAV with a high thrust-to-weight ratio ($\gamma_{\theta_{lim}} \geq 0$)

Travel distance is essential to find the flight time required and the velocity at the end of the segment.

$$d_{\text{plnd}} = \sqrt{x^2 + z^2} \tag{9}$$

We propose a new variable b to make calculations look simple.

$$b = \frac{1}{2} \cdot \rho \cdot S \cdot C_D \cdot V^2 \tag{10}$$

As waypoint coordinates are given, it is possible to find flight path angle γ:
If x $>$ 0, then γ = $\arctan(z/x)$.
If x = 0 and y $>$ 0, then γ=π/2.
If x = 0 and y $<$ 0, then γ=-π/2.
If x = 0 and y = 0, then the waypoint is at the origin.

No need to change UAV position. Now γ is known, thus θ can be found which is required for the flight modes IA, IB, and IIB according to table 1. As Δ is a vertical component of $\boldsymbol{F}_{tr}$ and $\boldsymbol{F}_x$ is a horizontal component of $\boldsymbol{F}_{tr}$, then:

$$F_{tr} = \sqrt{\Delta^2 + F_x^2} \tag{11}$$

$$\sin\gamma = \frac{\Delta}{F_{tr}} \tag{12}$$

$$\Delta = \sin\gamma \cdot \sqrt{\Delta^2 + F_x^2} \tag{13}$$

The magnitude of the horizontal force is found.

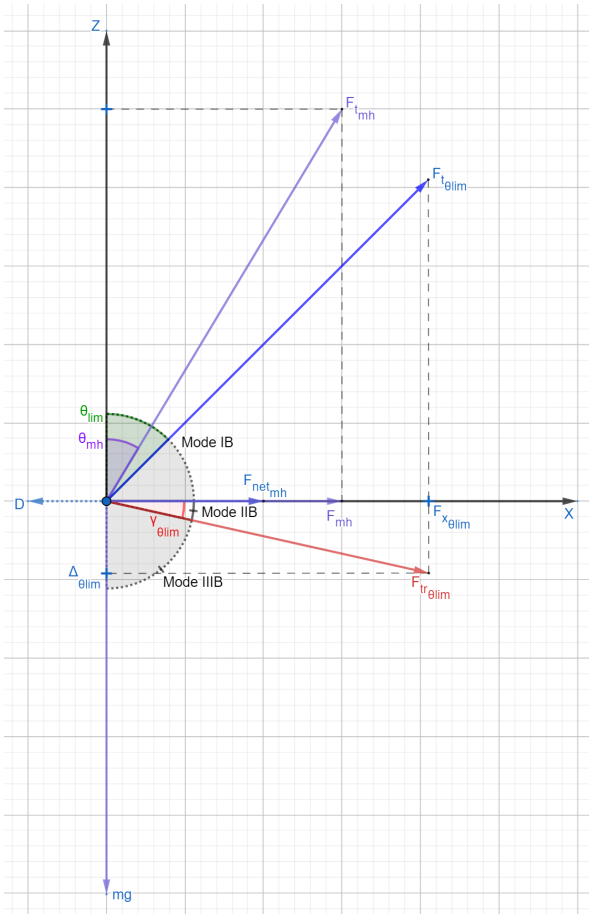

Figure 5: Flight modes B. For rotary-wing UAV with a low thrust-to-weight ratio ($\gamma_{\theta_{lim}} < 0$)

$$F_x = \sqrt{F_t^2 - (\Delta + m \cdot g)^2} \tag{14}$$

$$\Delta = \sin\gamma \cdot \sqrt{\Delta^2 + F_t^2 - (\Delta + m \cdot g)^2} \tag{15}$$

$$\Delta^2 - \sin^2\gamma \cdot \left(\Delta^2 + F_t^2 - \Delta^2 - 2 \cdot \Delta \cdot m \cdot g - (m \cdot g)^2\right) = 0 \tag{16}$$

$$\Delta^2 + 2 \cdot \sin^2\gamma \cdot m \cdot g \cdot \Delta - \sin^2\gamma \cdot F_t^2 + \sin^2\gamma \cdot (m \cdot g)^2 = 0 \tag{17}$$

To make the equation look simple, we enter p and c.

$$p = 2 \cdot \sin^2\gamma \cdot m \cdot g \tag{18}$$

$$c = -\sin^2\gamma \cdot F_t^2 + \sin^2\gamma \cdot (m \cdot g)^2 \tag{19}$$

$$\Delta^2 + p \cdot \Delta + c = 0 \tag{20}$$

$$D = p^2 - 4c \tag{21}$$

$$\Delta_{1,2} = \frac{-p \pm \sqrt{p^2 - 4c}}{2} \tag{22}$$

In the case of flight mode IA and IB $\Delta = \Delta_1$. For IIB $\Delta = \Delta_2$.

$$\cos\theta = \frac{\Delta + m \cdot g}{F_t} \tag{23}$$

$$\theta = \arccos\left(\frac{\Delta + m \cdot g}{F_t}\right) \tag{24}$$

As θ is calculated, it is possible to find the horizontal component of the thrust vector for the modes IA, IB, and IIB:

$$F_x = \sin\theta \cdot F_t \tag{25}$$

And F_{tr} can be found with an equation (11).

For the modes IIA, IIIA, and IIIB, the UAV flies at θ_{lim} and the reduced thrust shall be calculated. In this case, Δ is different.

$$\sin(\theta_{\text{lim}}) = \frac{F_{x\_red}}{F_{t\_red}} \tag{26}$$

$$\tan(\theta_{\text{lim}}) = \frac{F_{x\_red}}{m \cdot g + \Delta_{\text{red}}} \tag{27}$$

$$F_{x\_red} = \tan(\theta_{\text{lim}}) \cdot (m \cdot g + \Delta_{\text{red}}) \tag{28}$$

$$\tan\gamma = \frac{\Delta_{\text{red}}}{F_{x\_red}} \tag{29}$$

$$F_{x\_red} = \frac{\Delta_{\text{red}}}{\tan\gamma} \tag{30}$$

$$\tan(\theta_{\text{lim}}) \cdot (m \cdot g + \Delta_{\text{red}}) = \frac{\Delta_{\text{red}}}{\tan\gamma} \tag{31}$$

$$\Delta_{\text{red}} - \tan\gamma \cdot \tan\theta_{\text{lim}} \cdot m \cdot g - \tan\gamma \cdot \tan\theta_{\text{lim}} \cdot \Delta_{\text{red}} = 0 \tag{32}$$

$$\Delta_{\text{red}} = \frac{\tan\gamma \cdot \tan\theta_{\text{lim}} \cdot m \cdot g}{1 - \tan\gamma \cdot \tan\theta_{\text{lim}}} \tag{33}$$

F_{x_red} is known, see equation (28).

$$F_{tr_{\text{red}}} = \sqrt{\Delta_{\text{red}}^2 + F_{x\_red}^2} \tag{34}$$

Table 2 summarises the similarities and differences in the calculations in finding the vector of net force without drag impact. For example, vertical ascend and descent imply known thrust. Minimum thrust is essential for vertical descent. Flight modes IA, IB, and IIB have commonality in finding F_x and F_{tr}, but IIB differs in finding Δ and θ. Flight modes IIA, IIIA, and IIIB have the same line of calculations.

Flight mode	Similarities	Differences
Vertical ascend	F_t is given	
IA	*(11), (25)*	$\Delta = \Delta_1, \theta = \theta_1$
IB		
IIB		$\Delta = \Delta_2, \theta = \theta_2$
IIA	*(28), (33), (34)*	
IIIA		
IIIB		
Vertical descend	Given F_t and F_{t_red}	

Table 2: Comparison of flight modes

F_{tr} has been calculated for each mode, and the net force can be found via this equation:

$$F_{\text{net}} = F_{\text{tr}} - \frac{1}{2} \cdot \rho \cdot S \cdot C_D \cdot V^2 \tag{35}$$

The acceleration can be found through Newton's law.

$$a = \frac{F_{\text{net}}}{m} \tag{36}$$

Finding flight time between waypoints and final velocity, position update, and time update can be calculated via Euler's method.

$$v_{t+\Delta t} = v_t + a_t \cdot \Delta t \tag{37}$$

$$x_{t+\Delta t} = x_t + v_t \cdot \Delta t \tag{38}$$

$$t_{\text{new}} = t + \Delta t \tag{39}$$

Now, the travelling time between two waypoints can be calculated.

The calculations we did so far considered only two waypoints - Parent and Child. That is not an issue for the first direct segment of the 4-D trajectory because we assume that the initial velocity is known, and it is zero (the UAV takes off from a vertiport). However, in case of changing flight path direction, the inertia gets the higher impact the greater the angle of manoeuvre. Since we intend to plan a 4-D trajectory based on limited information about the UAV, a simplified model of inertia is required. In this light, we have to consider three waypoints - Grandparent (GP), Parent (Par), and Child ($Chld$). If we put a plane on the three waypoints, then it is possible to draw the manoeuvre in 2-D space which is simple for understanding. One of the cases is a horizontal manoeuvre in the vicinity of the Parent waypoint (Figure 6). Since rotary-wing UAVs have a small turning radius in comparison to separation requirements, it is possible to neglect the more precise calculations of the 4-D trajectory and accept the corresponding deviations.

Figure 6 gives an example of a 90-degree manoeuvre in a plane based on the initial three waypoints. The model implies minor deviations due to simplification: The UAV travels with a cutting angle and does not reach the Parent point precisely. The strong side of this approach is the simplicity of addressing inertia with having limited data on the rotary-wing UAV.

To simplify the calculation, we assume that a UAV approaching the Parent's waypoint vicinity can turn on the angle ε in advance (Figure 6) - in the point M_{start}. Due to inertia, the UAV will continue moving in its current state of motion, but aerodynamic drag will gradually decelerate it, potentially causing the UAV to move towards the Parent-Child segment. It is not necessary to find the precise position of M_{start}; this point is essential just for understanding how the issue of inertia can be addressed in a simplified way. However, angle ε is essential to calculate as it has a high impact on the initial velocity for the Parent-Child segment. For example, for very small ε, the final velocity for the Grandparent-Parent segment will be almost equal to the initial velocity of the Parent-Child. For $\varepsilon = \pi$, the initial velocity of the Parent-Child segment will be equal to the negative meaning of the final velocity for the Grandparent-Parent segment.

Since v is the final calculated velocity for a direct section of the flight path, and v_0 is the initial velocity, the next equation can be used as a simplification of the manoeuvre for any angle ε.

$$v_0 = \cos\varepsilon \cdot v \tag{40}$$

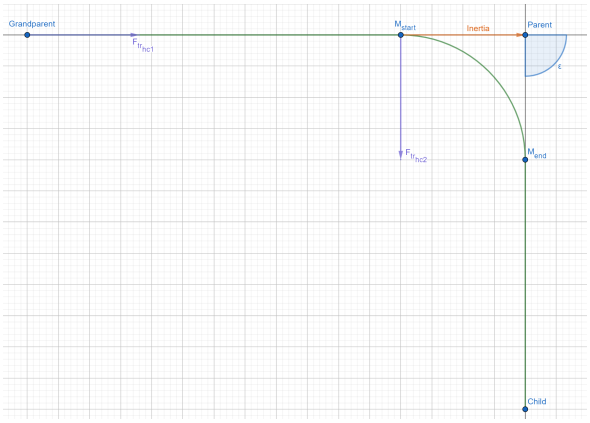

Figure 6: Manoeuvre at 90 degrees on the constant altitude. View from above

To find the angle ε, some calculations are required. The coordinates of three waypoints in NEU are known.

$$GP = \left[gp_N, gp_E, gp_U\right] \tag{41}$$

$$Par = \left[prnt_N, prnt_E, prnt_U\right] \tag{42}$$

$$Chld = \left[chld_N, chld_E, chld_U\right] \tag{43}$$

To compute vectors from point:

$$\boldsymbol{GP\_Par} = (prnt_N - gp_N, prnt_E - gp_E, prnt_U - gp_U) \tag{44}$$

$$\boldsymbol{Par\_Chld} = (chld_N - prnt_N, chld_E - prnt_E, chld_U - prnt_U) \tag{45}$$

To find dot product:

$$\begin{aligned}\text{dp} = \text{GP\_Par}_N \cdot \text{Par\_Chld}_N + \text{GP\_Par}_E \cdot \text{Par\_Chld}_E \\ + \text{GP\_Par}_U \cdot \text{Par\_Chld}_U\end{aligned} \tag{46}$$

To find magnitudes:

$$\text{mag}_{GP\_Par} = \sqrt{GP\_Par_N^2 + GP\_Par_E^2 + GP\_Par_U^2} \tag{47}$$

$$\text{mag}_{Par\_Chld} = \sqrt{Par\_Chld_N^2 + Par\_Chld_E^2 + Par\_Chld_U^2} \tag{48}$$

To find angle ε:

$$\cos\varepsilon = \frac{dp}{\text{mag}_{\text{GP\_Par}} \cdot \text{mag}_{\text{Par\_Chld}}} \tag{49}$$

$$\varepsilon = \arccos(\cos\varepsilon) \tag{50}$$

4.2 Map representation

In our study, the 3-D environment is modelled as large, bounded airspace with dimensions of 350x350x150 metres. This defined space simulates a real-world environment where UAVs need to navigate, such as in urban or controlled airspace. To effectively manage pathfinding within this environment, we constructed a grid-graph representation of the map. The grid is created by placing into the airspace the evenly spaced points, with intervals set to one-fifth of the map's total length. Each point on this grid becomes a node in our graph, and these nodes are interconnected by edges to their nearest neighbours, forming a lattice structure.

The start and goal positions for the UAVs are not just placed arbitrarily; they are generated using a normal probability distribution over the original continuous map. After the grid-graph map is built, these start and goal positions are incorporated into the graph as new nodes. We then create edges connecting these nodes to their nearest neighbours in the grid, integrating them seamlessly into the existing structure.

This method allows us to simulate a wide range of possible flight paths and conflict scenarios, providing a robust foundation for testing our pathfinding and collision avoidance algorithms. By combining a structured grid with randomly generated positions, we ensure that the UAVs are challenged to navigate efficiently and safely through a realistic 3-D environment.

4.3 Pathfinding technique

For efficient pathfinding in the 3-D environment, we implemented the theta-star algorithm, an advanced variation of the A-star algorithm tailored to handle the complexities of three-dimensional space. Unlike A-star, which strictly follows the grid's structure and generates paths constrained by the grid's edges, theta-star introduces the concept of line-of-sight between nodes. This capability allows theta-star to bypass unnecessary intermediate nodes by directly connecting a node to any visible successor, effectively "cutting corners" to produce shorter and smoother paths.

The main advantage of theta-star lies in its ability to reduce path length and optimise the path's smoothness, which is particularly important in 3-D environments where UAVs must navigate around static and dynamic obstacles. The algorithm operates by relaxing the parent-child relationship between nodes seen in A-star, allowing a node to be directly connected to its grandparent if the direct path is clear, thus reducing the number of turns and deviations in the trajectory. This leads to more efficient paths that minimise the UAVs' travel time and energy consumption.

In this application, the theta-star algorithm is integrated with a flight dynamics model, which accounts for the most essential physical constraints and manoeuvrability of the UAVs. This ensures that the paths generated are not only near-optimal in terms of distance but also feasible given the UAVs' dynamic capabilities. By combining theta-star with our flight dynamics model, we generate paths that are both computationally efficient and practical for real-world navigation, especially in complex environments with obstacles such as buildings or other UAVs.

4.4 LoS check

Each UAV in our system has a separation volume, modelled as a 4-D rectangle that spans both 3-D space and time. This separation volume is directed between two points the UAV will visit, and it consists of three parts: a middle segment, where the UAV is actively travelling, which is 10 metres long, and forward and back segments of the same length, this makes the total of 30 metres in length. These additional segments provide extra safety by extending the protected area around the UAV, reducing the risk of a LoS with other UAVs.

During the trajectory planning phase, the theta-star algorithm evaluates potential paths by examining the edges between different nodes. The theta-star checks if the edge between two nodes intersects any already reserved separation volumes by other UAVs. If the edge does not intersect a reserved volume, the transition to the next node in the theta-star algorithm is considered safe. If the edge is part of the optimal path, the associated separation volume is then reserved to prevent other UAVs from planning trajectories that would conflict with it.

Collision detection between separation volumes is managed by placing collision spheres along the long axis of the directed 4-D rectangle. These spheres have a radius equal to the separation volume width which is 10 metres and are spaced 1 metres apart. If any of these collision spheres from one separation volume intersects with those from another, and the separation volumes overlap in time, a LoS violation is detected. When this happens, the next node in the path is ignored, indicating an obstacle along the UAV's route.

5 SIMULATION RESULTS

We planned and performed 25,000 flights in 1,000 simulations for 25 rotary-wing UAVs flying in the shared airspace 350x350x150 metres with 5 skyscrapers in standard atmosphere (Figure 1). UAV flight characteristics and all positions were random for each case. For example, the mass of drones ranged from 2 to 500 kg. As a result, the Gen4jectory algorithm successfully planned the uncrewed air traffic for all cases. The LoS case was not registered for any flight which confirms the capability of the algorithm to plan collision-free 4-D trajectories based on the unique flight characteristics of the rotary-wing aircraft.

Simultaneously, the experimental setup calculated the maximum airspeed for every UAV based on its unique aircraft performance data - drag coefficient, maximum thrust, cross-sectional area, and mass. The interesting finding was that a physical model of the Gen4jectory algorithm gave flight time calculations very close to the calculation considering only maximum airspeed. The deviation was about 0-2%.

The calculations to find the maximum speed in level flight can be done before a flight. However, the calculated maximum speed can significantly differ from the value in the flight manual, for example, due to changes in air temperature, and/or aircraft mass. Also, drag coefficient and cross-sectional area can differ significantly due to the outboard cargo.

6 CONCLUSION

Based on the experimental results, we conclude that the Gen4jectory algorithm can plan 4-D trajectories for rotary-wing UAVs, even in congested urban environments, within the assumptions we made. The algorithm guarantees that all trajectories will be planned without LoS in a near-optimal way considering the unique aircraft performance data of every drone.

The algorithm realised the idea of equal access to the airspace. It implies that the sooner a UAV requests a 4-D trajectory, the more optimal trajectory the drone receives from the algorithm. Nevertheless, we expect further research on

the holistic optimisation of uncrewed air traffic as an alternative to our approach.

Another path for further research can be an optimisation of the algorithm using precalculated maximum speed instead of numerical methods for flight time calculations. As we know from the experiment, the difference is about a few percent. However, we can expect significant optimisation in computation time.

REFERENCES

[1] McKinsey & Company. Study on the societal acceptance of Urban Air Mobility in Europe. *European Union Aviation Safety Agency*, May 2021. Accessed: Jul. 06, 2022.

[2] I. Panov and A. Ul Haq. A Critical Review of Information Provision for U-Space Traffic Autonomous Guidance. *Aerospace*, 11(6), 2024.

[3] Single European Sky ATM Research 3 Joint Undertaking (EU body or agency). *European drones outlook study: unlocking the value for Europe*. Publications Office of the European Union, 2017. Accessed: Aug. 27, 2024.

[4] A. Islami, S. Chaimatanan, and D. Delahaye. *Large-Scale 4D Trajectory Planning*, pages 27–47. Springer Japan, Tokyo, 2017.

[5] M. Sun, K. Rand, and C. Fleming. 4 Dimensional waypoint generation for conflict-free trajectory based operation. *Aerospace Science and Technology*, 88:350–361, May 2019.

[6] E. Gallo, J. Lopez-Leones, M. A. Vilaplana, F. A. Navarro, and A. Nuic. Trajectory computation Infrastructure based on BADA Aircraft Performance Model. In *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, pages 1.C.4–1–1.C.4–13, Dallas, TX, USA, 2007. Institute of Electrical and Electronics Engineers.

[7] S. Tang, K. Sreenath, and V. Kumar. *Multi-robot Trajectory Generation for an Aerial Payload Transport System*, pages 1055–1071. Springer International Publishing, Cham, 2020.

[8] Q. He, R. Wang, Z. Wang, and Y. Wang. A Dynamic Trajectory Planning Algorithm for Urban Ultra-low Altitude UAVs Based on a Fusion Heuristic Algorithm. In *2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 148–153, May 2023.

[9] Y. Zuqiang, F. Zhou, and L. Ping. A Bio-inspired Collision-free 4D Trajectory Generation Method for Unmanned Aerial Vehicles Based on Tau Theory. In *2015 34th Chinese Control Conference (CCC)*, pages 6961–6968, Jul. 2015.

[10] S. Aggarwal and N. Kumar. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 149:270–299, Jan. 2020.

[11] C. A. P. Neto, G. de C. Bertoli, and O. Saotome. A-star path planning simulation for UAS Traffic Management (UTM) application, Jul. 2021.

[12] X. Lin, C. Wang, K. Wang, M. Li, and X. Yu. Trajectory planning for unmanned aerial vehicles in complicated urban environments: A control network approach. *Transportation Research Part C: Emerging Technologies*, 128:103120, Jul. 2021.

[13] F. Ho et al. Decentralized Multi-Agent Path Finding for UAV Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):997–1008, Feb. 2022.

[14] K. Daniel, A. Nash, S. Koenig, and A. Felner. Theta*: Any-Angle Path Planning on Grids. *jair*, 39:533–579, Oct. 2010.

[15] A. Nash, S. Koenig, and C. Tovey. Lazy Theta*: Any-Angle Path Planning and Path Length Analysis in 3D. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, Jul. 2010.

[16] M. Yuan, T. Zhou, and M. Chen. Improved lazy theta* algorithm based on octree map for path planning of UAV. *Defence Technology*, 23:8–18, May 2023.

[17] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan. A Systematic Literature Review of A* Pathfinding. *Procedia Computer Science*, 179:507–514, Jan. 2021.

[18] J. R. Sánchez-Ibáñez, C. J. Pérez del Pulgar, and A. García-Cerezo. Path Planning for Autonomous Mobile Robots: A Review. *Sensors*, 21(23), Jan. 2021.

[19] M. Faria, A. S. Ferreira, H. Pérez-Leon, I. Maza, and A. Viguria. Autonomous 3D Exploration of Large Structures Using an UAV Equipped with a 2D LIDAR. *Sensors*, 19(22), Jan. 2019.

[20] M. Xue and M. Wei. Small UAV Flight Planning in Urban Environments. In *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, 2020.

[21] R. Rey, J. A. Cobano, L. Merino, and F. Caballero. Adaptation of Lazy-Theta* for UAS 3D path planning considering safety costs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 387–393, Jun. 2021.

[22] X. He, F. He, L. Li, L. Zhang, and G. Xiao. A Route Network Planning Method for Urban Air Delivery. *Transportation Research Part E: Logistics and Transportation Review*, 166:102872, Oct. 2022.

[23] M. Faria, R. Marín, M. Popović, I. Maza, and A. Viguria. Efficient Lazy Theta* Path Planning over a Sparse Grid to Explore Large 3D Volumes with a Multirotor UAV. *Sensors*, 19(1), Jan. 2019.

[24] M.-H. Do, C.-E. Lin, and Y.-C. Lai. Validation of the Flight Dynamics Engine of the X-Plane Simulator in Comparison with the Real Flight Data of the Quadrotor UAV Using CIFER. *Drones*, 7(9), Sep. 2023.

[25] D. Mandloi, R. Arya, and A. K. Verma. Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment. *International Journal of System Assurance Engineering and Management*, 12(5):990–1000, Oct. 2021.

[26] M. Boumediene and I. Panov. Gen4jectory algorithm and experimental setup, Sep. 2024. Python, https://gitlab.com/uaswork/gen4jectory.