CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-024-00753-7

ORIGINAL PAPER

Constrained path planning for manned–unmanned rotorcraft teaming in emergency medical service missions

Francesca Roncolini[1] · Giovanni Galante[1] · Giuseppe Quaranta[1] · Pierangelo Masarati[1]

Received: 16 March 2023 / Revised: 17 May 2024 / Accepted: 17 June 2024
© The Author(s) 2024

Abstract
This paper investigates the path-planning problem applied to an innovative Unmanned Air Vehicle teaming with a helicopter to increase safety during Helicopter Emergency Medical Services operations. The unmanned vehicle, a drone that optionally can be launched from the helicopter, has the mission to explore the area of operation to determine the meteorological and environmental conditions and to detect physical obstacles. It is initially found that the combination of probabilistically optimal Rapidly-exploring Random Tree (RRT*) as the global planner and of Bidirectional Rapidly-exploring Random Tree (BiRRT) as the local planner provides a nearly optimal global path and a rapid replanning in case new obstacles are detected. Adopting a Savitzky–Golay filter in an optional post-processing phase enables trajectory smoothing, thus improving its practicability. The feasibility of the identified trajectory for a rigid-body helicopter model is assessed by computing a first estimate of attitude, forces, control inputs, and rotor power from the trajectory points and curvature. This assessment shows that the RRT * used as a local planner provides replanned trajectories more feasible than BiRRT with comparable computational times.

Keywords HEMS · UAV · Path planning

Abbreviations

AAA	Anywhere, anytime, in all-weather conditions
AAM	Advanced air mobility
ABC	Artificial Bee colony
ACO	Ant colony optimization
ANN	Artificial neural network
BA	Bat algorithm
BiRRT	Bidirectional RRT
CFIT	Controlled flight into terrain
CONOPS	Concept of operations
DRL	Deep Reinforcement learning
DTED	Digital terrain elevation data
FAA	Federal aviation administration
GA	Genetic algorithms
GNSS	Global navigation satellite system
HAA	Helicoper air ambulance
HEMS	Helicopter emergency medical services
LALT	Low altitude operations
LOC	Loss of control
MILP	Mixed-integer linear programming
MUM-T	Manned–unmanned teaming
NED	North-East-down
NLP	Non-linear programming
PinS	Point in space
PIC	Pilot in command
PRM	Probabilistic RoadMap
PSO	Particle swarm optimization
RRT	Rapidly-exploring random tree
RRT*	Improved version of RRT (see text)
SAR	Search and eescue
SRTM	Shuttle radar topography mission
TPP	Tip path plane
UAV	Unmanned air vehicle
UIMC	Unintended instrumental meteorological conditions
USGS	United States Geological Survey
VD	Voronoi diagrams
VFR	Visual flight rules

✉ Francesca Roncolini
francesca.roncolini@polimi.it

Giovanni Galante
ggalante95@gmail.com

Giuseppe Quaranta
giuseppe.quaranta@polimi.it

Pierangelo Masarati
pierangelo.masarati@polimi.it

[1] Politecnico di Milano, Milan, Italy

Published online: 02 July 2024

VG Visibility graph
VMC Visual meteorological conditions

1 Introduction

1.1 Context description

In modern society, Helicopter Emergency Medical Service (HEMS)—or Helicopter Air Ambulance (HAA), as per FAA's Advisory Circular 135-14B [1]—missions are part of the trauma management systems and health care [2]. The deployment of HEMS/HAA in sparsely populated and rural areas may be essential to allow a fast transport of patients who are in danger of life and the rapid availability of a competent medical crew.

HEMS and Search and Rescue (SAR) missions are typically Low Altitude Operations (LALT) that must be performed according to Visual Flight Rules (VFR), which in turn need appropriate Visual Meteorological Conditions (VMC). However, such conditions are not always available. The sudden deterioration of weather is not uncommon in mountainous areas and may lead to flight into Unintended Instrumental Meteorological Conditions (UIMC) , hence to mission abortion, with an impact on the rescue timing or, in the worst case scenarios, to the danger of collisions, Controlled Flight Into Terrain (CFIT), and Loss of Control (LOC).

In 2018, the Federal Aviation Administration (FAA) reported that UIMC and LALT represented two of the three main causes of helicopter accidents [3]. To achieve the reliability level expected for HEMS and SAR operations, the involved rotorcraft must ensure operability as close as possible to Anywhere, Anytime, in All-weather conditions (AAA).

Currently, the feasibility of the mission is evaluated by the pilot-in-command (PIC) based on the available weather bulletins and the analysis of the meteorological situation at the departure base, combined with the experience and knowledge of the characteristics of the mission area, using this information to correlate the weather conditions at the departure station with those at the site of operations.

Nevertheless, this information does not provide complete and reliable knowledge of the weather and obstacles in the mission area. To safely fly AAA, the pilot and the crew must be provided with the most accurate and complete information possible.

1.2 HEMS+ Scout Drone Project

To reach the goal of providing the pilot and the crew with the most accurate and complete information possible, the Italian technical university Politecnico di Milano is collaborating with the industry to develop and test innovative solutions based on the cooperation of the helicopter with a UAV. The drone, in this case , is used as a system that, through a series of sensors, can detect the presence of not-mapped obstacles, dangerous weather conditions, or other elements that can contribute to increasing the mission risks. Substantially, the drone would fill the lack of weather radars in the remote sites that are typically the scene of HEMS and SAR missions.

The HEMS+ Scout Drone project is funded by the European Funds for Regional Development allocated to the Italian region Sardinia and is carried out by Politecnico di Milano in partnership with the Italian engineering companies ANT-X,[1] designer of the UAV, and TXT,[2] developer of the drone-helicopter interface and drone control station. The proposed UAV in this scope is called "Scout Drone" because it explores the mission area, increasing the crew's situational awareness. The project as a whole is described in detail in [4].

The operative missions proposed for the Scout Drone are the following:

1. Detection of weather or environmental conditions in the areas subjected to helicopter rescue operations or in any other area where information relevant for the completion of the mission needs to be known in advance;
2. Detection and verification of all the elements (deterioration of Global Navigation Satellite System (GNSS) signal, physical obstacles) that may constitute a danger to flight safety along a route designated to become a Point in Space (PinS) route[3] to allow its quicker and cheaper certification.

This paper investigates the first operative mission and, in particular, the possibility of using the environmental data collected by the drone as input to an automatic path planner that re-plans the trajectory according to the meteorological and physical obstacles detected.

Within this project, we intend to develop a control station to plan the route that the helicopter involved in the HEMS operation needs to follow. This route can be flown by helicopter only if VMC are guaranteed. However, if the possibility of flying into UIMC is foreseen, the crew may deploy the drone using the hoist and put it into operation. The drone can follow the planned route while sensing the GNSS signal level, turbulence, and cloud ceiling and detecting the physical obstacles, such as high-voltage pylons. The data collected by the drone can then be sent to the helicopter

[1] https://antx.it/ , last accessed February 2024.

[2] https://www.txtgroup.com/ , last accessed February 2024.

[3] A PinS route is a route between points in space defined through GNSS which can be flown in IMC, whereas flight from and to heliports and PinS must be accomplished in VMC.

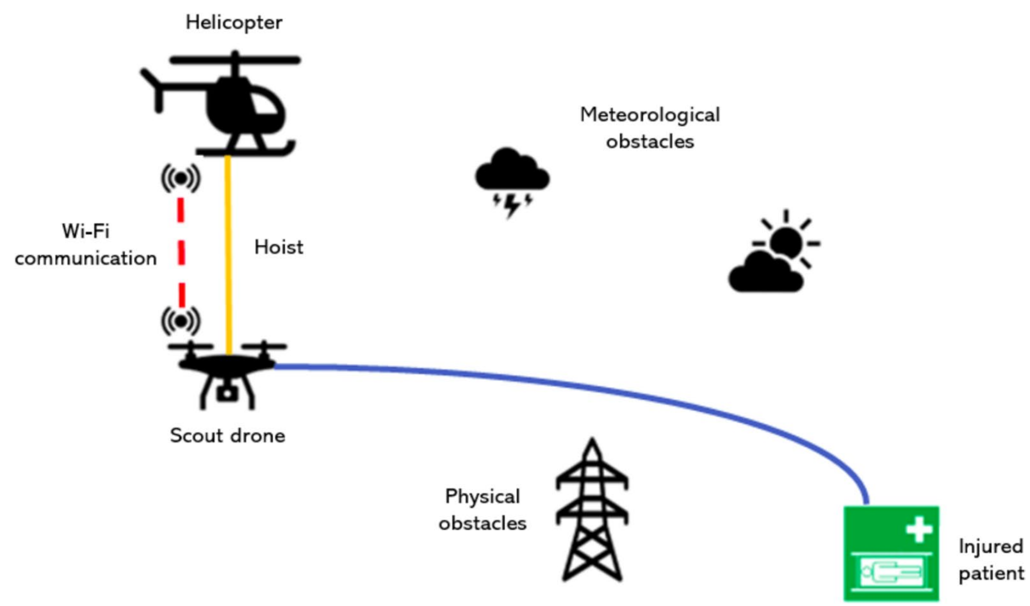

Fig. 1 Schematic cooperative HEMS mission. The scout drone is released from the helicopter using the hoist and flies the reference route, detecting unknown obstacles in the area

user interface, increasing the crew's situational awareness by observing the obstacles in the area. The data is also sent to the control station to update the maps for path planning. The automatic path planner checks the validity of the reference route at a predefined frequency. In light of the possible newly detected weather conditions or obstacles, it re-plans the route until the drone has explored the entire mission space and a successful and safe path has been found.

This Concept of Operations (CONOPS) is shown in Fig. 1. Other CONOPS that involve the drone being launched and/or operated by ground vehicles or stations are discussed in [4].

1.3 Objective of the paper

Path planning for air vehicles operating in low-altitude environments is an active research field. For instance, advanced air mobility (AAM) systems that are envisioned to fly autonomously in urban environments must develop path-planning strategies that not only avoid obstacles and obstructions, but also manage conflicts in high-density operational environments [5].

The objectives of this work are to describe and motivate the strategy identified to perform the path planning, briefly explain the implementation of the path planning code, and illustrate a methodology to assess the quality and feasibility of the computed trajectory, as discussed in [6], which this paper extends. Section 2 contains a review of path planning algorithms and describes the criteria of algorithm selection and some methods to assess the feasibility of the trajectory. Section 3 features a description of the adopted testing procedure and discusses the results of the tests.

2 Methods and algorithms

The most advanced and reliable path-planning algorithms are categorized and compared in Sect. 2.1, and a family of planners is selected. Path planning algorithms provide a sequence of waypoints. They are connected through Dubins curves to obtain a smooth trajectory. Alternatively, the trajectory can be smoothed via filtering (see Sect. 2.2). In Sect. 2.3, once the trajectory is defined in terms of time and position, its feasibility is tested through simulations. This evaluation is based on two indicators: helicopter attitude and rotor power required to follow the planned path.

2.1 Review of algorithms and planner selection

Before choosing a path planning algorithm, it is fundamental to analyse the problem and consider all the possible options. A careful review of the available algorithms increases the chances of choosing one that best meets the mission requirements.

First of all, it is convenient to define some essential terminology:

– Global path planning: in global path planning, path searching takes place in a known environment. The reference path is computed offline: there are no particular restrictions on the path generation time because the goal is to reach length optimality (shortest path), which implies longer computations.

– Local path planning: in local path planning, path searching takes place in a completely or partially unknown environment. The path is computed in real-time, updating the environment map with sensed data and re-planning the path in case of collision detection. The focus is on path generation time, which must be as short as possible because of the real-time computation requirement.
– Real-time reactivity: a local planner is real-time reactive when it has very fast collision avoidance capability, with reaction time typically < 200 ms (of course the appropriate reference figure depends on the application context), implying that if a new obstacle is detected very near to the helicopter while it is heading towards the obstacle, the control station can plan a new path in time to avoid the collision.

The problem can now be formulated according to the path planning terminology: after receiving an emergency call, the control station runs a global path planning algorithm to generate a reference path based on known environment maps. At the end of this operation, the helicopter flies the reference path until unsafe conditions are foreseen. The scout drone is therefore released to explore the reference route, looking for undetected physical or meteorological obstacles in the area and sending the information to the control station, which continuously updates the Occupancy Map (a digital map containing environment data) and checks the trajectory validity. If the original trajectory collides with a newly found obstacle, a new path is planned using a local path planning algorithm. Real-time reactivity is not strictly essential in this application as long as, for safety reasons, the scout drone has a sufficient head start from the helicopter; nevertheless , a shorter computation time is preferred, given the intrinsic urgency of HEMS missions.

Furthermore, the computed path should be constrained by the helicopter performance: a proper range of attitude, rates, and acceleration is typically prescribed to guarantee passengers' comfort while preventing the aircraft from exceeding its flight envelope limits.

2.1.1 Algorithms review

An overview of existing path planning algorithms is presented, based on the analysis of recent studies, which are evaluated based on the requirements of the HEMS helicopter-UAV teaming application. Table 1 contains a classification of the algorithms according to their family (Family, as detailed below), reference (Reference), year of publication (Year), offline (Offline) and/or real-time (Real-Time) applicability, re-planning options (Re-Plan), reactivity (Reactive), the inclusion of performance constraints in path planning (Perf. Constraints), post-processing for trajectory smoothing (Post-Proc.), validation performed by simulation or experiments (Sim/Exp), and static and/or dynamic obstacles management (St/Dyn Obstacles).

– Sampling-based: these algorithms systematically explore the space of operations, often privileging paths that result in a reduction of the distance from the target, until a sufficiently short path (not necessarily the shortest one) is identified. They are structured in two phases. During the learning phase, they build a road map by randomly generating a finite number of nodes in the free space and connecting them using collision-free segments; during the query phase, the algorithm finds the path from a start node to a goal node inside the road map. These methods are mature, of simple structure and easy to implement, suitable for both global and local planning. Some algorithms belonging to this family are Visibility Graph (VG) [7–10], Voronoi Diagrams (VD) [11], Probabilistic RoadMap (PRM) [12], and Rapidly-exploring Random Tree (RRT) [13–16].
– Graph-based: they search the least-cost path through the available grid points in a graph previously built from the given start to goal nodes. They are well mature algorithms, easy to implement, and are often combined with other methods to achieve global optimal solutions. They can be applied both real-time and offline. Some algorithms of this class are: Dijkstra [17], A^* [18, 19], D^* [20], and θ^* [21].
– Numerical optimization: they mathematically model the environment as well as the body, considering kinematic, dynamic, environmental, and mission constraints and binding a cost function to all constraint equations to achieve an optimal solution. They are computationally expensive, in particular when constraints grow in number and complexity, and therefore especially implemented in global planning, when the focus is on optimality. Examples of this class of algorithms are Mixed-Integer Linear Programming (MILP) [22] and Non-Linear Programming (NLP) [23].
– Bio-inspired: they optimize paths based on rules and considerations that mimic some biological behaviour. Up to date, these methods are still the subject of research. They are often rather complex and their long iteration time makes them suitable only for global planning. Some examples are Genetic Algorithms (GA) [24], Artificial Neural Network (ANN) [25], Particle Swarm Optimization (PSO) [26], Artificial Bee Colony (ABC) [27], Ant Colony Optimization (ACO) [28], Bat Algorithm (BA) [29], and Deep Reinforcement Learning (DRL) [30].

Table 1 Path planning algorithms classification

Family	Approach	References	Year	Offline	Real-time	Re-plan	Reactive	Perf. constraints	Post-proc.	Sim/exp	St/dyn obstacles
Sampling based	Visibility graph	[7]	2019	x	x	x		–	–	S	S
		[8]	2020		x	x		x	x	S	S/D
		[9]	2018	x				x	x	S	S
		[10]	2017	x				x		S	S
	Voronoi	[11]	2017	x	x	x		x		S	S/D
	PRM	[12]	2013	x				x	x	S/E	S
	RRT	[13]	2021	x	x	x		x	x	S	S/D
		[14]	2014	x	x	x		x	x	S/E	S/D
		[15]	2017	x						S	S
		[16]	2019	x	x	x		x		S	S/D
Graph based	Dijkstra	[17]	2016		x	x				S	S/D
	A^*	[18]	2020		x	x		x		S	S/D
		[19]	2011	x		x				S	S
	D^*	[20]	1993								
	θ^*	[21]	2010								
Numerical optimization	MILP	[22]	2017		x			x		S	S/D
	NLP	[23]	2009	x				x		S	S
Bio-inspired	GA	[24]	2020	x				x		S	S
	ANN	[25]	2014	x				x		S	S
	PSO	[26]	2018		x					S	S/D
	ABC	[27]	2007								
	ACO	[28]	2013	x						S	S
	BA	[29]	2019	x				x		S	
	DRL	[30]	2020		x	x	x	x	x	S/E	S/D
Fusion	PSO + D^*	[31]	2018		x	x				S	S/D
	A^* + GA	[32]	2020	x	x	x	x	x		S	S
	MPC + PSO + RRT	[33]	2021		x	x		x		S	S/D
	PF + A^*	[34]	2011	x						S	S
	PRM + ABC	[35]	2021		x	x		x		S	S/D
	MILP + A^*	[36]	2020	x				x		S	S

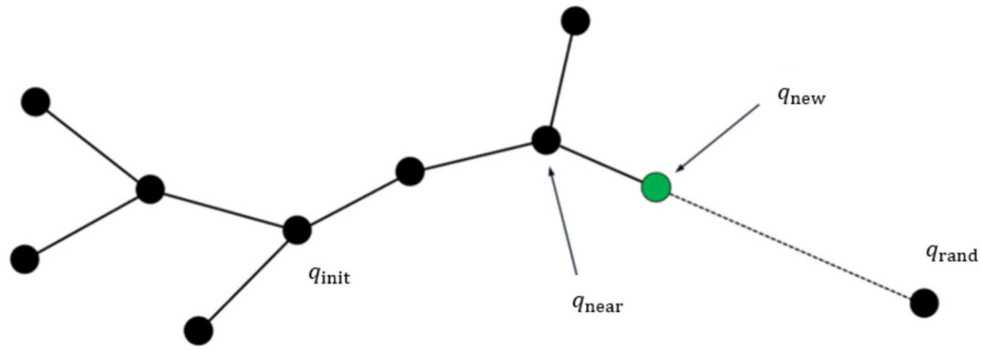

Fig. 2 RRT approach

– Fusion: they result from the combination of methods belonging to the previously mentioned categories to complement their features and achieve optimal routes and minimum computational cost, as proposed by several authors; for example, PSO and D* [31], A* and GA [32], MPC, PSO, and RRT [33], PF and A* [34], PRM and ABC [35], and MILP and A* [36].

2.1.2 Algorithm selection: RRT

Given the safety requirements of HEMS tasks, a mature and consolidated algorithm with proven applications in real contexts should be selected. Moreover, the scout drone-helicopter path planner requests both offline and real-time capability, re-planning possibility, inclusion of performance constraints in the computation and a post-processing phase to smooth the trajectory.

In light of the above mentioned considerations and of the information presented in Table 1, the sampling-based Rapidly-exploring Random Tree algorithm is selected. Two improved versions are also considered: RRT* and BiRRT (bidirectional RRT).

2.1.3 RRT

The algorithm's name, Rapidly-exploring Random Tree, refers to its particular path-searching technique. As explained below, it creates a structure of segments connecting the nodes. This structure resembles a tree with many branches. The tree is constructed incrementally from samples drawn randomly from the search space, as explained in [37], which also contains the algorithm's pseudocode.

This algorithm organizes the environment as an occupancy grid map, where information on occupancy (occupied/free) is stored in every grid point. In this context, an "occupied" grid point contains an obstacle, and thus is unavailable for path planning purposes. The nodes of the tree are identified by states. A state is defined by its 3D position coordinates, x, y, z, and the heading of the vehicle, ψ, collected in the vector $q = \{x; y; z; \psi\}$.

The expansion of the tree, shown in Fig. 2, is described below. The starting node, representing the initial state of the helicopter, is the root of the tree, q_{init}. A random state q_{rand} in the state-space is selected during the *sampling* phase; the node of the existing tree that is nearest to the random state, called the nearest node q_{near} , is pinpointed in the *nearest node selection* phase.

At this point, the *node expansion* takes place. A maximum connection distance δ that the new state q_{new} can be separated from q_{near} is specified. If the distance of q_{rand} from q_{near} is less than δ, then $q_{new} := q_{rand}$ is selected; otherwise , a new node q_{new} is is created along the straight line that connects q_{near} to q_{rand}, at a distance δ from q_{near}. If an obstacle is present between q_{near} and q_{new}, the latter is not added to the tree and a new q_{rand} is selected, reiterating the process from the sampling phase.

The tree growth process ends when the path reaches a point that lies within a threshold of the goal.

2.1.4 RRT*

The RRT* algorithm is the probabilistically optimal extension of RRT. As the number of nodes in the tree grows to infinity, the probability of finding the optimal path converges to 1. The cost is an increased path generation time.

The difference between RRT and RRT*, shown in Fig. 3, lies in the nearest node selection and the stop criteria; indeed, not necessarily the nearest node ends up being connected with q_{new}, other nodes in a given search radius are also checked and could be selected if they are able to provide a shorter connection path. Furthermore, the process does not stop when the goal is reached but continues refining the path in search of shorter routes until the maximum number of iterations is achieved.

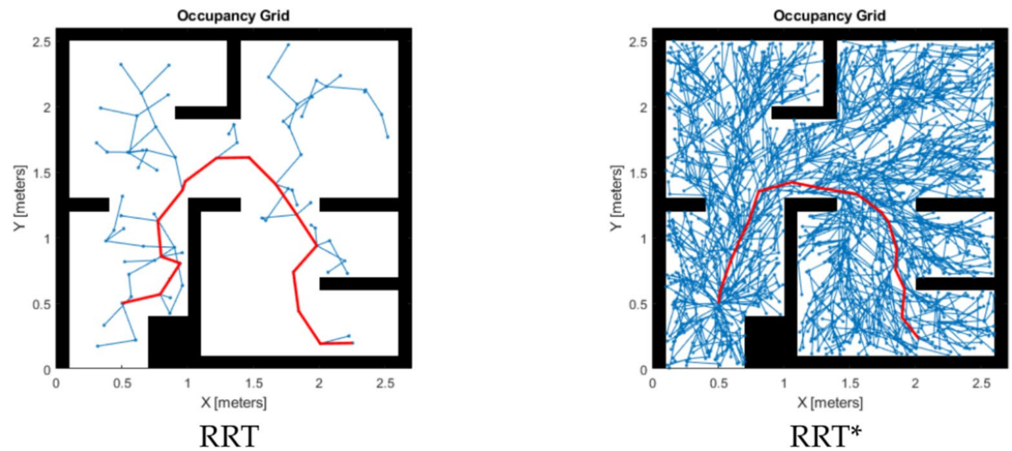

Fig. 3 Comparison between RRT and RRT*

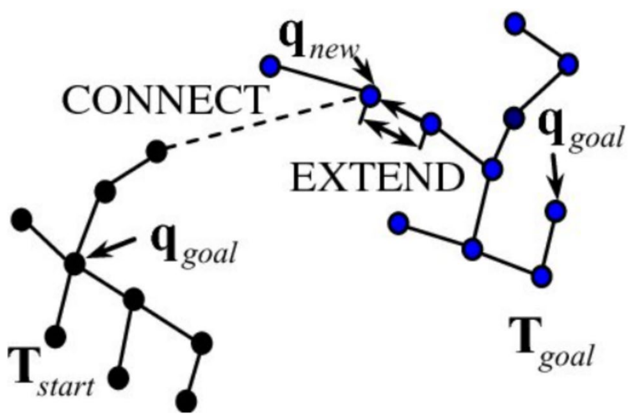

Fig. 4 Bidirectional rapidly-exploring random tree

2.1.5 BiRRT

The bidirectional RRT algorithm creates one tree with the root node at the specified start state, and another tree with the root node at the specified goal state, alternating the extension progress until the two trees connect. The connection can take place neglecting the maximum connection distance if a straight line can connect the two new nodes from the start and goal tree without impacting any obstacles. The process is shown in Fig. 4. This algorithm can be very fast, at the cost of sacrificing the asymptotic optimality of RRT*.

2.2 Implementation

The path planning code and the simulation environment have been implemented in MATLAB 2021b, leveraging the Matlab Navigation Toolbox, which features the occupancy map generation command and several built-in or customizable motion planning algorithms.

2.2.1 Map

The scenario is built in the shape of an occupancy map, which consists of a 3D grid of cells that can be either occupied or free. Each cell state (occupied/free) is determined according to the Digital Terrain Elevation Data (DTED), obtained from the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Archive. In particular, the data collected by the Shuttle Radar Topography Mission (SRTM)[4] are used, with a resolution of 1 arc-second. The DTED data consists of vectors $\{x;y;z\}$, where x and y are the latitude and longitude, and z is the corresponding terrain elevation. The occupancy map can be inflated to guarantee a safe distance from obstacles. The map used for the simulations is shown in Fig. 5. It corresponds to La Maddalena island, in northern Sardinia. It is further described in Sect. 3.2.

2.2.2 Planner

The developed code provides the user with the possibility to choose among three different planner types (i.e.: path planning algorithm), for both local and global planning: RRT, RRT* and BiRRT.

As illustrated in Fig. 6, the *planner* object has two inputs, the State Space and the State Validator. The State Space, further described in the next paragraph, represents all the possible states the helicopter can occupy according to performance and other constraints. The State Validator contains occupancy information about its state; namely, whether a state in the occupancy map is occupied or free.

Some properties of the planner can be prescribed:

[4] https://earthexplorer.usgs.gov/, last accessed October 2022.

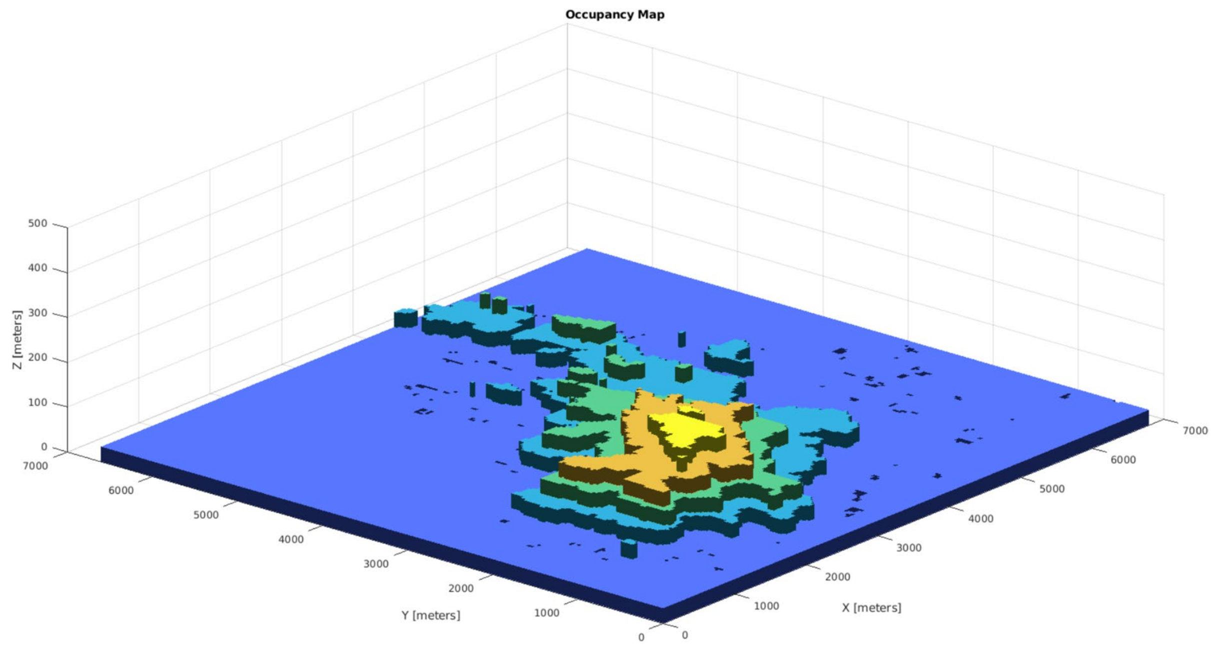

Fig. 5 Occupancy map of La Maddalena

```
planner = plannerRRT(StateSpace,StateValidator);

planner.MaxConnectionDistance = 10;
planner.GoalBias = 0.2;
planner.MaxIterations = 1000;

[pthObj,solnInfo] = plan(planner,startPose,goalPose);
```

Fig. 6 Planner settings and plan function

- maximum connection distance: the maximum distance δ between q_{near} and q_{new} prescribed during the node expansion phase;
- goal bias: a number between 0 and 1 that defines the level of orientation towards the goal area, during the sampling phase;
- maximum number of iterations: the maximum number of steps to achieve the goal;
- maximum number of nodes in the tree;
- callback function to define the threshold within which the goal is considered reached.

The function *plan* computes a path between two states using the selected planner, as shown in Fig. 6.

2.2.3 State space

A state space consists of all the feasible states of a vehicle during path planning. A state is intended as the elements $\{x;y;z;\psi\}$, where x and y are the position components in the horizontal plane, z is the altitude, and ψ is the heading angle of the vehicle. In Matlab, the state space is represented by the state space object constructed by the *nav.StateSpace* class. To impose compliance with performance constraints, a customized state space bounded by such constraints has been created using the function *createPlanningTemplate*. The adopted performance constraints are the maximum roll angle, the minimum and maximum flight path angle, and the airspeed. 3D Dubins curves have been used in node connection. They represent the shortest segments with prescribed maximum turning radius and flight path angle range that connect two states [14]. A segment of the path, planned with and without Dubins curves, is illustrated in Fig. 7.

2.2.4 Replanning/reconnecting

When the scout drone detects an obstacle, the Occupancy Map is updated with the new information, and the validity of the global path is checked using the Matlab functions *isStateValid* and *isMotionValid*. If at least one state along the path is invalid, a replanning or reconnecting of the path is necessary. The *replan* strategy deletes the path states after the current position and replans a path originating from it and ending at the goal state. The *reconnect* strategy truncates path states from the actual position to a user-defined number of states after the obstacle. The local planner then reconnects the current position to the truncated branch attached to the goal state. Generally, the use of the replan strategy results in a shorter path, while the reconnect strategy results in a shorter computational time. Reconnecting will be preferred in the first half of the path, where a full replan could imply

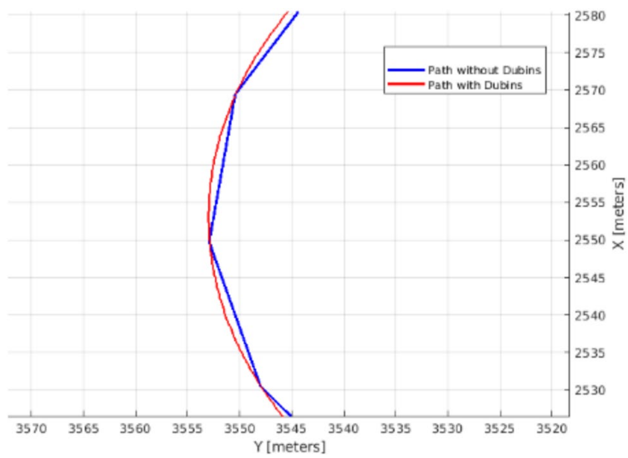

Fig. 7 Different types of node connections: with straight lines (blue); with Dubins curves (red)

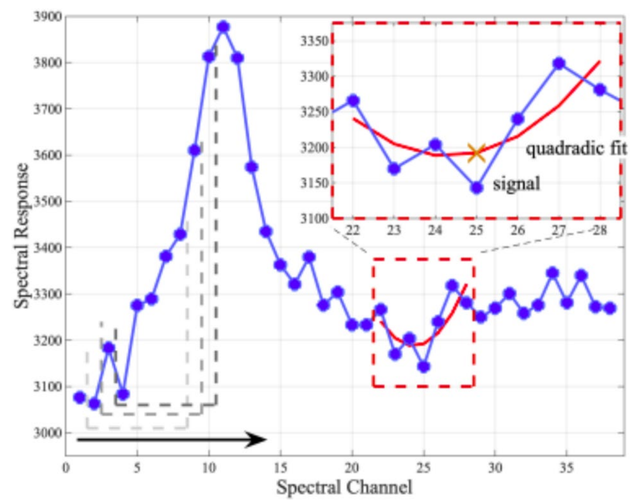

Fig. 8 Savitzky–Golay filter with window width = 7 and polynomial order = 2

an excessively slow computation. Replanning will be preferred in the second half of the path.

2.2.5 Smoothing

Although Dubins curves enforce compliance with performance constraints, they are only C^1-continuous, i.e., continuous and differentiable, with a continuous first derivative, implying that their second derivative, and therefore their curvature, are not necessarily continuous, thus resulting in uncomfortable and possibly unfeasible trajectories. For this reason, a further smoothing phase has been introduced using the Savitzky–Golay filter [38]. This filter is usually employed to smooth digital signals. It was chosen because of its simple implementation (Matlab features the designated function *sgolayfilt*). More specific path-smoothing techniques are presented in [39].

For a given signal measured at N points and a filter of window width w, the Savitzky–Golay filter computes a polynomial fit of order o in each filter window as the filter is moved across the signal. The filter estimation at the centre of each window is given by the polynomial fit at the centre point, as shown by the yellow cross in the subplot in the top right corner of Fig. 8. The lower the polynomial order and the higher the window width, the smoother the path at the price of precision loss.

2.3 Assessment of trajectory feasibility

At this point, a trajectory that guarantees obstacle avoidance and satisfies constraints on the minimum turning radius and maximum flight path angle has been planned. However, the trajectory needs to be validated. Trajectory feasibility can be assessed by computing the required helicopter attitude, rotor power, and control inputs. The computed attitude, in turn, can be compared with the prescribed maximum roll and flight path angles to establish if the path-planning algorithm complies with the performance constraints. Rotor power and control inputs can be checked to be within typical operational ranges. Those quantities are evaluated and assessed for each path waypoint.

2.3.1 Calculation of Euler angles

The Euler angles corresponding to pitch θ, roll ϕ and heading ψ (Fig. 9) are computed for each waypoint along the path to characterize the helicopter motion. These angles represent the three consecutive rotations ψ, θ, ϕ required to transform the North-East-Down (NED) reference frame into the Body reference frame. They describe the orientation of the helicopter. The fixed reference frame adopted up to this point for the trajectory computation is the East-North-Up (ENU), so a rotation from ENU to NED is applied before computing the Euler angles.

The following assumptions have been made for Euler angles computation:

– the velocity vector is contained in the helicopter longitudinal plane (no sideslip),
– when the trajectory is straight and uniform, the roll angle ϕ is equal to 0 deg,
– during curved portions of the trajectory, the Tip Path Plane (TPP) is assumed perpendicular to the yaw axis and fixed to the helicopter.

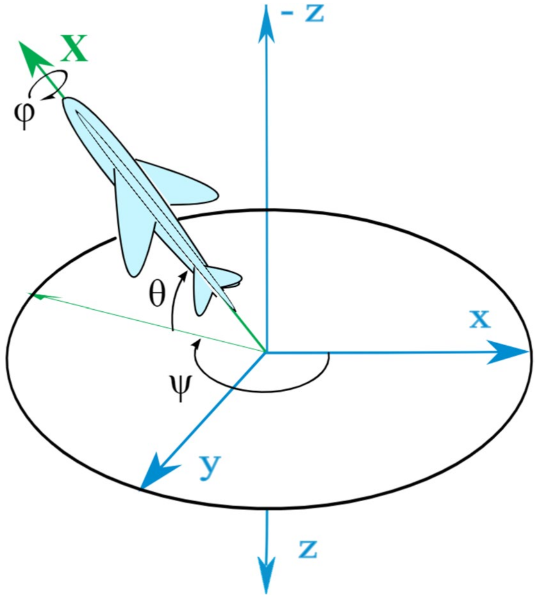

Fig. 9 Heading, pitch and roll angles (Z–Y'–X'') for an aircraft. The aircraft's pitch and yaw axes Y and Z are not shown, and its fixed reference frame xyz has been shifted backwards from its center of gravity (preserving angles) for clarity

2.3.2 Heading angle

If the helicopter is travelling from the ith to the $(i+1)$th trajectory waypoint, the first Euler angle is:

$$\psi(i) = \arctan\left(\frac{x(i+1) - x(i)}{y(i+1) - y(i)}\right), \tag{1}$$

where x and y are the East and North axes of the ENU frame, as indicated in Fig. 10.

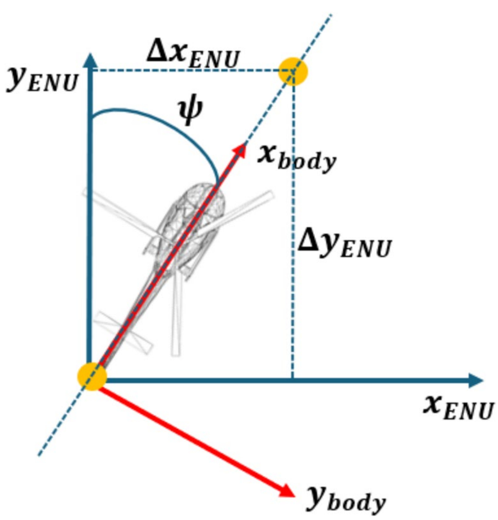

Fig. 10 Heading angle computation. The yellow circles are the waypoints among which the helicopter is traveling

2.3.3 Pitch attitude

The pitch attitude on a straight segment of the trajectory and during a steady turn is computed from the longitudinal trim equations (described in Sect. 2.3.3) and depends on the flight path angle and the airspeed.

However, it is very uncommon to have a perfectly straight trajectory: the trajectory is usually curved, and the curvature generates centrifugal forces. The thrust has to compensate not only the weight but also the centrifugal forces. This equilibrium is achieved by increasing and deflecting the thrust to perfectly oppose the sum of the forces. If the local curvature is known, the centrifugal forces are known and the magnitude and deflection of thrust with respect to the z_{ENU} axis can be computed. The third assumption (TPP normal to the yaw axis) allows us to obtain the helicopter attitude from the thrust orientation.

The problem of the orientation can be split into a pitch attitude problem and a roll attitude problem. The pitch attitude mainly depends on the curvature of the vertical projection of the trajectory, and the roll attitude depends on the curvature of the horizontal projection. The horizontal projection of the trajectory is the projection of the trajectory on the x_{ENU}–y_{ENU} plane. The vertical projection is less straightforward: with "vertical projection of the trajectory" here is intended the unwrapping of the trajectory on a sheet that is perpendicular to the x_{ENU}–y_{ENU} plane and tangent to the trajectory in each point. It can be thought of as a sheet which is initially wrapped around the trajectory and then unwrapped together with the trajectory, which will remain projected on the sheet. An example of this vertical projection can be visualized in Fig. 11, where a spiral curve is unwrapped and shows to have no vertical curvature. The curvilinear coordinate is the length of the horizontal projection of the trajectory up to the evaluated point. Therefore the vertical curvature is the curvature of the altitude of the trajectory as a function of the curvilinear coordinate.

The curvature of a trajectory in a point is the reciprocal of the osculating circle radius R_v in that point, therefore the trajectory projected on the vertical plane can be approximated in every point to a curvilinear manoeuvre whose radius is $1/k_v$. As one can observe in Fig. 12, the static equilibrium of all forces acting on the helicopter, including weight and the centrifugal force, is established at each trajectory waypoint. The thrust balances the weight and the centrifugal forces and is perpendicular to the TPP which, according to the third hypothesis, remains aligned with the pitch axis. From the radial and tangential components of the thrust (see Fig. 12) one can compute the angle α between the TPP and the airspeed (see Fig. 13, where α is indicated as α_{TPP}):

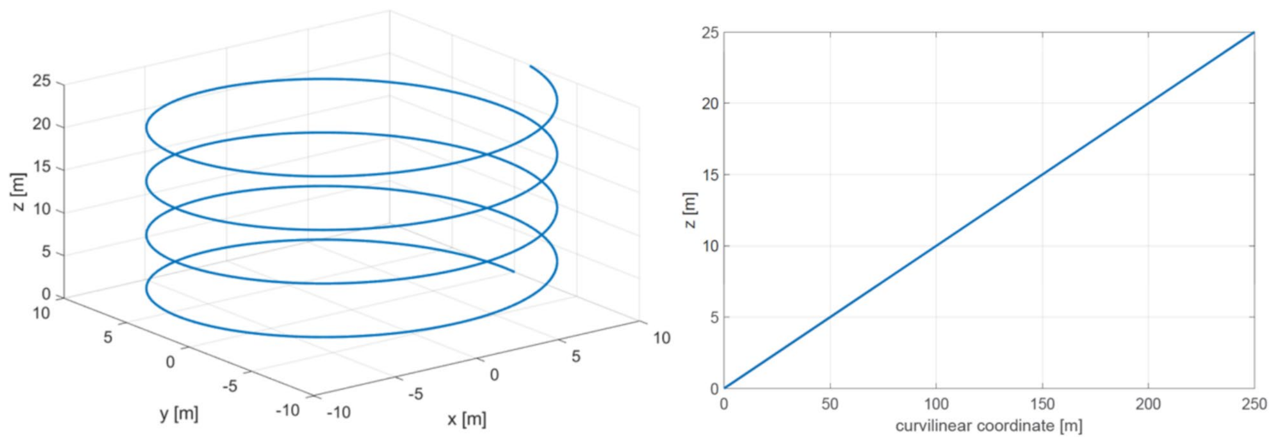

Fig. 11 On the left: spiral curve. On the right: spiral curve unwrapped. Altitude is a function of the curvilinear coordinate

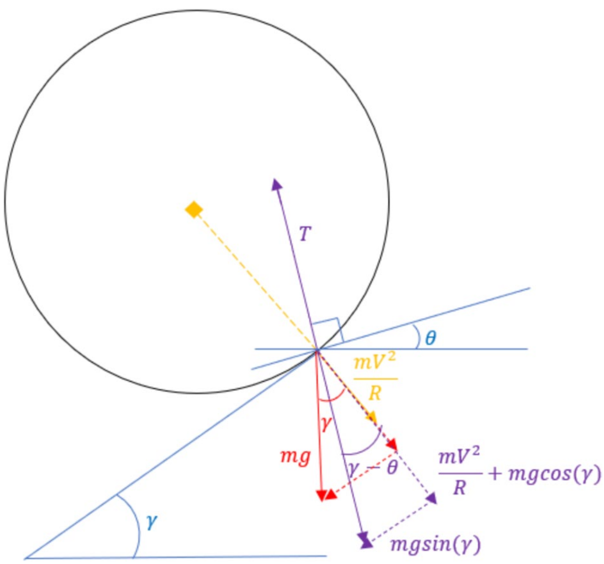

Fig. 12 Pitch computation: equilibrium of forces during pull-up. The black circle is the osculating circle tangent to the vertical projection of the trajectory. The yellow colour indicates the centrifugal force. The red colour indicates the weight force and the red dashed vectors are the weight components along the radial and tangential directions. The purple colour indicates the resultant of the weight and centrifugal forces. The thrust, which balances the resultant force, is also drawn in purple

$$\gamma - \theta = \alpha = \arctan\left(\frac{mg\sin\gamma}{\frac{mV_v^2}{R_v} + mg\cos\gamma}\right), \quad (2)$$

where γ is the flight path angle, namely the angle formed by the airspeed vector and the horizontal plane.

The formula for the computation of the curvature of a three-dimensional curve $\underline{r}(t) = \{x(t); y(t); z(t)\}$, parameterized by a

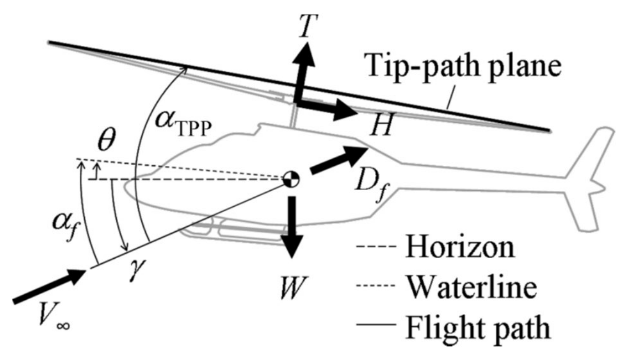

Fig. 13 Helicopter angles on the longitudinal plane

generic parameter, in this case , without loss of generality, the time t, was used:

$$k(t) = \frac{|\underline{r}'(t) \times \underline{r}''(t)|}{|\underline{r}'(t)|^3}. \quad (3)$$

In Eq. (3), $\underline{r}'(t)$ is the first derivative of the curve with respect to the parameter t, in this case the time, and $\underline{r}''(t)$ is the second derivative. Therefore, the curvature is not defined when $\underline{r}'(t) = 0$, which corresponds to the hovering situation. However, this is never the case because the airspeed of the helicopter has been assumed constant and different from zero during the simulations as a simplification, as explained in Sect. 3.3. In particular, to compute k_v a change of variables was operated to obtain the curvature of the vertically projected trajectory: k_v was computed using $\underline{r}_v(t) = \{v(t); z(t); 0\}$, where

$$v(t(N)) = \sum_{1}^{N} \sqrt{(x(t(i)) - x(t(i-1)))^2 + (y(t(i)) - y(t(i-1)))^2}, \quad (4)$$

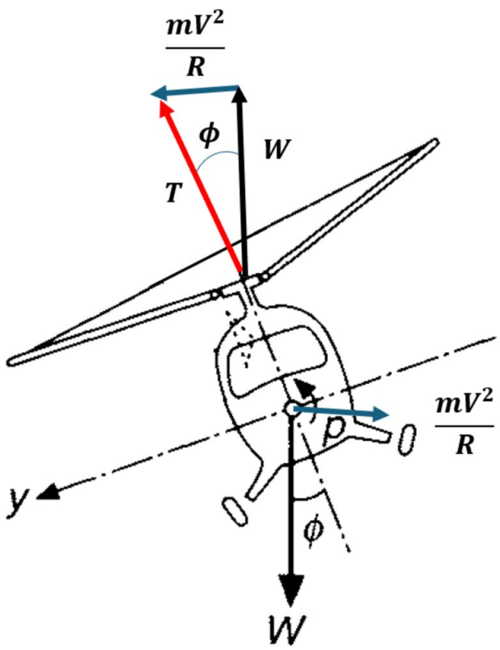

Fig. 14 Bank angle Φ during a turn

(that is the length of the horizontal projection of the trajectory up to the generic Nth point).

2.3.4 Roll angle

The procedure to obtain the bank angle Φ (see Fig. 14) is very similar to that for the computation of the pitch angle applied to the horizontal plane. The turn radius R_Φ is obtained from the horizontal curvature k_h, and the bank angle is computed as the angle opposed to the centrifugal force in Fig. 14:

$$\Phi = \arctan\left(\frac{(V\cos(\gamma))^2}{gR_\Phi}\right). \tag{5}$$

The roll angle, ϕ, is computed as a function of the bank and pitch angles, Φ and θ:

$$\sin\phi = \sin\Phi\cos\theta \tag{6}$$

2.3.5 Thrust computation

The thrust is computed as the force that balances the centrifugal forces and the weight. The thrust T_γ that compensates the weight force and centrifugal force of a vertical manoeuvre is the vectorial sum of the weight force and centrifugal force of the curved manoeuvre:

$$T_\gamma = \sqrt{\left(\frac{mV^2}{R_v} + mg\cos(\gamma)\right)^2 + (mg\sin(\gamma))^2}. \tag{7}$$

The total thrust must balance also the centrifugal force of the turning manoeuvre in the horizontal plane. Consequently, the thrust is equal to the vectorial sum of T_γ and the turn centrifugal force:

$$T = \sqrt{T_\gamma^2 + \left(\frac{m(V\cos(\gamma))^2}{R_\phi}\right)^2}. \tag{8}$$

2.3.6 Calculation of the control inputs and flapping angles

An iterative process for the trim computation is presented in [40] on page 198. Given the four prescribed trim states (flight speed, flight path angle, turn rate and sideslip angle) and initializing the unknown flight states (helicopter attitude, main rotor flapping angles and main and tail rotor inflow) one can compute the mentioned flight states and the control inputs (main rotor collective angle θ_0, main rotor longitudinal cyclic angle θ_{1S}, main rotor lateral cyclic angle θ_{1C} and tail collective angle θ_{0T}).

The trim states are known at each trajectory waypoint because the flight speed is prescribed, the flight path angle γ can be easily computed from the horizontal and vertical components of the distance between two consecutive waypoints, and the turn rate can be calculated from the flight speed V_∞ and the turn radius R_Φ. The sideslip angle can be assumed to be zero. Therefore, for each waypoint, the control inputs can be computed and compared to the operative ranges of the specific helicopter to assess the trajectory feasibility.

The complete trim calculation proposed can be limited to the longitudinal trim to decrease the computation complexity. The longitudinal trim equations are shown in Appendix A. They do not constitute a rigorous analysis but can be considered an acceptable approximation to estimate the rotor power, which depends on the advance and inflow ratios computed during the trim calculation and the collective input. The prescribed trim states for the longitudinal trim analysis are the flight speed and path angle. Considering an instantaneous equivalent weight equal to the (vectorial) sum of the actual weight and the centrifugal forces obtained in the previous Section, a steady approximation of the trimmed state in curved portions of the path can be obtained from the usual trim analysis in the longitudinal plane.

2.3.7 Calculation of the rotor power

To estimate the rotor power, one can use the formula provided by Leishman [41] in the forward flight performance chapter (page 163). The forward flight power is the sum of the induced power, the blade profile power and the parasitic power:

$$C_p = \frac{1.15C_T^2}{2\sqrt{\mu^2+\lambda^2}} + \frac{\sigma C_{d_0}}{8}(1+4.6\mu^2) + \frac{1}{2}\frac{f}{A}\mu^3, \tag{9}$$

where C_T is the thrust coefficient, μ is the advance ratio, λ is the inflow ratio, σ is the rotor solidity, C_{d_0} is the drag coefficient of the airfoil, f is the equivalent friction area of the helicopter, and A is the rotor disc area.

The power is obtained from the power coefficient as follows:

$$P = \rho A(\Omega R)^3 C_p, \tag{10}$$

where Ω is the rotor rotation speed, and R is the rotor radius.

3 Tests and results

3.1 Testing procedure

The tests presented in this Section consist of simulations intended to assess the performance of the selected path-planning algorithms. The simulations have been run on a Dell Inspiron 15 5510 notebook, provided with an 11th Gen Intel(R) Core(TM) i7-11390 H at 3.40GHz CPU, 16 GB RAM and Microsoft Windows 11 Home OS.

The testing procedure to assess the path planning algorithm is designated as follows: given a start point and a goal point, the algorithm shall compute a global route that connects them without colliding with environmental obstacles. If waypoints imposed by authorities are present, the algorithm runs iteratively and computes multiple consecutive routes from each prescribed waypoint to the subsequent one.

As could happen in real flights involving the scout drone, whose purpose is to send information to the helicopter about meteorological , and thus non-persistent by definition, or unmapped physical barriers, an obstacle that modifies the Occupancy Map is introduced at a certain point of the simulation. The algorithm shall update the Occupancy Map and quickly replan the path using a local planner, choosing one among the replanning and reconnecting strategies. The replanning strategy is used in these tests because the obstacle is placed towards the end of the path.

3.2 Test environment

The environment chosen for the tests is La Maddalena (Sassari), a small island (about 20 km^2) in Northern Sardinia; the related Occupancy Map is illustrated in Fig. 5. The following start, goal and waypoints were selected:

- start point (chosen randomly): latitude 41°13′45″, longitude 9°22′56″, altitude 107 m;
- goal point: latitude 41°13′15″, longitude 9°24′37″, altitude 107 m. It corresponds to a football field from which the patient could be safely transported to the island's hospital by road, as there is no helipad at the hospital;
- the trajectory must pass through a waypoint (chosen randomly): latitude 41°14′23″, longitude 9°23′53″, altitude 150 m.

For the sake of convenience, all angular coordinates have been converted into distances during the computations.

Table 2 Performance constraints

Maximum roll angle	±30°
Minimum flight path angle	−10°
Maximum flight path angle	10°

3.3 Tests description

Three tests are discussed in this Section. The first two aim to investigate the effect of some tunable parameters on the trajectory planning. These tunable parameters are the maximum connection distance of the path planning algorithm, the usage (or not) of a smoothing filter and the filter window width and polynomial order. These tests focus on global path planning. Their outcome is the tuning of the mentioned parameters.

The third test aims at assessing the quality of the trajectory—planned with the chosen global planning parameters, and replanned with a local planning algorithm to avoid an obstacle placed on the reference route—as described in Sect. 3.1.

The outputs evaluated for the choice of a combination of parameters over another in the first two tests are the helicopter attitude, the helicopter thrust and the length of the trajectory. The helicopter attitude is compared with the performance constraints summarized in Table 2. No real limitation is placed on the maximum thrust, as it is expected to be

Table 3 Characteristics of the trajectories compared in the first test, that investigates the effects of the maximum connection distance

Trajectories	Global planner	Local planner	Max connection distance (MCD, m)	Smoothing
A	RRT*	No	20	No
B	RRT*	No	200	No

Table 4 Characteristics of the trajectories compared in the second test, that investigates the effects of the smoothing filter

Trajectories	Global planner	Local planner	Max connection distance (MCD, m)	Smoothing	Window width	Polynomial order
A	RRT*	No	20	No	–	–
C	RRT*	No	20	Yes	27	3
D	RRT*	No	20	Yes	151	3

Table 5 Characteristics of the trajectory implemented in the third test, that assesses the obstacle avoidance capability

Trajectories	Global planner	Local planner	Max connection distance (MCD, m)	Smoothing	Window width	Polynomial order
E	RRT*	BiRRT	20	Yes	151	3

intrinsically limited by the available torque and power. The outputs evaluated to assess the quality of the final trajectory in the third test are the helicopter attitude, the rotor power and the collective input, which should be compared with the specific helicopter limits.

The tests are performed at a fixed airspeed of 30 m/s.

3.3.1 Test on maximum connection distance (MCD)

The first test concerns the global planning of a reference route from the start to the goal point, passing from the waypoint. The test focuses on the effect of the MCD parameter discussed in Sect. 2.1.2. In this test, two trajectories are compared, both planned with the RRT* algorithm, the first with a MCD of 20 m and the second with a MCD of 200 m. The trajectories are not smoothed. The characteristics of the trajectories compared in this test are summarised in Table 3.

3.3.2 Test on path smoothing

The second test analyzes the effect of path smoothing. The best of the two trajectories compared in the first test is smoothed using the Savitzky-Golay filter described in Sect. 2.1.2. Two combinations of the window width and filter polynomial order are compared. The best combination is the one that satisfies the performance constraints with lower oscillations of thrust and attitude.

The characteristics of the trajectories compared in this test are shown in Table 4.

3.3.3 Test on obstacle avoidance capability

In this test, the complete operation described in Sect. 3.3 is simulated. The trajectory is planned (and replanned) using the maximum connection distance and path smoothing chosen in the previous tests. The characteristics of the trajectory are summarized in Table 5.

3.4 Helicopter model

The helicopter model used to compute the attitude, the collective input and the rotor power is characterised by the parameters reported in Table 6.

Table 6 Helicopter model parameters

Mass (kg)	3100
Rotor radius (m)	5.4
Rotor blades	4
Non-dimensional flapping hinge offset	0.04
Blade mass (kg)	50
Flapping hinge stiffness	515
Rotor speed (rad/s)	40
Rotor solidity	0.08
C_{d_0}	0.008
Equivalent friction area (m^2)	0.4
Maximum power (kW)	800

Table 7 Path length and computation time of trajectories A and B

Trajectories	Path length (m)	Global comp. time (s)
A	4310	9.43
B	8858	13.23

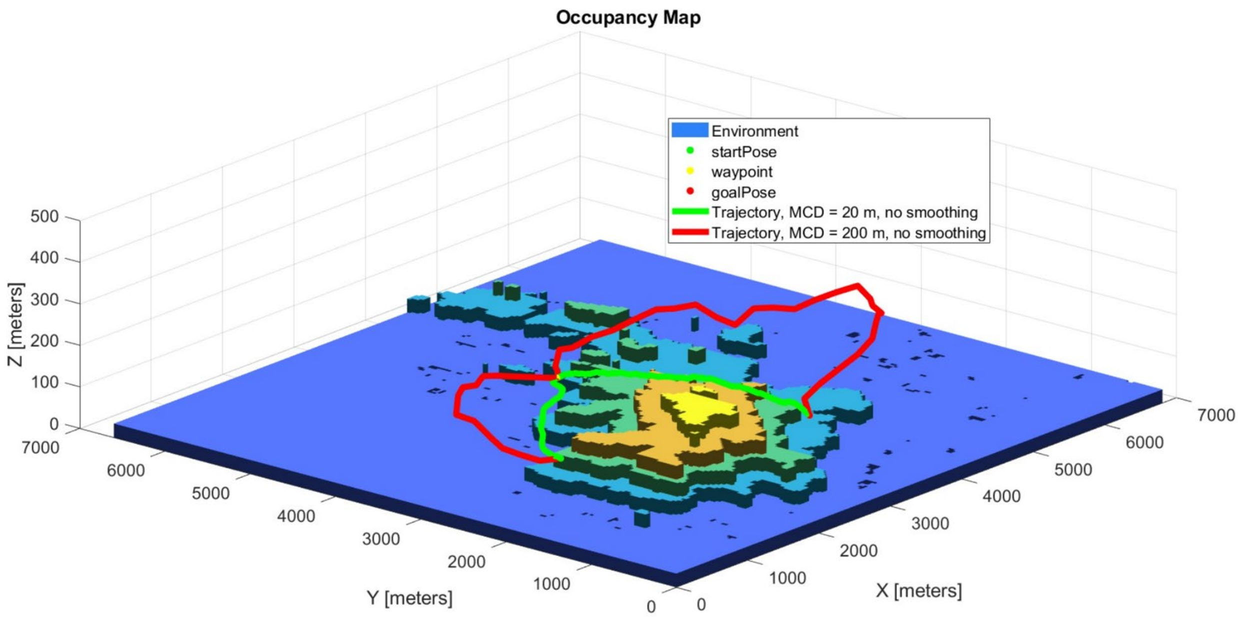

Fig. 15 Test on the MCD. The trajectory A, generated with a MCD of 20 m, is half the length of the trajectory B, generated with a MCD of 200 m

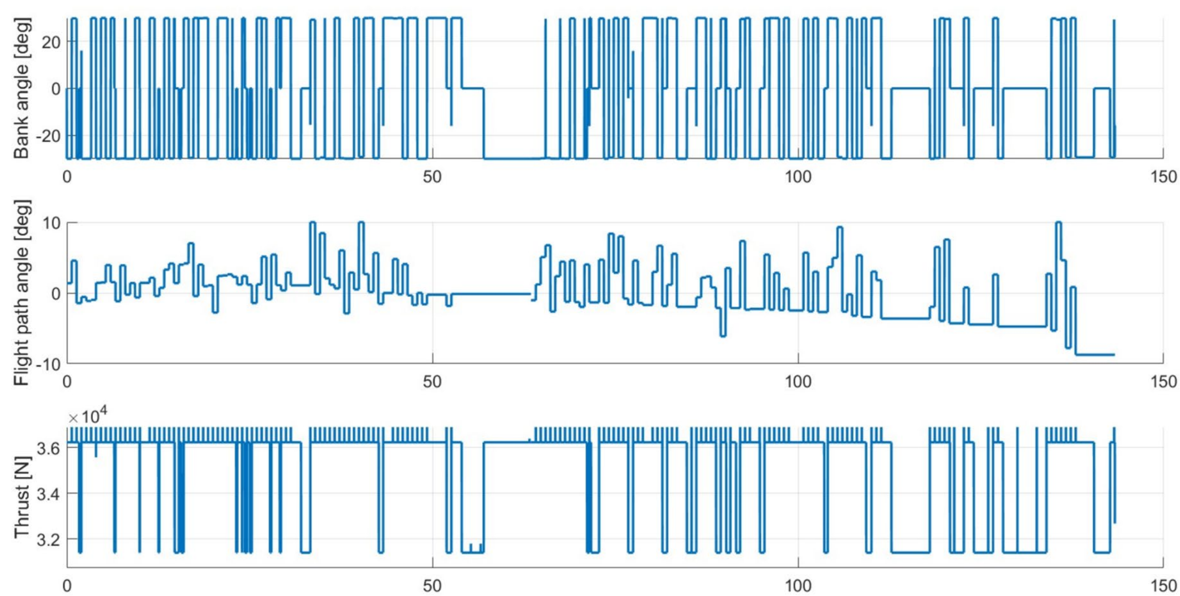

Fig. 16 Test on the MCD. Bank angle, flight path angle and thrust in the trajectory A, with MCD = 20 m and Dubins curves connection of nodes

3.5 Test results

3.5.1 Effect of the maximum connection distance

The comparison between trajectories A and B shows the impact of the maximum connection distance δ on the computed path. The lower MCD generates a shorter path (Fig. 15). The computation times and path length are shown in Table 7.

Since a smaller maximum connection distance leads to a shorter path, trajectory A, with a MCD of 20 m, is preferred to trajectory B. However, a lower MCD entails a trajectory populated with more waypoints, each at a maximum distance of 20 m from the adjacent one. This results in a series

Table 8 Path length before and after smoothing, and smoothing computational time

Trajectories	Path length (m)	Smoothing time (s)
A	4310	–
C	4284	0.075
D	3927	0.330

The larger window width entails a larger computational time, but also a shorter path

of short Dubins curves that make the path unfeasible for a helicopter. Indeed, the Dubins curves do not provide curvature continuity; this would lead to abrupt manoeuvres, as can be observed in the attitude and thrust plots of Fig. 16. To overcome this issue, the trajectory can be smoothed with the Savitzky–Golay filter.

3.5.2 Effect of path smoothing

The Savitzky–Golay filter described in Sect. 2.1.2 is used to smooth the trajectory. As explained in the dedicated Section, the tunable parameters of this filter are the window width and the polynomial order. By changing these parameters, one can obtain very different effects. In general, a large window width combined with a small polynomial order provides smoother paths, as can be observed in Fig. 19, where the trajectories A, C and D are compared. A is the original path; C is filtered with a window width of 27, and D with a window width of 151 (the number of states in the path is 219). A polynomial order of 3 is selected because a high polynomial order does not sufficiently smooth the curve, and a low polynomial order, like 2, creates excessive deviations from the original route.

Figures 17 and 18 illustrate the bank angle, flight path angle and thrust of trajectories C and D, respectively. As one can expect, the larger window width provided to trajectory D generates a smoother path also in terms of angles and forces (Fig. 19). A comparison with Fig. 16 reveals how filtering does not guarantee compliance with the prescribed performance constraints. Indeed, the bank and flight path angles in some points exceed the maxima of ±30° and ±10° since applying a filter bypasses the Dubins curves that enforce them. However, the transitions are significantly smoother and the angles and thrust curves are less oscillatory when filtering is applied.

3.5.3 Result of the test on obstacle avoidance

In this test, the reference route is planned before the start of the mission using the strategies shown in the previous Sections (RRT* + smoothing). At a certain point, the crew decides to deploy the drone, which flies ahead of the helicopter to detect possible unknown obstacles. If an obstacle is found along the planned route, the path is replanned from the current helicopter position to avoid the obstacle. The result is shown in Fig. 20, where the algorithm successfully replans a path that avoids the obstacle.

Figure 21 shows how in the truncation area, where the path is replanned (near $t = 80$ s) the manoeuvre is demanding, and the bank angle exceeds the maximum prescribed value. Also in Fig. 20, one can notice how the trajectory bends after the truncation. From the truncation point onward,

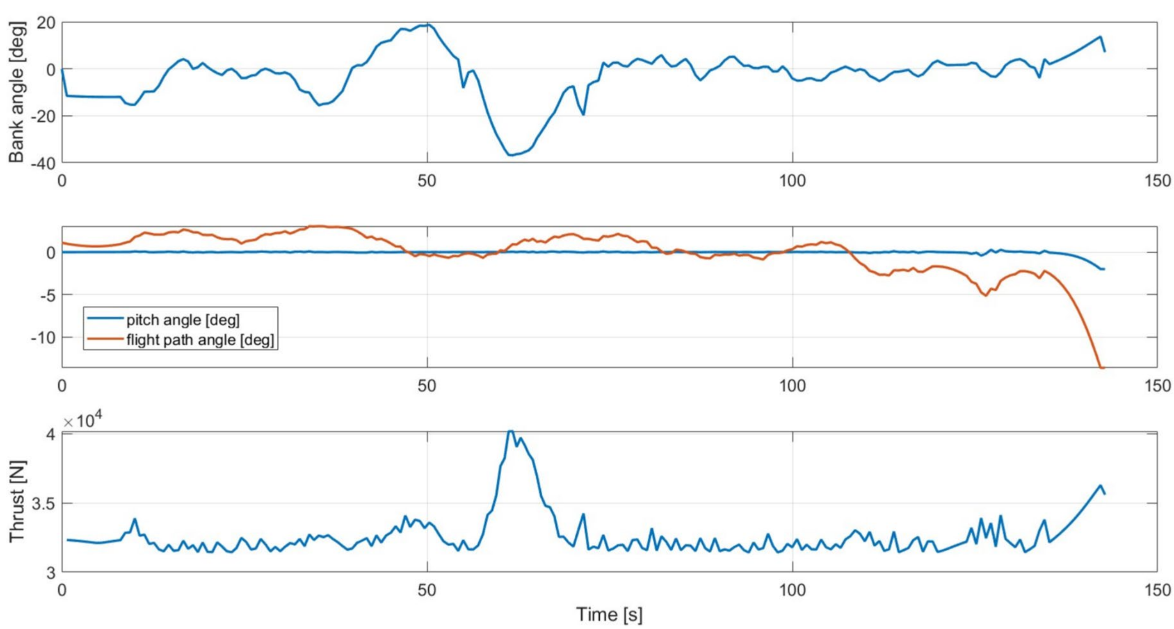

Fig. 17 Path smoothing test. Trajectory C has a window width of 27 and a polynomial order of 3

the trajectory was not smoothed, because smoothing does not improve sharp-cornered trajectories. At the truncation, the angles and the thrust reach large, unsustainable values. However, those large values of thrust and bank angle are computed considering a fixed airspeed of 30 m/s, while a pilot would slow down or remain in hover during the replanning phase, waiting for the new trajectory.

Replanning can also be performed with the RRT or RRT* algorithms. The results of replanning with RRT* are shown in Figs. 22 and 23. Also in this case, an abrupt manoeuvre is required at the truncation point to fly from the initial to the replanned trajectory at 30 m/s. However, as explained earlier, the airspeed at the truncation point would be nearly equal to zero in a real mission; therefore, the bank angle and the thrust during the trajectory change would be much smaller and more tolerable. Since the RRT* is an optimal algorithm, the replanned trajectory would be as short as possible and could be drawn at the very border of the obstacle, as one can see in Fig. 22. As a consequence, it is conservative to inflate the obstacle beyond its detected size and avoid unrefined smoothing. For this reason, the window width for the replanning was kept small compared with the number of states on the path (window width of 11 vs 56 path states) (Table 8).

The local computation times for replanning with BiRRT and RRT* are compared in Table 9.

Table 10 reports the maximum collective input and rotor power computed for the two replanned trajectories, including and excluding the effects in the truncation area. As explained earlier, the truncation area is the point along the trajectory where the helicopter stops flying the reference route to start flying the replanned one. The transition from one trajectory to another should be very smooth if the flight is conducted at constant speed. However, it can be more abrupt (e.g., a sharp heading change) if the helicopter holds in hover to wait for the information from the drone to be collected and the path to be replanned accordingly.

Table 10 shows how, if the transition from the reference route to the replanned trajectory is performed at 30 m/s, the maximum collective input and rotor power reach large values. In particular, according to Table 6, the maximum rotor power is exceeded by ten times, making the trajectories replanned with both BiRRT and RRT* unfeasible. If the transition is performed at a very low speed, or the helicopter holds in hover at the truncation point, the trajectory replanned with BiRRT is still unfeasible, but the one replanned with RRT* is feasible. Moreover, Table 9 suggests that the computational time required by RRT* when used as a local planner does not significantly differ from that of BiRRT.

4 Conclusion and future work

A path-planning strategy for HEMS missions featuring innovative helicopter-scout drone cooperation has been proposed. Two improved versions of the well-known Rapidly-exploring Random Tree, the RRT* (a probabilistically optimal extension of RRT) and BiRRT (a bidirectional formulation of RRT), have been investigated for the roles of global and local planners to plan the reference and replanned route, respectively. The planner finds a path between an initial and a goal point through intermediate assigned waypoints, assuring a safe distance from the terrain. When the scout drone detects a new obstacle, the

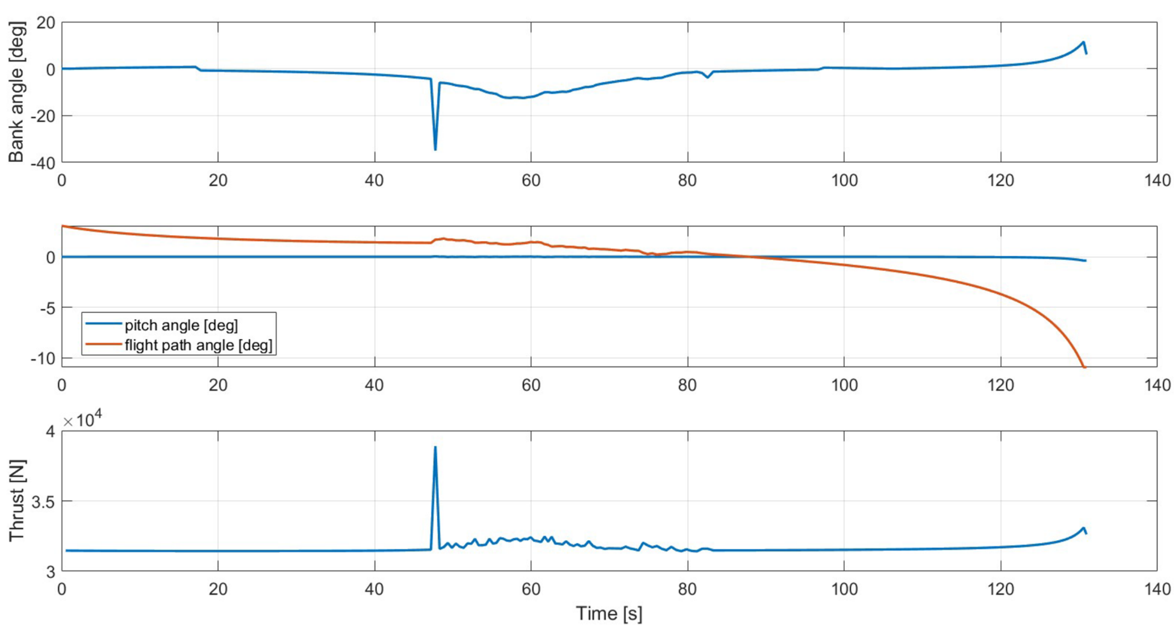

Fig. 18 Path smoothing test. Trajectory D has a window width of 151 and a polynomial order of 3

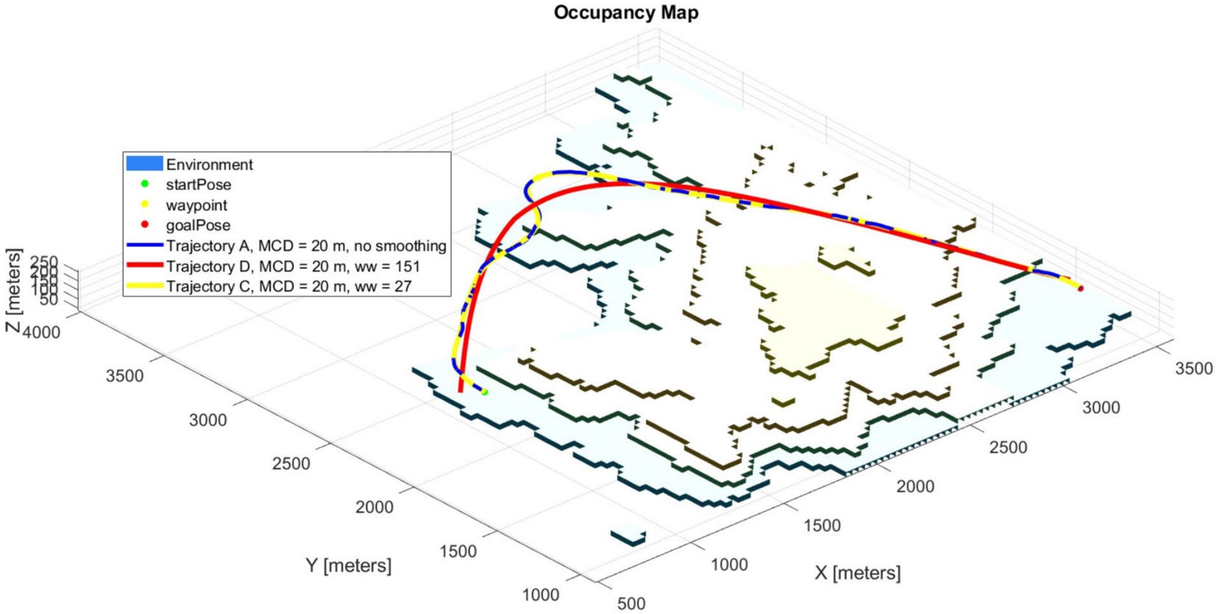

Fig. 19 Path smoothing test. Trajectory C is more adherent to the unfiltered path with respect to trajectory D. The larger the window width, the greater the smoothing effect

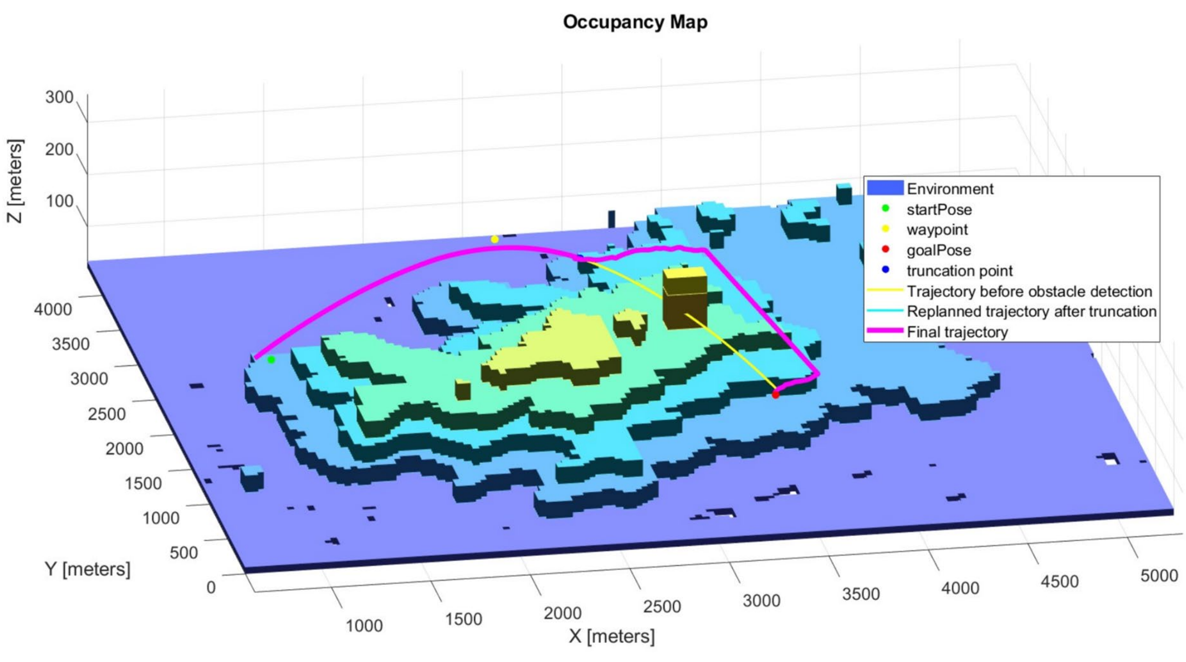

Fig. 20 Obstacle avoidance test. The initially planned path, in yellow, impacts on an obstacle detected by the drone (the parallelepipedal solid). The initial path is cut 40 states before the obstacle and replanned from the truncation state with the BiRRT algorithm

planner quickly computes a new safe path. However, the BiRRT, as the local planner, appears unable to guarantee a feasible trajectory in terms of attitude and rotor power, at least in the tested conditions corresponding to fixed airspeed and a maximum connection distance of 20 m. It is not excluded that the BiRRT could perform well if these parameters are tuned differently. On the contrary, the RRT * provides a feasible replanned trajectory in a time comparable to that of the BiRRT; therefore, it can be considered a good local planner. Smoothing the trajectory is proved to allow the decrease of the path length and the effective practicability of the route by a human-operated vehicle,

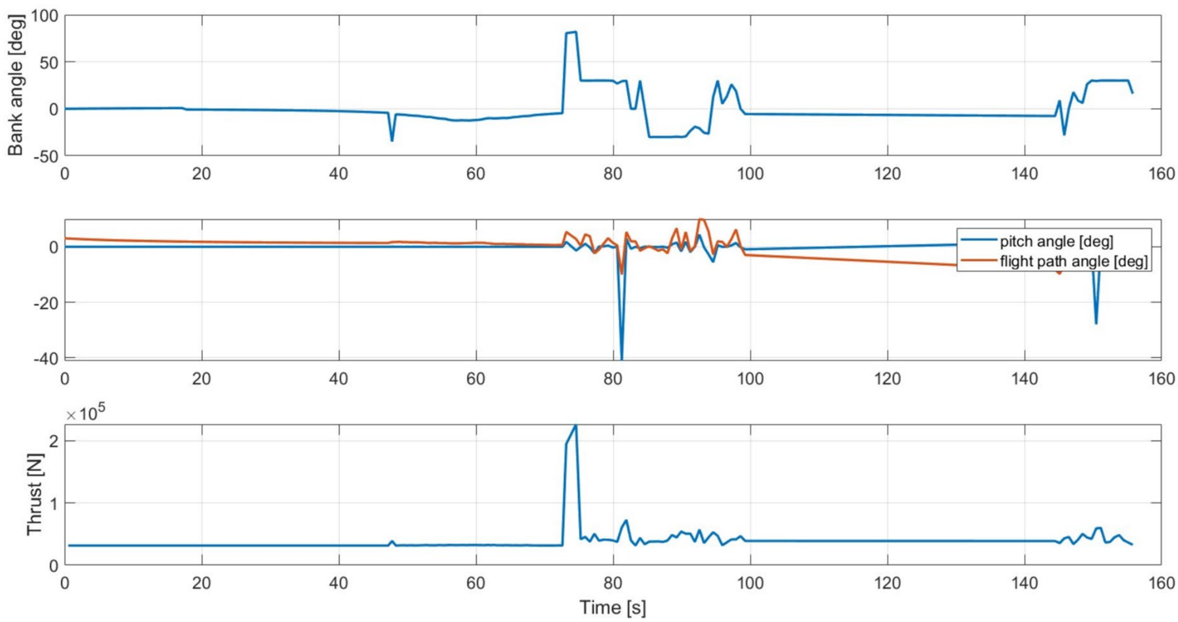

Fig. 21 Obstacle avoidance test. The path is truncated and replanned with the BiRRT algorithm. Right after the truncation, if the airspeed is fixed at 30 m/s the bank angle reaches a very large value (≃ 80°) increasing the thrust by 10 times

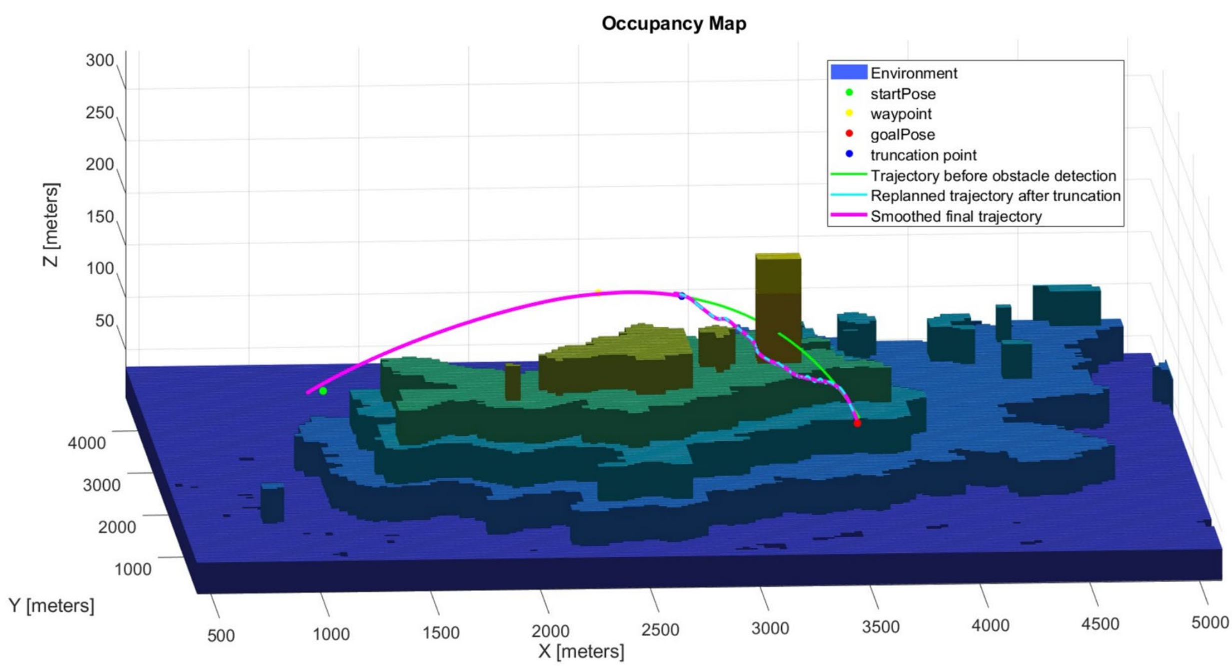

Fig. 22 Obstacle avoidance test. The initially planned path, in green, impacts on an obstacle detected by the drone (the parallelepipedal solid). The initial path is cut 40 states before the obstacle and replanned from the truncation state with the RRT* algorithm and filtered with a window width of 11 states

aspects that will be further explored in future work involving tests performed by expert pilots at a flight simulator to assess the feasibility of the trajectory in terms of applied forces and load factors.

Appendix A: The longitudinal trim problem

The equations of the longitudinal trim problem are reported in this Appendix. The problem is composed of the three equations (two components of force and one component of moment) that

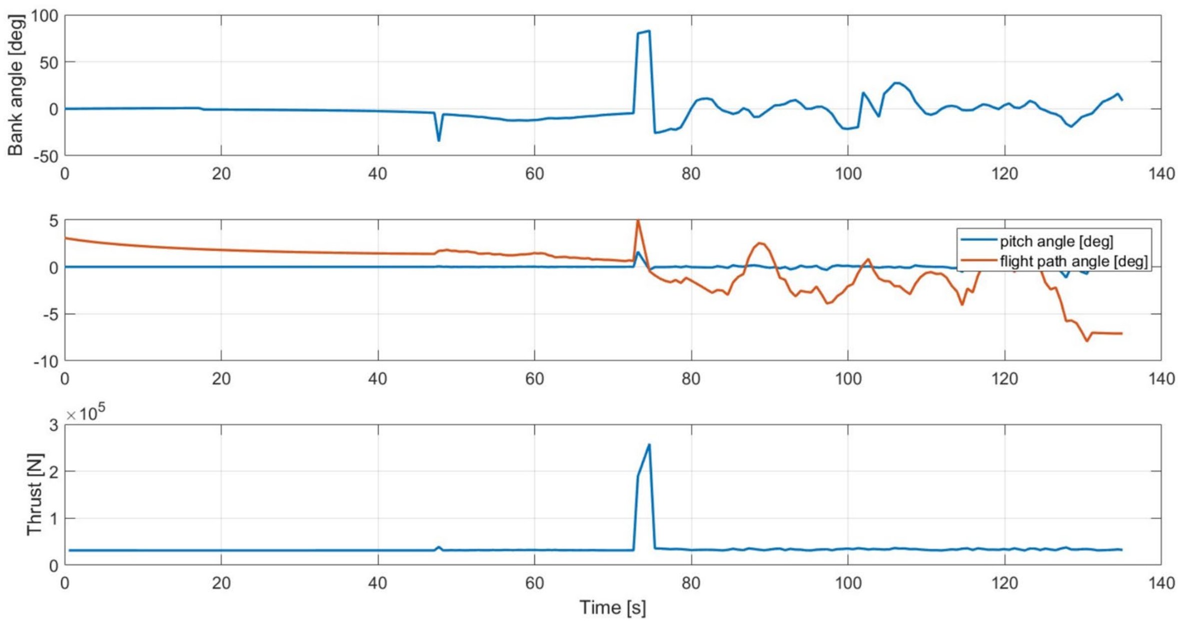

Fig. 23 Obstacle avoidance test. The path is truncated and replanned with the RRT* algorithm. Right after the truncation, if the airspeed is fixed at 30 m/s the bank angle reaches a very large value (≃ 80 deg) increasing the thrust by 10 times

Table 9 Comparison between the local computation times of BiRRT vs. RRT* replanning

Replanning algorithm	Local computation time (s)
BiRRT	3.96
RRT*	6.00

Table 10 Comparison between the maximum power and maximum collective input of BiRRT vs. RRT* replanning

Replanning algorithm	Max collective (including truncation) (°)	Max power (including truncation) (kW)	Max collective (excluding truncation) (°)	Max power (excluding truncation) (kW)
BiRRT	37	6600	13	950
RRT*	43	8000	7.5	360

describe the longitudinal equilibrium of the helicopter and 12 constitutive equations that describe the main rotor flapping and implicitly define the parameters used in the equilibrium equations. The problem is symbolically and numerically solved using Matlab.

A.1 Force and moment equilibrium

With reference to Fig. 24, the equilibrium equations are

$$T_D \cos(a_1 - B_1) - H_D \sin(a_1 - B_1) - W\cos\gamma - R_f \sin(\gamma+\tau) + P_c\cos(\gamma+\tau) = 0 \tag{11}$$

$$T_D \sin(a_1 - B_1) + H_D \cos(a_1 - B_1) - W\sin\gamma + R_f \cos(\gamma+\tau) + P_c\sin(\gamma+\tau) = 0 \tag{12}$$

$$W(h\sin\gamma - x_{CG}\cos\gamma) - R_f(h\cos(\gamma+\tau) + x_{CG}\sin(\gamma+\tau)) - P_c l\cos(\gamma+\tau) + M_f + K_H(a_1 - B_1) = 0 \tag{13}$$

where:

$$K_H = \frac{b}{2}(eS_\beta\Omega^2 + K_\beta). \tag{14}$$

S_β is the static flapping moment, K_β is the flapping stiffness and b is the number of blades; a_1 is the flap angle in the reference system of the swashplate, B_1 is the pitch angle in the reference frame of the shaft. Other quantities are defined in the next section.

A.2 Constitutive relations

Thrust in the disk frame:

$$T_D = \rho A v_{\text{tip}}^2 \sigma C_{L_\alpha} \frac{1}{2}\left(\left(\frac{1}{3} + \frac{\mu^2}{2}\right)\theta_0 - \frac{1}{2}\lambda - \frac{1}{2}\mu a_1\right). \tag{15}$$

Longitudinal force in the disk frame:

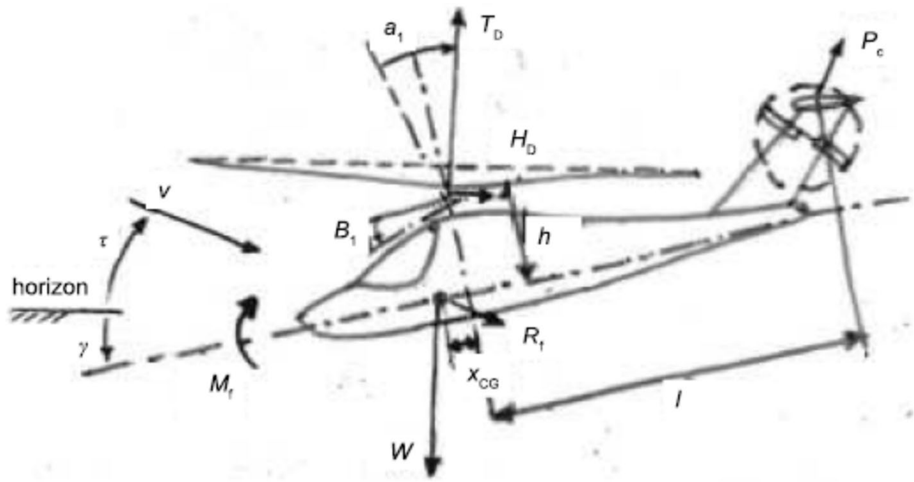

Fig. 24 Helicopter trim

$$H_D = \rho A v_{\text{tip}}^2 \sigma \left(\frac{1}{4}\mu C_D + C_{L_\alpha} \frac{1}{4} \left(\mu \lambda \theta_0 - \frac{1}{2}\lambda a_1 \right) \right). \tag{16}$$

Airframe drag:

$$R_f = \frac{1}{2}\rho V_\infty^2 f. \tag{17}$$

Airframe moment:

$$M_f = \frac{1}{2}\rho V_\infty^2 S l C_{m_f}. \tag{18}$$

Tailplane lift:

$$P_c = \frac{1}{2}\rho V_\infty^2 S_c C_{L_\alpha}(-(\gamma + \tau)). \tag{19}$$

Longitudinal flap angle (from rotor lateral moment equilibrium):

$$a_1 = \frac{2\mu\left(\frac{4}{3}\theta_0 - \lambda\right)}{1 - \frac{1}{2}\mu^2}. \tag{20}$$

Advance ratio:

$$\mu = \frac{V_\infty \cos \alpha_D}{v_{\text{tip}}}. \tag{21}$$

Inflow ratio:

$$\lambda = \frac{V_\infty sin\alpha_D + u}{v_{\text{tip}}}. \tag{22}$$

Induced velocity:

$$u = \frac{T_D}{2\rho A |v_1|}. \tag{23}$$

Inflow velocity:

$$v_1 = v_{\text{tip}} \sqrt{\mu^2 + \lambda^2}. \tag{24}$$

Inflow angle in the disk frame:

$$\alpha_D = \tau + \lambda - (a_1 - B_1). \tag{25}$$

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement. The project HEMS+ Scout Drone is supported by the POR-FESR 2014–2020, European Fund for Regional Development for Regione Sardegna, and by Sardegna Ricerche through the Project Number (CUP) I64D20000000006.

Availability of data and material Data used in this work may be requested from the HEMS+ Consortium.

Declarations

Conflict of interest The fourth author, Pierangelo Masarati, is an Associate Editor of the CEAS Aeronautical Journal but has not been involved in the review of this manuscript. The authors have no other conflict of interest nor Conflict of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Helicopter air ambulance operations. AC 135-14B, FAA (2015)
2. Raatiniemi, L., Liisanantti, J., Tommila, M., Moilanen, S., Ohtonen, P., Martikainen, M., Voipio, V., Reitala, J.I.T.:

Evaluating helicopter emergency medical missions: a reliability study of the HEMS benefit and NACA scores. Acta Anaesthesiol. Scand. **61**(5), 557–565 (2017)
3. USHST. Review of 2018 U.S. fatal accident data. https://ushst.org/reports/ (2019)
4. Avi, A., Frisco, N., Giurato, M., Lovera, M., Masarati, P., Panza, S., Parnisari, G., Roncolini, F., Sesana, M., Quaranta, G.: Scout drone: a drone-helicopter collaboration to support HEMS missions. In: Proceedings of the 48th European Rotorcraft Forum, Winterthur, Switzerland (2022)
5. Tang, H., Zhang, Y., Mohmoodian, V., Charkhgard, H.: Automated flight planning of high-density urban air mobility. Transp. Res. Part C Emerg. Technol. **131**, 103324 (2021)
6. Roncolini, F., Galante, G., Quaranta, G., Masarati, P.: Path planning for innovative solutions based on UAV-helicopter cooperation in HEMS missions. In: Proceedings of the 48th European Rotorcraft Forum, Winterthur, Switzerland (2022)
7. Huang, S., Teo, R.S.H.: Computationally efficient visibility graph-based generation of 3D shortest collision-free path among polyhedral obstacles for unmanned aerial vehicles. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1218–1223. IEEE (2019)
8. Blasi, L., D'Amato, E., Mattei, M., Notaro, I.: Path planning and real-time collision avoidance based on the essential visibility graph. Appl. Sci. **10**(16), 5613 (2020)
9. Majeed, A., Lee, S.: A fast global flight path planning algorithm based on space circumscription and sparse visibility graph for unmanned aerial vehicle. Electronics **7**(12), 375 (2018)
10. Ahmad, Z., Ullah, F., Tran, C., Lee, S.: Efficient energy flight path planning algorithm using 3-D visibility roadmap for small unmanned aerial vehicle. Int. J. Aerosp. Eng. **2017**, 2849745 (2017)
11. Magid, E., Lavrenov, R., Afanasyev, I.: Voronoi-based trajectory optimization for UGV path planning. In: 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), pp. 383–387. IEEE (2017)
12. Yan, F., Liu, Y.-S., Xiao, J.-Z.: Path planning in complex 3D environments using a probabilistic roadmap method. Int. J. Autom. Comput. **10**(6), 525–533 (2013)
13. Li, L., Zhan, H., Hao, Y.: The online path planning method of UAV autonomous inspection in distribution network. In: E3S Web of Conferences, vol. 256, pp. 01047. EDP Sciences (2021)
14. Lin, Y., Saripalli, S.: Path planning using 3D Dubins curve for unmanned aerial vehicles. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 296–304. IEEE (2014)
15. Adiyatov, O., Sultanov, K., Zhumabek, O., Varol, H.A.: Sparse tree heuristics for RRT* family motion planners. In: 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1447–1452. IEEE (2017)
16. Samaniego, F., Sanchis, J., García-Nieto, S., Simarro, R.: Recursive rewarding modified adaptive cell decomposition (RR-MACD): a dynamic path planning algorithm for UAVs. Electronics **8**(3), 306 (2019)
17. Palossi, D., Furci, M., Naldi, R., Marongiu, A., Marconi, L., Benini, L.: An energy-efficient parallel algorithm for real-time near-optimal UAV path planning. In: Proceedings of the ACM International Conference on Computing Frontiers, pp. 392–397 (2016)
18. Zhang, Z., Wu, J., Dai, J., He, C.: A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment. IEEE Access **8**, 122757–122771 (2020)
19. Jan, S.S., Hsiang, L.Y.: Integrated flight path planning system and flight control system for unmanned helicopters. Sensors **11**(8), 7502–7529 (2011)
20. Stentz, A.: Optimal and efficient path planning for unknown and dynamic environments. Technical report, Carnegie-Mellon Univ Pittsburgh PA Robotics Inst (1993)
21. Nash, A., Koenig, S., Tovey, C.: Lazy Theta*: any-angle path planning and path length analysis in 3D. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 24, pp. 147–154 (2010)
22. Zhang, Z., Wang, J., Li, J., Wang, X.: UAV path planning based on receding horizon control with adaptive strategy. In: 2017 29th Chinese Control And Decision Conference (CCDC), pp. 843–847. IEEE (2017)
23. Hartjes, S., Visser, H.G., Pavel, M.D.: Optimization of simultaneous non-interfering rotorcraft approach trajectories. In: The conference proceedings of the 35th European Rotorcraft Forum (2009)
24. Zhou, H., Xiong, H.-L., Liu, Y., Tan, N.-D., Chen, L.: Trajectory planning algorithm of UAV based on system positioning accuracy constraints. Electronics **9**(2), 250 (2020)
25. Khan, M.T., Raza, M.S., Malik, R., Yang, S., Junho, K.D.: Aspects of unmanned aerial vehicles path planning: overview and applications. Int. J. Commun. Syst. **34**(10), e4827 (2021)
26. Goel, U., Varshney, S., Jain, A., Maheshwari, S., Shukla, A.: Three dimensional path planning for UAVs in dynamic environment using glow-worm swarm optimization. Procedia Comput. Sci. **133**, 230–239 (2018)
27. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
28. He, Y., Zeng, Q., Liu, J., Xu, G., Deng, X.: Path planning for indoor UAV based on ant colony optimization. In: 2013 25th Chinese Control and Decision Conference (CCDC), pp. 2919–2923. IEEE (2013)
29. Lin, N., Tang, J., Li, X., Zhao, L.: A novel improved bat algorithm in UAV path planning. J. Comput. Mater. Contin. **61**, 323–344 (2019)
30. Hasanzade, M., Koyuncu, E.: A dynamically feasible fast replanning strategy with deep reinforcement learning. J. Intell. Robot. Syst. **101**(1), 1–17 (2021)
31. Raheem, F.A., Hameed, U.I.: Path planning algorithm using D* heuristic method based on PSO in dynamic environment. Am. Acad. Sci. Res. J. Eng. Technol. Sci. **49**(1), 257–271 (2018)
32. Ma, N., Cao, Y., Wang, X., Wang, Z., Sun, H.: A fast path re-planning method for UAV based on improved A* algorithm. In: 2020 3rd International Conference on Unmanned Systems (ICUS), pp. 462–467. IEEE (2020)
33. Chen, Y., Li, W., Qi, R.: Research and simulation of UAV three-dimensional path replanning in complex environment. In: 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), pp. 746–751. IEEE (2021)
34. Khuswendi, T., Hindersah, H., Adiprawita, W.: UAV path planning using potential field and modified receding horizon A* 3D algorithm. In: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp. 1–6. IEEE (2011)
35. Poudel, S., Moh, S.: Hybrid path planning for efficient data collection in UAV-aided WSNs for emergency applications. Sensors **21**(8), 2839 (2021)
36. Chen, J., Li, M., Yuan, Z., Gu, Q.: An improved A algorithm for UAV path planning problems. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, pp. 958–962. IEEE (2020)
37. LaValle, S.M., et al.: Rapidly-exploring random trees: a new tool for path planning. Research Report 9811, Department of Computer Science, Iowa State University (1998)
38. Gallagher, N.B.: Savitzky-golay Smoothing and Differentiation Filter. Eigenvector Research Incorporated, Washington (2020)
39. Ravankar, A., Ravankar, A.A., Kobayashi, Y., Hoshino, Y., Peng, C.-C.: Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges. Sensors **18**(9), 3170 (2018)

40. Padfield, G.D.: Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling. Blackwell Publishing, New York (2007)
41. Leishman, J.G.: Principles of Helicopter Aerodynamics, 2nd edn. Cambridge University Press, Cambridge (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.