PATH PLANNING FOR INNOVATIVE SOLUTIONS BASED ON UAV-HELICOPTER COOPERATION IN HEMS MISSIONS

Francesca Roncolini*, Giovanni Galante, Giuseppe Quaranta, Pierangelo Masarati
Politecnico di Milano, Dipartimento di Scienze e Tecnologie Aerospaziali, Milano - Italy
* Corresponding author, email:francesca.roncolini@polimi.it

Abstract

The paper investigates the path-planning problem applied to an innovative UAV–helicopter cooperation system that aims at increasing safety during HEMS operations. The drone, that could be optionally launched by the helicopter, will have the mission to explore the area of operation to detect meteorological and physical obstacles. The combination of Rapidly-exploring Random Tree* as global planner and of Bidirectional Rapidly-exploring Random Tree as local planner is proved to provide a nearly-optimal global path and a rapid re-planning in case of new obstacles detection. The adoption of Savitzky–Golay filter enables trajectory smoothing, improving its practicability. The feasibility of the identified trajectory for a three-dimensional helicopter is assessed through computation of attitude and forces, the latter carried out by means of a multibody analysis software.

1. INTRODUCTION

In modern society, Helicopter Emergency Medical Service (HEMS) mission are part of the trauma management systems and health care [1]. The usage in sparsely populated and rural areas may be essential to allow a fast transport of patients that 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 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, and sometimes, the sudden deterioration of weather is not uncommon and it may lead to flight into Unintended Instrumental Meteorological Conditions (UIMC). In 2018, the Federal Aviation Administration (FAA) reported that UIMC and Low Altitude Operations represented two of the three main causes of helicopter accidents [2].
However, to achieve the level of reliability that is expected for HEMS and SAR operations, there is a need for the rotorcraft involved to fly Anywhere, Anytime, in All-weather conditions (AAA). To get closer to reaching this goal, Politecnico di Milano has started a collaboration with industry to develop and test innovative solutions based on cooperation of the helicopter with a drone. 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 increase of the mission risks. More details on this project can be found in [3].

Within this project it is planned to have a control station that is used to plan the route to be followed by the helicopter involved in the HEMS operation. Such route can be flown by the helicopter if VMC conditions are guaranteed. However, in case the possibility to fly into UIMC condition is foreseen, the crew may deploy the drone through the hoist and put it into service. The drone can follow the planned route sensing GNSS signal level, turbulence and cloud ceiling, and detecting physical obstacles. The data collected by the drone can then be sent to helicopter control station to update the maps, allowing the crew to increase the situational awareness during the mission. Using an automatic path planner, with a predefined frequency, the reference route validity is checked. In light of the possible newly detected weather conditions or obstacles, the route could be re-planned, possibly until the drone has explored the entire mission space and a successful and safe path has been found. This strategy is shown in Figure 1:

Presented at the 48th European Rotorcraft Forum, Winterthur, Switzerland, 6-8 September, 2022

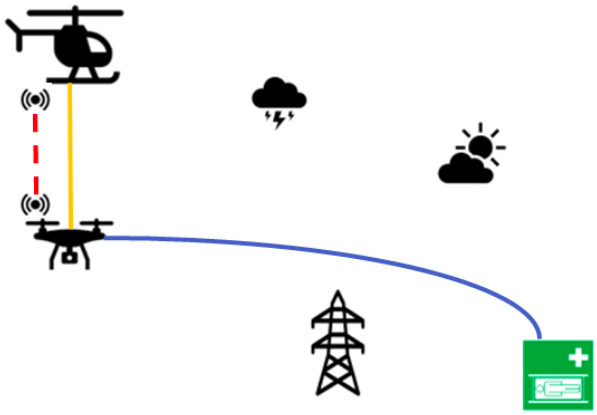

Figure 1: Scheme of HEMS mission. A scout drone is released from the helicopter by hoist and flies the reference route, detecting obstacles.

Path planning for vehicle that fly in low altitude environments is an active research field. For instance Advance Air Mobility systems that are envisioned to fly, autonomously, in the urban environment, have to develop path planning strategies that can avoid obstacles and obstructions and also manage conflicts in high-density operational environment [4].

The objective of this work is to better describe and motivate the strategy identified to perform the Path Planning, and to illustrate a methodology to assess quality and feasibility of the computed trajectory. Section 2 contains a review of Path Planning algorithms and describes criteria of algorithm selection and methods to assess trajectory feasibility. Section 3 features a description of the adopted testing procedure and discusses tests results.

2. METHODS AND ALGORITHMS

The most advanced and reliable path planning algorithms are categorized and compared in Section 2.1, in which a family of planners is selected.
Path planning algorithms provide a sequence of waypoints, which are connected by means of Dubins curves to obtain a smooth trajectory. In alternative, the trajectory can be smoothed via filtering, as explained in Section 2.2.
Once trajectory is defined in terms of time and positions, simulations are performed to test whether it is feasible or not. This evaluation is based on two indicators: helicopter attitude and rotor thrust required to follow the planned path. Section 2.3 addresses the definition of a methodology to carry out this analysis.

2.1. Review of algorithms and planner selection

Before choosing a path planning algorithm, it is fundamental to analyse the problem and bring to the table all possible options. Devoting effort to a wide algorithms review raises the chances to choose an algorithm that successfully meets mission requirements.
First of all, it is useful to define some basic terminology:
Global path planning: in global path planning, path searching is carried out in a known environment. The reference path is computed **offline**, with focus on safety and length optimality.
Local path planning: in local path planning, path searching is carried out in a completely or partially unknown environment. The reference path is computed **real-time**, updating environment information with sensors data and re-planning the path in case of collision detection. The focus is on path generation time.
Real-time reactivity: a local planner is real-time reactive when it has very fast collision avoidance capability, with reaction time $< 200\ ms$.

It is now advantageous to formulate the problem according to path planning terminology: after receiving an emergency call, the helicopter computer runs a Global planner to generate a reference path basing on known environment maps. At the end of this operation, the scout drone explores the planned route looking for undetected physical or meteorological obstacles in the area, and sends the information to the helicopter computer, which continuously updates the Occupancy Map (digital map containing environment data) and checks trajectory validity. If the trajectory is discovered to intersect with one obstacle, a new path is planned with a Local planner. Real-time reactivity is not strictly essential in this application, seeing as for safety reasons the scout drone has a sufficient head start with respect to the helicopter, nevertheless a shorter computation time is preferred, given the urgency of HEMS missions.
The computed path should be constrained to the helicopter performance: a maximum range of attitude should be prescribed in order to enforce passengers comfort.

2.1.1. Algorithms review

An overview of the existing path planning algorithms is presented in the next paragraphs.

Sampling-based These algorithms 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 by means of 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, simply structured and easy to implement, suitable both for global and local planning.
Some algorithms belonging to this family are: Visibility Graph (VG), Voronoi Diagrams, Probabilistic RoadMap (PRM), Rapidly-exploring Random Tree (RRT).

Graph-based Graph-based algorithms search the least-cost path through the available grid points in a graph previously built from given start to goal nodes. Well mature algorithms, easy to implement and often combined with other methods to achieve global optimal solutions, their application can be both real-time and offline.
Some algorithms of this class are: Dijkstra, A*, D*, θ^*.

Numerical optimization These methods mathematically model the environment as well as the body, considering kinematic, dynamic, environmental, 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 implemented especially in global planning, when focus is on optimality. Examples of this class of algorithms are Mixed-Integer Linear Programming (MILP) and Non-Linear Programming (NLP).

Bio-inspired These algorithms optimize path by mimicking biological behaviour. Up to date these methods, which are still research object, are complex and their long iteration time make them suitable only for global planning. Some examples are: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Bat Algorithm (BA), Artificial Neural Network (ANN) and Deep Reinforcement Learning (DRL).

The above-mentioned families of algorithms have been better investigated by analyzing recent studies reported in scientific papers. Table 1 contains a classification of algorithms according to the information acquired from this investigation. The sorted characteristics are: paper reference, year of publication of the work, offline or real-time application, re-planning options, reactivity, inclusion of performance constraints in path planning, post-processing for trajectory smoothing, validation tests performed by simulation or real experiments, static and/or dynamic obstacles management.

2.1.2. Algorithm selection: RRT

Given the safety requirements of HEMS tasks, a mature and consolidated algorithm, with proved applications in real contexts, should be selected. Moreover, the scout drone-helicopter problem requests both offline and real-time capability, re-planning option, and inclusion of performance constraints.
In light of this and of the information presented in Table 1, the sampling-based Rapidly-exploring Random Tree algorithm has been chosen, along with two improved versions: RRT* and BiRRT (bidirectional RRT).

RRT This algorithm files the environment as an occupancy grid map, where information of the type occupied/free is stored in every grid point.
Tree nodes are identified by states: a state is defined by its 3D position and heading $q = [x, y, z, \psi]$. The starting node is the tree root q_{init}.
A random state q_{rand} in the state-space is selected during *sampling* phase, and the nearest node q_{near} in the existing tree 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 away from q_{near} is specified; if q_{rand} is closer than δ, then $q_{new} = q_{rand}$. If an obstacle is present between q_{near} and q_{new}, the latter is simply not added to the tree. Node expansion is shown in Figure 2.
This growing process continues until path gets to within a threshold of the goal.

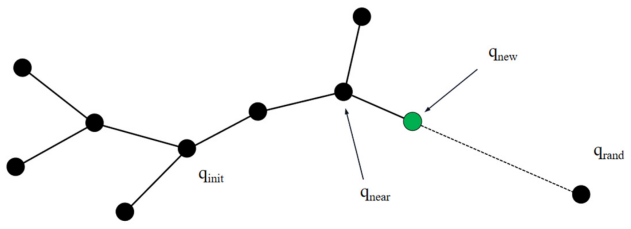

Figure 2: RRT approach

Table 1: Path planning algorithms classification

Family	Approach	Paper	Year	Offline	Real time	Re-plan	Reactive	Perf. constraints	Post-proc.	Sim/Exp	St/Dyn Obstacles
Sampling based	Visibility graph	[5]	2019	x	x	x		-	-	S	S
		[6]	2020		x	x		x	x	S	S/D
		[7]	2018	x				x	x	S	S
		[8]	2017	x				x		S	S
	Voronoi	[9]	2017	x	x	x		x		S	S/D
	PRM	[10]	2013	x				x	x	S/E	S
	RRT	[11]	2021	x	x	x		x	x	S	S/D
		[12]	2014	x	x	x		x	x	S/E	S/D
		[13]	2017	x						S	S
		[14]	2019	x	x	x		x		S	S/D
Graph based	Dijkstra	[15]	2016		x	x				S	S/D
	A^*	[16]	2020		x	x		x		S	S/D
		[17]	2011	x		x				S	S
	D^*	[18]	1993								
	θ^*	[19]	2010								
Numerical optimization	MILP	[20]	2017		x			x		S	S/D
	NLP	[21]	2009	x				x		S	S
Bio-inspired	GA	[22]	2020	x				x		S	S
	ANN	[23]	2014	x				x		S	S
	PSO	[24]	2018		x					S	S/D
	ABC	[25]	2007								
	ACO	[26]	2013	x						S	S
	BA	[27]	2019	x				x		S	
	DRL	[28]	2020		x	x	x	x	x	S/E	S/D
Fusion	PSO + D^*	[29]	2018		x	x				S	S/D
	A^* + GA	[30]	2020	x	x	x	x	x		S	S
	MPC + PSO + RRT	[31]	2021		x	x		x		S	S/D
	PF + A^*	[32]	2011	x						S	S
	PRM + ABC	[33]	2021		x	x		x		S	S/D
	MILP + A^*	[34]	2020	x				x		S	S

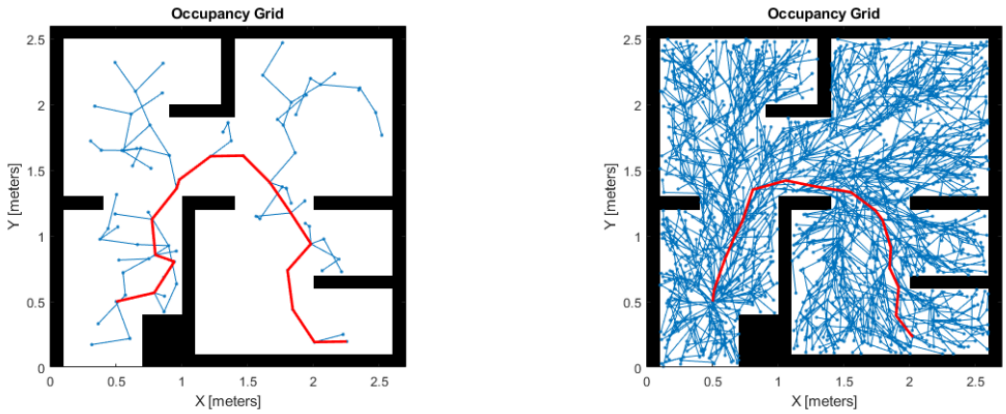

Figure 3: Comparison between RRT and RRT*

RRT* It is the probabilistically optimal extension of RRT: as number of nodes in the tree increases to infinity, probability of finding optimal path converges to 1. The cost is an increased path generation time. The difference between RRT and RRT*, shown in Figure 3, lies in the nearest node selection (it is not necessarily the nearest node to be connected with q_{new}, other nodes in a given search radius are checked) and in the stop criteria (process does not stop when goal is reached, but continues refining the path until max iterations are achieved).

BiRRT Bidirectional RRT 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 they connect. It can be very fast, however sacrificing asymptotical optimality of RRT*.

2.2. Implementation

Implementation of simulation environment and of path planning problem has been carried out in MATLAB 2021b, with the inclusion of Matlab Navigation Toolbox, which features tools for Occupancy Maps generation and several built-in or customizable algorithms for motion planning.

Map It has been built in the shape of an Occupancy Map, which consists in n vectors containing $[x\ y\ z]$ positions of Digital Terrain Elevation Data (DTED) points obtained from USGS EROS Archive - Shuttle Radar Topography Mission with resolution of 1 arc-second [35]. The map used for simulations is shown in Figure 4 and described in Section 3.2.

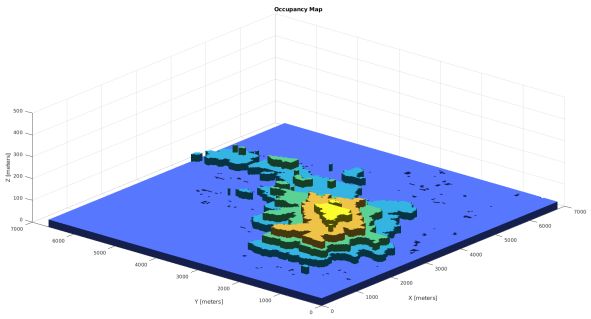

Figure 4: Occupancy Map of La Maddalena

The Occupancy Map can be inflated in order to guarantee a safety distance from obstacles.

Planner The developed code provides the possibility to choose among RRT, RRT* and BiRRT both for global and local planning. The *planner* object has two input: State Space and State Validator. State Space, better described in next paragraph, represents all possible positions the helicopter can occupy according to performance and other constraints; State Validator is an object containing information about whether a state is occupied or not. Planner properties (such as Maximum connection distance or Maximum iterations) are set in a dedicated section.
The function *plan* is adopted to plan a path between two states with the selected planner, as shown in the following instructions:

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

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

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

State Space A State Space is defined as all feasible $[x, y, z, \psi]$ of a vehicle during path planning (where the first three coordinates are position and ψ is heading) and, in Matlab, is represented by a State Space object constructed by *nav.StateSpace* class. To impose the respect of performance constraints, a customized State Space bounded by such constraints has been created using the function *createPlanningTemplate*. The adopted performance constraints are: maximum roll angle, maximum flight path angle and airspeed. 3D Dubins curves have been used in node connection, being the shortest segments that connect two points with constraints on the turning radius and prescribed initial and final headings [12]. A segment of path planned with and without Dubins curves is illustrated in Figure 5.

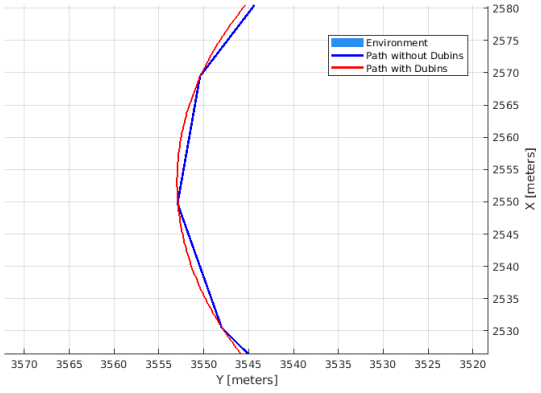

Figure 5: Node connection with Dubins curves

Re-planning/reconnecting After the scout drone detects an obstacle, the Occupancy Map is updated with the new information and validity of the global path is checked by means of the Matlab functions *isStateValid* and *isMotionValid*: if at least one state on the path is invalid, re-planning or reconnecting is necessary.

Re-plan strategy deletes the path states after the actual position and re-plans a path from it to the goal state.

Reconnect strategy truncates path states from actual position to a user-defined number of states after the obstacle. The local planner reconnects actual position to the truncated branch attached to goal state.

Generally, re-plan strategy gives a faster computational time, while reconnect strategy gives a shorter path length.

Smoothing Although Dubins curves enforce the respect of performance constraints, they are only C1-continuous, meaning they are continuous and differentiable, and their first derivative is continuous. This implies that the second derivative, and therefore curvature, is not necessarily continuous, resulting in an uncomfortable and possibly unfeasible trajectory.

For this reason, a further smoothing phase has been involved, the smoothing filter being of the Savitzky-Golay type, described in [36]. This filter is usually applied to smooth digital signals and has been chosen because of its simple implementation (Matlab features the designated function *sgolayfilt*). More specific path smoothing techniques are presented in [37].

For a given signal measured at N points and a filter of Window width w, Savitzky-Golay filter computes a polynomial fit of order o in each filter window as the filter is moved across the signal. The filter estimate at the center of each window is given by the polynomial fit at the center point, as shown by the yellow cross in the subplot in the top right of Figure 6. The lower the Polynomial order and higher the Window width, the smoother the path, at the price of precision loss.

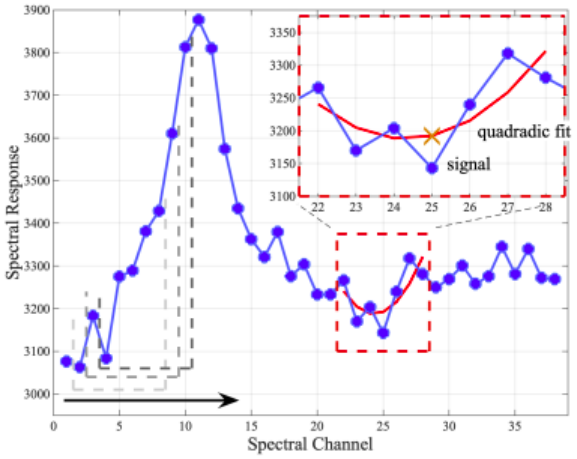

Figure 6: Savitzky-Golay filter with Window width = 7 and Polynomial order = 2

2.3. Assessment of trajectory feasibility

2.3.1. Calculation of Euler angles

Euler angles pitch θ, roll ϕ and yaw ψ are computed for each path waypoint in order to fully characterize helicopter motion. These angles represent the three consecutive rotations ψ, θ, ϕ to turn North-East-Down (NED) reference frame into Body reference frame: they describe helicopter orientation.
The following assumptions have been made for Euler angles computation:

- velocity vector is contained in the helicopter symmetry plane;
- when trajectory is straight and uniform, θ and ϕ are null;
- during maneuvers producing inertia forces, the Tip Path Plane is supposed perpendicular to the yaw axis, and fixed to the helicopter.

Yaw angle Thanks to the first assumption it is possible to infer that the yaw angle ψ coincides with heading. Therefore, being x the coordinate towards East and y towards North, the first Euler angle is:

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

Pitch angle Pitch angle has been computed starting from curvature k_v of the trajectory projected on vertical plane. Indeed, 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 vertical plane can be approximated in every point to a curvilinear maneuver which radius is $\frac{1}{k_v}$.

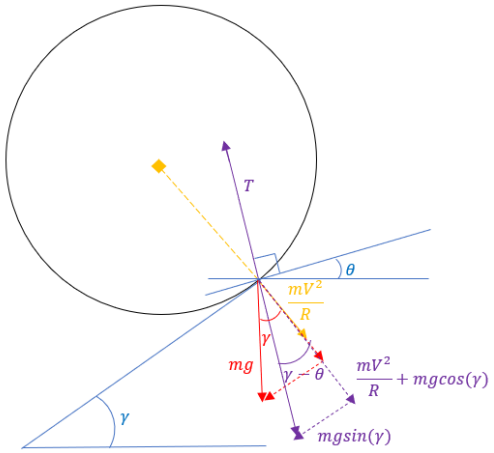

Figure 7: Pitch computation: equilibrium of forces during pull-up

As can be observed in Figure 7, a static equilibrium of forces including weight and centrifugal force is performed at each time step, and it is found that:

$$\gamma - \theta = \alpha = \arctan\left(\frac{mgsin\gamma}{\frac{mV^2}{R_v} + mgcos\gamma}\right) \tag{2}$$

where $\theta = \gamma - \alpha$. Signs of Eq. 2 depend on the direction of binormal vector and on the direction of vertical motion.
The formula for curvature of a curve parameterized by a generic parameter, in this case time, was used to compute k_v:

$$k(t) = \frac{|\underline{\alpha}'(t) \times \underline{\alpha}''(t)|}{|\underline{\alpha}'(t)|^3} \tag{3}$$

For a 3D trajectory, $\alpha(t) = [x(t), y(t), z(t)]$, where t is the vector of times. In this particular instance, a change of variables was operated in order to obtain the curvature of the vertically projected trajectory: k_v was computed using $\alpha_v(t) = [v(t), z(t), 0]$, where $v(t) = \sqrt{x^2 + y^2}$.

Roll angle The procedure to obtain bank angle Φ is very similar to the computation of pitch angle, applied to horizontal plane: the turn radius R_Φ is obtained from horizontal curvature k_h, and thanks to the equilibrium of forces bank angle is computed:

$$\Phi = atan\left(\frac{V^2}{gR_\Phi}\right) \tag{4}$$

Roll angle is computed in dependency of bank and pitch angles:

$$sin(\phi) = sin(\Phi)\, cos(\theta) \tag{5}$$

2.3.2. Thrust computation

The thrust needed to follow the planned route is another good indicator of trajectory feasibility: if the helicopter model is known, the needed thrust can be compared to its typical range of thrust.
Thrust computation problem has been solved by means of MBDyn, which is a free multibody dynamics analysis software [38] developed at DAER (Politecnico di Milano), capable of solving both initial value and inverse dynamics problems. The latter concerns the issue of computing forces given motion.
Motion was provided as a set of times, positions and orientation vectors, times being the ones corresponding to each path waypoint. Times have been calculated by hypothesizing a constant airspeed during the whole trajectory.

3. TESTS AND RESULTS

3.1. Testing procedure

Tests were performed only via simulation.
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 next one.
As it will happen in real flights involving the scout drone, in which the latter will send to helicopter information about meteorological or physical barriers, an obstacle modifying the Occupancy Map is introduced at a certain point of the simulation. The algorithm shall update the Occupancy Map and quickly re-plan a path by means of a local planner, choosing a strategy between re-planning and reconnecting. As explained above, re-planning strategy is computationally faster than reconnecting, therefore it has been selected as permanent choice.

3.2. Tests environment

The environment chosen for tests is La Maddalena (SS), a small island in Northern Sardinia; the related Occupancy Map is represented in Figure 4. 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 hospital by road, having the hospital no heliport;
- trajectory must pass through 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 computations.

3.3. Tests description

Among all the performed tests, three are discussed here, each one exploring different planner options as shown in Table 2:

Test	Global planner	Local planner	Max Connection Distance	Smoothing
1	RRT*	BiRRT	20 m	no
2	RRT*	BiRRT	200 m	no
3	RRT*	BiRRT	200 m	yes

Table 2: Tests options

Tests undergo the simplifying assumption that airspeed is constant along path. Performance constraints and filter options are displayed in Table 3.

Performance constraints		Filter options	
Max Roll Angle	$60°$	Window Width	9
Airspeed	$30\frac{m}{s}$	Polynomial Order	3
Flight Path Angle	$25°$		

Table 3: Performance constraints and filter options

3.4. Tests results

3.4.1. Effect of Maximum connection distance

The comparison between Test 1 and Test 2 shows the impact of planner Maximum connection distance δ on computed path: both strategies succeed in finding a new path after the insertion of a parallelepipedal obstacle, with the difference that the second trajectory is

slightly longer but requires considerably less computing time both for global and local planning, as reported in Table 4. A greater Maximum connection distance is overall more advantageous, sacrificing a slight amount of length optimality to obtain an important gain in reactivity.

Test	Path length	Global comp. time	Local comp. time
1	5630 m	30.12 s	3.06 s
2	6130 m	1.52 s	0.13 s

Table 4: Path length and computation time of tests 1 and 2

3.4.2. Effect of path smoothing

Test 2 vs. Test 3 point out the effect of trajectory smoothing: Figure 9 illustrates the trajectory before and after Savitzky-Golay filtering. Filtering improves smoothness and length-optimality, as shown in Table 5.

Test	Path length	Smoothing time
2	6130 m	-
3	5701 m	0.37 s

Table 5: Path length before and after smoothing, and computational time for smoothing

However, smoothing alters the trajectory final position, which does not reach goal point, but a nearby point. With a Window width of 9 and a Polynomial order of 3, the distance from goal point is 221 m. If Window width was reduced to 7, the distance would result 57 m but there would be a loss in trajectory smoothing and comfort. In future works, a criterion to select an acceptable goal threshold should be identified to tune Window width and Polynomial order accordingly.
In Figure 9 it is also possible to observe how re-planning works: the light blue path is global path, computed at the mission beginning, before the detection of an obstacle on the path. After the detection, the algorithm checks each path state validity by means of the State Validator object: since the global path is found to be invalid, it is truncated a user-defined number of waypoints before the first invalid state, and re-planned until goal state (green path).

3.4.3. Attitude and thrust analysis

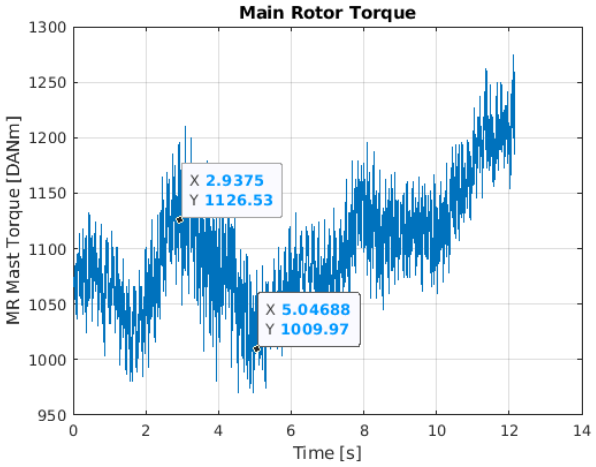

Figure 8: Main rotor mast torque data from recorded flight

Attitude and expected thrust during trajectory execution of Test 3 are represented in Figure 10. The maximum acceptable bank angle of 60° prescribed by performance constraints is reached in a point of the trajectory, at around $107s$, but attention should be brought to the fact that airspeed was supposed constant during the simulation. Plausibly, during a narrow turn airspeed would be temporarily dropped by the pilot reducing bank angle according to Eq. 4.
To analyze thrust, a numerical model inspired to a real helicopter (here not specified) was implemented in MBDyn. Figure 10 illustrates the output of MBDyn analyses: static thrust is $3200\ kg$, which is indeed the weight of the helicopter, while during high curvature maneuvers thrust reaches an increase of 12%.
To evaluate if the magnitude of these peaks is reasonable and realistic, mast torque data from a real recorded flight of the modelled helicopter have been analyzed. A piece of record is shown in Figure 8, involving a maneuver that increases the torque by 12%. Torque-thrust relation can be supposed locally linear, therefore an increase of thrust by 12% is deemed reasonable and not hazardous.

4. CONCLUSION AND FUTURE WORK

A Path Planning strategy for HEMS missions featuring the innovative helicopter-scout drone cooperation has been proposed. It is based on two improved versions of the well-known Rapidly-exploring Random Tree: RRT* and BiRRT. The planner is capable of finding a path between an initial point and a goal point

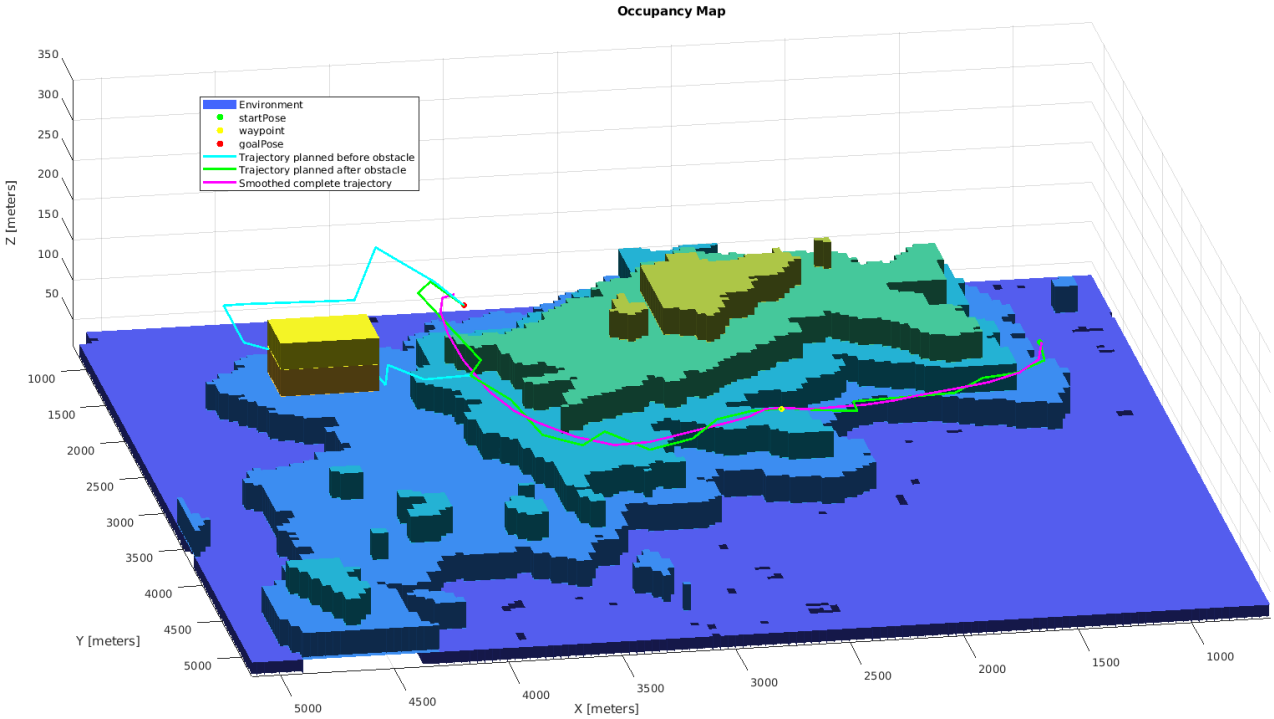

Figure 9: Test 2 vs. Test 3

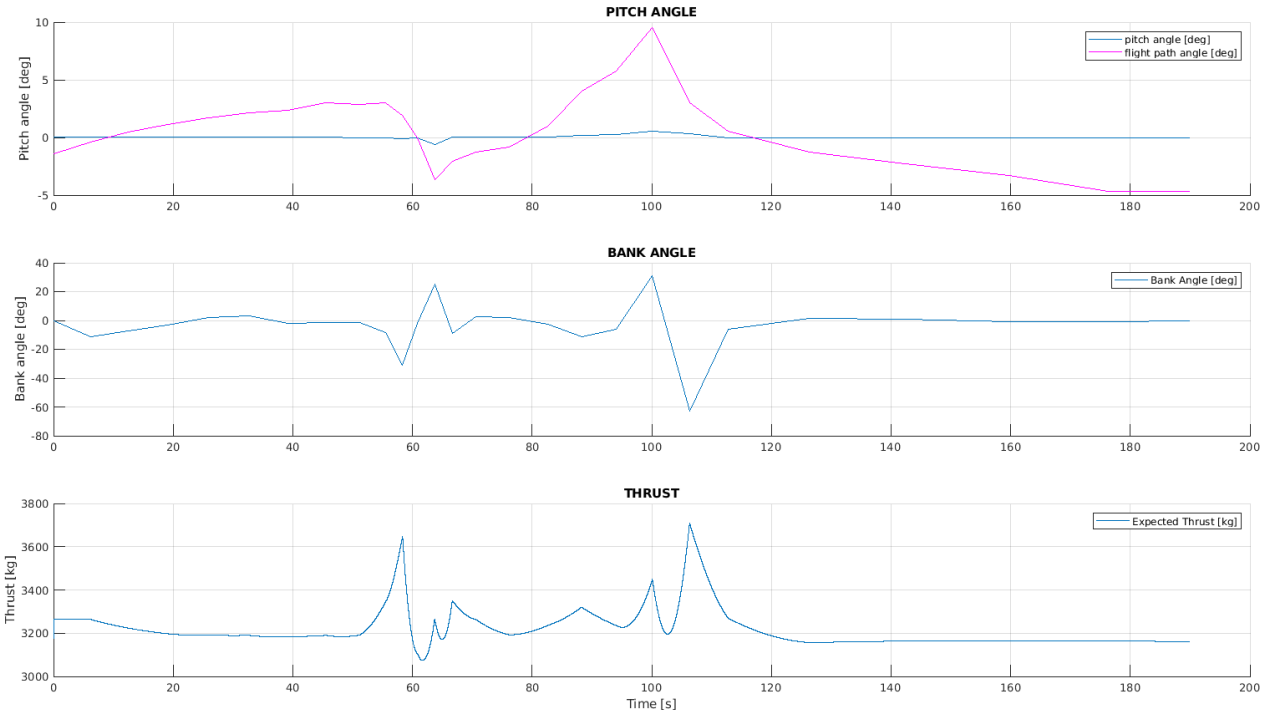

Figure 10: Test 3 angles and thrust

through intermediate assigned waypoints assuring a safety distance from terrain.
When the scout drone detects a new obstacle, the planner successfully computes a new safe path in short time.
Smoothing the trajectory is proved to allow the decrease of path length and the effective practicability of the route by a human-operated vehicle. This aspect will be further explored in future work, that will involve tests performed by expert pilots at a Virtual Reality flight simulator to assess the feasibility of the trajectory in terms of applied forces and load factor.

ACKNOWLEDGEMENTS

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.

COPYRIGHT STATEMENT

References

[1] L Raatiniemi, J Liisanantti, M Tommila, S Moilanen, P Ohtonen, M Martikainen, V Voipio, Janne Reitala, and T Iirola. Evaluating helicopter emergency medical missions: a reliability study of the HEMS benefit and NACA scores. *Acta anaesthesiologica Scandinavica*, 61(5):557–565, 2017.

[2] USHST. Review of 2018 u.s. fatal accident data, 2019. https://ushst.org/reports/.

[3] N. Avi, A.and Frisco, M. Giurato, M. Lovera, P. Masarati, S. Panza, G. Parnisari, F. Roncolini, M. Sesana, and G. Quaranta. Scout drone: a drone-helicopter collaboration to support HEMS missions. In *Proceedings of the 48th European Rotorcraft Forum*, Winterthur, Swizerland, 2022.

[4] Hualong Tang, Yu Zhang, Vahid Mohmoodian, and Hadi Charkhgard. Automated flight planning of high-density urban air mobility. *Transportation Research Part C: Emerging Technologies*, 131:103324, 2021.

[5] Sunan Huang and Rodney Swee Huat Teo. 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)*, pages 1218–1223. IEEE, 2019.

[6] Luciano Blasi, Egidio D'Amato, Massimiliano Mattei, and Immacolata Notaro. Path planning and real-time collision avoidance based on the essential visibility graph. *Applied Sciences*, 10(16):5613, 2020.

[7] Abdul Majeed and Sungchang Lee. A fast global flight path planning algorithm based on space circumscription and sparse visibility graph for unmanned aerial vehicle. *Electronics*, 7(12):375, 2018.

[8] Zahoor Ahmad, Farman Ullah, Cong Tran, and Sungchang Lee. Efficient energy flight path planning algorithm using 3-d visibility roadmap for small unmanned aerial vehicle. *International Journal of Aerospace Engineering*, 2017, 2017.

[9] Evgeni Magid, Roman Lavrenov, and Ilya Afanasyev. Voronoi-based trajectory optimization for ugv path planning. In *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, pages 383–387. IEEE, 2017.

[10] Fei Yan, Yi-Sha Liu, and Ji-Zhong Xiao. Path planning in complex 3d environments using a probabilistic roadmap method. *International Journal of Automation and computing*, 10(6):525–533, 2013.

[11] Li Li, Hong Zhan, and Yongjing Hao. The online path planning method of uav autonomous inspection in distribution network. In *E3S Web of Conferences*, volume 256, page 01047. EDP Sciences, 2021.

[12] Yucong Lin and Srikanth Saripalli. Path planning using 3d dubins curve for unmanned aerial vehicles. In *2014 international conference on unmanned aircraft systems (ICUAS)*, pages 296–304. IEEE, 2014.

[13] Olzhas Adiyatov, Kazbek Sultanov, Olzhas Zhumabek, and Huseyin Atakan Varol. Sparse tree heuristics for rrt* family motion planners. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1447–1452. IEEE, 2017.

[14] Franklin Samaniego, Javier Sanchis, Sergio García-Nieto, and Raúl Simarro. Recursive rewarding modified adaptive cell decomposition (rr-macd): a dynamic path planning algorithm for uavs. *Electronics*, 8(3):306, 2019.

[15] Daniele Palossi, Michele Furci, Roberto Naldi, Andrea Marongiu, Lorenzo Marconi, and Luca Benini. An energy-efficient parallel algorithm for real-time near-optimal uav path planning. In *Proceedings of the ACM International Conference on Computing Frontiers*, pages 392–397, 2016.

[16] Zhe Zhang, Jian Wu, Jiyang Dai, and Cheng He. A novel real-time penetration path planning algorithm for stealth uav in 3d complex dynamic environment. *IEEE Access*, 8:122757–122771, 2020.

[17] Shau Shiun Jan and Yu Hsiang Lin. Integrated flight path planning system and flight control system for unmanned helicopters. *Sensors*, 11(8):7502–7529, 2011.

[18] Anthony Stentz. Optimal and efficient path planning for unknown and dynamic environments. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1993.

[19] Alex Nash, Sven Koenig, and Craig Tovey. Lazy theta*: Any-angle path planning and path length analysis in 3d. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 147–154, 2010.

[20] Zhe Zhang, Jun Wang, Jianxun Li, and Xing Wang. Uav path planning based on receding horizon control with adaptive strategy. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 843–847. IEEE, 2017.

[21] S Hartjes, HG Visser, and MD Pavel. Optimization of simultaneous non-interfering rotorcraft approach trajectories. 2009.

[22] Hao Zhou, Hai-Ling Xiong, Yun Liu, Nong-Die Tan, and Lei Chen. Trajectory planning algorithm of uav based on system positioning accuracy constraints. *Electronics*, 9(2):250, 2020.

[23] Muhammad Toaha Raza Khan, Malik Muhammad Saad, Yang Ru, Junho Seo, and Dongkyun Kim. Aspects of unmanned aerial vehicles path planning: Overview and applications. *International Journal of Communication Systems*, 34(10):e4827, 2021.

[24] Utkarsh Goel, Shubham Varshney, Anshul Jain, Saumil Maheshwari, and Anupam Shukla. Three dimensional path planning for uavs in dynamic environment using glow-worm swarm optimization. *Procedia computer science*, 133:230–239, 2018.

[25] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[26] Yufeng He, Qinghua Zeng, Jianye Liu, Guili Xu, and Xiaoyi Deng. Path planning for indoor uav based on ant colony optimization. In *2013 25th Chinese control and decision conference (CCDC)*, pages 2919–2923. IEEE, 2013.

[27] Na Lin, Jiacheng Tang, Xianwei Li, and Liang Zhao. A novel improved bat algorithm in uav path planning. *J. Comput. Mater. Contin*, 61:323–344, 2019.

[28] Mehmet Hasanzade and Emre Koyuncu. A dynamically feasible fast replanning strategy with deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, 101(1):1–17, 2021.

[29] Firas A Raheem, Umniah I Hameed, et al. Path planning algorithm using d* heuristic method based on pso in dynamic environment. *American Academic Scientific Research Journal for Engineering, Technology, and Sciences*, 49(1):257–271, 2018.

[30] Ning Ma, Yunfeng Cao, Xinyao Wang, Zhaoyang Wang, and Houjun Sun. A fast path re-planning method for uav based on improved a* algorithm. In *2020 3rd International Conference on Unmanned Systems (ICUS)*, pages 462–467. IEEE, 2020.

[31] Yang Chen, Wei Li, and Rui Qi. 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)*, pages 746–751. IEEE, 2021.

[32] Taufik Khuswendi, Hilwadi Hindersah, and Widyawardana Adiprawita. 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*, pages 1–6. IEEE, 2011.

[33] Sabitri Poudel and Sangman Moh. Hybrid path planning for efficient data collection in uav-aided wsns for emergency applications. *Sensors*, 21(8):2839, 2021.

[34] Jinchao Chen, Mengyuan Li, Zhenyu Yuan, and Qing Gu. An improved a algorithm for uav path planning problems. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 958–962. IEEE, 2020.

[35] `https://earthexplorer.usgs.gov/`.

[36] Neal B Gallagher. Savitzky-golay smoothing and differentiation filter. *Eigenvector Research Incorporated*, 2020.

[37] Abhijeet Ravankar, Ankit A Ravankar, Yukinori Kobayashi, Yohei Hoshino, and Chao-Chung Peng. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors*, 18(9):3170, 2018.

[38] `https://www.mbdyn.org/`.