An Obstacle Avoidance Approach for UAV Path Planning

Nouman Bashir*, Saadi Boudjit*, Gabriel Dauphin*, and Sherali Zeadally†
*L2TI, Institut Galilée, Université Sorbonne Paris Nord, Villetaneuse 93430, France.
†University of Kentucky, Lexington, KY 40506-0224, USA. szeadally@uky.edu
Corresponding author: Nouman Bashir (email:nouman.bashir@edu.univ-paris13.fr)

***Abstract*—The recent pandemic of COVID-19 has proven to be a test case for Unmanned Aerial Vehicles (UAVs). UAVs have shown great potential for plenty of applications in the face of this pandemic, but their scope of applications becomes limited due to the dependency on ground pilots. Irrespective of the application, it is imperative to have an autonomous path planning to utilize UAVs to their full potential. Collision-free trajectories are expected from the path planning process to ensure the safety of UAVs and humans on the ground. This work proposes a path planning technique where collision avoidance is mathematically proven under an uncertainty prerequisite, that the UAV follows its requested moving position within some threshold distance. This scheme ensures UAV safety by considering the underlying control's system overshoots. Obstacles play a guiding role in selecting collision-free trajectories. These obstacles are modeled as rectangular shapes with interest points defined around their corners. These points further define collision-free permissible edges, and later we apply the Dijkstra algorithm to these edges before having the desired trajectory. Regardless of the size of deployment area, our proposed scheme incurs low computational load due to the dependency on pre-defined interest points only thereby making it suitable for real-time path planning. Simulation results obtained using MATLAB's UAV Toolbox show that the proposed method succeeds in getting short collision-free trajectories.**

***Index Terms*—Control of a UAV network, collision-free trajectories, non-heuristic motion-planning, Unmanned Aerial Vehicle (UAV), UAV motion-planning.**

I. Introduction

OWING to rapid technological advancements, Unmanned Aerial Vehicles (UAVs) are emerging as a reliable choice in various application domains. This is substantiated by emergence of recent UAV deployments in a wide range of applications. In addition to the military domain, UAVs are playing their part in the realm of civil, public, and personal [1]–[5]. Transportation of medical equipment, lab sample collection, spraying over different areas, and public space monitoring with guidance are novel use cases during the ongoing pandemic of COVID-19 [6], [7]. There are numerous other applications that are not limited to surveillance, post-disaster assessment, and rescue operations [8]–[11]. The future may see UAVs delivering goods and collect data from the Internet of Things (IoT) devices as in [66]. Minimizing energy consumption is one of the major research goals in recent research efforts [12] and [13]. In the future, UAVs may be used to provide additional computing resources [14].

In many countries, rules and regulations for flying aerial UAVs in public areas are still in the phase of development [15]. Nonetheless, a pilot on the ground or UAVs with an autopilot system on-board should be capable of avoiding flying over unsafe, risky, and no-fly zones [16], [17]. Smart autonomous path planning and navigation become inevitable because it is not always possible to keep the flying UAV in the line of sight of a pilot. Furthermore, the use of autonomous navigation also reduces the probability of human errors from the system.

Path planning is an important issue which must be considered during UAV mission planning [18]. Many of the currently deployed applications for UAVs have autopilot functionalities along with the capability to fly them according to the pre-planned path or even make real-time decisions in case of any unforeseen scenario [19], [20]. However, some of the commercially used UAVs still use off-board pilots and fixed trajectories. In the literature, researchers have proposed three-dimensional path planning approaches to navigate UAVs in the presence of obstacles.

In this work, we focus on the ability of a UAV to travel to a target while avoiding obstacles. Some of these methods used to achieve this objective have a learning capacity and may adapt to unknown obstacles and to specific UAV navigation capabilities. They may even discover by themselves navigation techniques using a method such as Reinforcement Learning (RL) [21] [22]. These RL based navigation techniques generally use Markov decision processes along with time-difference or Monte-Carlo methods and the information learned is stored in a Look Up Table (LUT). Hsu et al. [66] also used RL for collision avoidance.

As surveyed in [23] and demonstrated in [24] and [13], neural networks, (NN), have also been used to improve reinforcement learning because they improve performance, replace the LUT, and support high-dimensional inputs. Samir et al. [68] used the Actor-Critic (AC) algorithm of the deep RL algorithm to direct the flight route of UAVs on highways without communication infrastructure. To successfully optimize vehicular coverage, the given framework comprehends the dynamics of the vehicular environment and the ideal trajectory of the deployed UAVs. It is precisely their ability to adapt to a wide variety of contexts, that makes their behavior difficult to predict, and, in the same way, it makes their use inappropriate in addressing collision avoidance as a safety issue.

Other methods, grouped into two classes, formulate path planning as an optimization problem in [25], [26], [66], [69]. Heuristic methods are those wherein the optimality of the solutions compromises for better computational time

efficiency. We include in this first class, techniques using NN, RL, AC when they are derived from an optimization problem as in [66], [69]. Non-heuristic approaches, however, provide optimal solutions but demand high computational resources. In these techniques, the process starts by breaking down the whole detected area or map into computational domains via tool such as matrix decomposition. This data is utilized further in developing possible UAV trajectories [8].

Path planning is a way to find a feasible, optimal/near-optimal, shortest, smooth, and a low-cost path between a starting point and the desired destination point by considering specific operational constraints [19]. These constraints usually involve velocity, acceleration, environmental disturbances such as wind, sensor uncertainties, and flying over restricted areas. From a control view point, we distinguish two kinds of UAVs depending on whether they have a minimum speed (i.e. fixed wings, as described in [54], [55]). We focus on UAVs with hovering capacities as in [17] using quadrotors. Generally, path planning consists of two stages, namely, graph building and pathfinding [16]. A graph construction considers the start and end points, and all available vertices obtained by matrix decomposition or area tessellation.
Different tessellation resolutions result in a different number of vertices which in turn determine the computational work needed [8]. A higher number of vertices require a high computational load, while a lower value requires a smoothing process to get stable trajectories [19]. This smoothing process is necessary to avoid having any acute angle turns within a path. The pathfinding process assigns the respective cost of each vertex and selects a flight path with a minimum overall cost. In this context, Dijkstra, A* or genetic algorithms are applied to these vertices to get optimal trajectories [16]. By considering the entire environment during the tesselation process, these algorithms incur a higher computational load for the existing works.

While on a mission, a UAV may collide with other UAVs or nearby obstacles. UAVs must have an obstacle avoidance mechanism to prevent collisions by maintaining a safe distance from nearby objects. The efficiency of this mechanism highly relies on the accurate operation of positioning sensors. These sensors help UAVs in the navigation process by providing positioning information during the entire flight duration. The Global Navigation Satellite System (GNSS) is the most commonly used sensor in this regard [27]. The precision of GNSS is greatly influenced by the multipath effect and by the blockage of the received satellite signal. GNSS operates well in open areas, whereas these conventional methods cannot be relied upon in urban areas having large buildings with a higher probability of signal blockage. UAVs employed for civilian application need to pay more attention to the precision of localization values which, on the other hand, could prove to be very dangerous for the safety of humans on the ground [28], [29]. Many existing works [30]–[32] have proposed a collaborative architecture of GNSS with active sensors, such as LiDAR, to mitigate the multipath effect. However, these architectures incur high deployment costs and have high computational load and memory needs [33].

Environmental disturbances or sensor uncertainties and, specifically, their role in UAV path planning is an outstanding research field that needs further investigation. These uncertainties may arise from GNSS values, wind speed, or variation in the velocity of UAVs [34]. In this paper, we assume that uncertainties related to the environment, the sensors, and the navigation capabilities are modeled into an uncertainty prerequisite, that is, the existence of a threshold distance between the UAV and its requested moving position. We propose a collision-free path planning technique that reduces the computational load required by replacing area tessellation with interest points connected into a graph. This computational load reduction is a desired feature for applications requesting the UAV to reach targets whose locations are provided with an ongoing flow of information. These interest points are located around each corner of rectangles, each modeling an obstacle and all together model the environment. It is worth noting that obstacles with complex shapes can also be approximated by a union of overlapping rectangles.

We summarize the main research contributions of this work as follows:

- We model uncertainties arising due to environmental disturbances as an uncertainty prerequisite and we consider them in the UAV path planning process to design collision-free trajectories. This uncertainty prerequisite as a guard distance and protects the UAV from collision even if the underlying control system fails to meet its defined overshoot limitations.
- The proposed path planning method considers obstacles in an environment as rectangles because a tool like a visibility graph allows modeling of the environment as a graph. We seek guidance from obstacles by including their corners as graph nodes into the area where the tesselation process occurs instead of considering the entire environment, thus reducing the computational load on the path planning process.
- We demonstrate the efficacy of the proposed scheme, in terms of reduced computational load, shorter and collision-free trajectories, through simulations carried out using the MATLAB's UAV Toolbox.

We organize the rest of the paper as follows. Section II presents related works on UAV path planning and in particular in the presence of obstacles. Section III describes the problem statement and main contributions of this work, followed by Section IV which presents the mathematical model of our proposed scheme. Section V describes the testbed used and experimental procedures. Section VI discusses the results obtained. Finally, Section VII concludes the paper.

II. Related Work

A robust UAV path planning strategy must possess important attributes which should provide a computationally efficient solution while complying with the given constraints. The strategy development depends on different planning requirements such as real-time planning, performance optimization, risk minimization, and obstacle avoidance [35], [36]. Many path planning techniques are available in the literature that leverage results from other research fields such as potential

field algorithms from physics, probabilistic approaches from mathematics, and graph-based solutions from the computer science field. Many traditional path planning techniques have been proposed such as artificial potential field, probability roadmap, and rapidly exploring trees methods [34].

The Artificial Potential Field (APF) [37] path planning is a popular method to avoid obstacles having a concise mathematical model and simple algorithm structure [27]. It creates an attractive and repulsive field for destination and restricted areas, respectively, and the route by displacement is planned based on the resultant force. [38] is one of the first publications on APF path planning. Besides its application for single UAV, the APF approach applies to multi-UAV systems as well. The APF approach faces local minima and destination unreachable scenarios due to closely spaced obstacles and the presence of an obstruction between the destination and a UAV respectively.

Many improvements [39]–[42] have since been made to the original APF. In [43], the authors developed an APF based reactive controller for UAVs to avoid collision with terrain as well as from each other. Their scheme models obstacles as points with latitude, longitude, and altitude information provided by a Digital Elevation Map. To address the issue of local minimum in APF, the authors of [44] present an improved artificial potential field that finds an optimal path and avoids collisions with obstacles. In [45], the authors proposed a hybrid model involving APF and optimal control theory. The additional force introduced is an optimization variable that transforms path planning into an optimization problem. The optimal control law is applied to solve the optimization problem after converting the constrained problem into an unconstrained optimization problem.

Probabilistic RoadMap (PRM) and Rapidly Exploring Trees (RRT) come under the domain of sample-based path planning algorithms [46]. In PRM, at first, sampling of the configuration space is done using a probabilistic model. A connected graph is created by applying a local planner which connects the configuration sampled to the nearest configuration space. At last, any graph search algorithm can be applied to the connected graph to get a possible path from a start point to the desired endpoint. The authors of [47] proposed a PRM based 3D path planning approach for a complex environment. The octree algorithm divides the configuration space into voxels. The PRM random method selects samples from all the available voxels. The connected graph produced by the local planar is utilized further by the A* algorithm to have a feasible path. RRT is another path planning method that uses random spatial sampling for high-dimensional spaces. RRT grows a tree with its roots at the start configuration. With each sample taken, the tree grows to include more feasible trajectories. Many RRT-based path planning approaches have been reported in the literature [48]–[51].

Graph-based search techniques are extensively explored in many fields and are popular in the UAV path planning domain [46]. In these approaches, a grid map represents the entire environment. Depending on the presence of obstacles, each cell in the grid represents either an occupied or a free cell. Any graph exploration algorithm can be applied to the graph to find a feasible path between the start and destination cell. Fast searching capabilities make these algorithms very useful for real-time path planning, but the generation of non-smooth trajectories renders them inefficient for large environments. Dijkstra algorithm finds the shortest path between the start cell and the destination cell [52], and many graph-based path planning techniques, in conjunction with the Dijkstra algorithm, have been implemented [53]–[56]. A* is another graph traversal or path search algorithm which has attributes of optimality and completeness [57]. The performance of the A* algorithm highly depends on the heuristic function used. Recent works such as [16], [27] are the few path planning implementations involving the A* algorithm. Several other graph-based algorithms are available in the literature that finds smooth flight trajectories by incorporating some smoothing process [19], [35].

Furthermore, some studies have improved path planning of UAVs in a Vehicular Ad Hoc Network (VANET) environment. For instance, Samir et al. in [67] investigate RL based trajectory planning for UAVs in a specific roadway segment. The authors carefully evaluated the unknown area and mapped out the trajectory with the fewest possible UAVs to provide network connectivity for vehicles while minimizing UAV energy usage. To further this investigation, Samir et al. in [68] used the Actor-Critic algorithm of the Deep Reinforcement Learning (DRL) algorithm, in particular, to manage the flight route of UAVs on highways without communication infrastructure. To successfully optimize vehicular coverage, the given framework comprehends the dynamics of the vehicular environment and the ideal trajectory of the UAVs deployed. Jiang et al. [69] discussed path planning and collision avoidance for UAVs in a 3D environment. In this study, the authors first modeled the path planning as a constraint optimization problem and later drone to drone collision avoidance is modeled as Markov Decision Process (MDP). Recently, Tu et al. in [70] used RL for path planning and obstacle avoidance for a single UAV in the 3D environment. This work primarily focuses on how UAVs fly and address the application of UAV for sea farming. Moreover Zhou et al. [65] also studied a path planning for UAVs in 3D environment. The authors developed a real UAV autonomous obstacle avoidance path planning experimental platform in which a flight test for UAV obstacle avoidance path planning based on real-world conditions is conducted.

The computational needs of path planning algorithms grow exponentially with an increase in the dimensional size of the configuration space along with the reduction in the operational time [27], [58]. Moreover, in algorithms such as APF, an improper definition may lead to a local minimum or unreachable destination scenarios [59]. Most path planning techniques rarely consider environmental disturbances and sensor uncertainties. In [27], the authors considered positioning errors that require an additional map of predicted satellite positions generated by a 3D building model. All path planning approaches, including that do consider positioning error maps in their UAV path designs, still rely entirely on the UAV's control system to track the desired trajectory. None of the existing approaches have addressed overshooting of a UAV beyond its desired control system's limitations. To address

TABLE I
SUMMARY OF VARIOUS PATH PLANNING METHODS FOR UAVS

Reference	Research objective	Approach	Strengths	Weaknesses
Oland et al. 2013. [43]	Collison and terrain avoidance with multiple UAVs	Kinematic model with dynamic feedback linearization using APF	Maintaining rigid formation and low altitude flight in a rolling terrain Uniformly stable controller	Need of a digital elevation map Need of precise definition for APF
Lifen et al. 2016. [44]	Address local minima issue in APF	Collision free trajectories using APF with change in repulsive potential function	Optimal path especially in complex environment	Dependent on obstacle size Need of precise definition for APF
Chen et al. 2016. [45]	Solution to additional control force in APF to deal with time varying variable	Remodeling the functional optimization model by taking the additional control force as an independent variable	Solution to dead-end problem in APF Shorter and smoother trajectories with irregular obstacles	Dependent on precise definition of repulsive and attractive potential
Yan et al. 2013. [47]	Path planning in 3D environment with less time complexity	Octree algorithm to divide the work space into voxels and random selection of free voxels	Allows UAVs to fly through narrow-passage areas	Random selection of voxel may lead to longer paths
Maini et al. 2016. [53]	Obstacle free paths satisfying UAV kinematic constraints	Visibility graph to represent environment and validation for maximum steering angle	Suitable for on-line implementation Easy validation of the steering angle constraint	No tolerance for sensor uncertainties No definition as how to add points of obstacles as vertices Steering angle constraint may result in longer paths
Zhu et al. 2021. [12]	Traveling salesman problem and path planning in the context of energy minimization	Actor-critic algorithm, sequence-to-sequence learning approach, recurrent neural network	Significant decrease in energy consumption	High computational complexity calling for the use of Google Cloud Platform with a NVIDIA TESLA P100 Graphical Processing Unit (GPU)
Our proposed method	Collision-free trajectories taking environmental disturbances into consideration	Defines interest points around rectangular obstacles while taking environmental disturbances into consideration	Shortest possible linear paths Low configuration space complexity Tolerant to sensor uncertainties Suitable for real-time path planning Is aware if the underlying control system fails to track the desired trajectory within the UAV's control system's overshoot limitations	Does not incorporate UAV kinematics into the path planning model (will be addressed in our future work)

these shortcomings, in this article, we propose a simple yet robust path planning method. We model obstacles as rectangles, and for the configuration space, instead of involving the entire environment, we only include interest points defined around corners of the rectangles[1]. Additionally, we consider a disk-based uncertainty scheme to make our model resilient to uncertainties arising as a result of environmental disturbances. Table I provides a summary of various path planning approaches for UAVs. For each approach, this table enlists the research objective, the method adopted, strengths, and weaknesses.

III. PROBLEM STATEMENT AND RESEARCH CONTRIBUTIONS OF THIS WORK

Intended Use Case

Our proposed path planning technique assumes that obstacles and the target are static and are well-known before departure. It also assumes the navigation's system ability to maintain itself at a distance no greater than ρ from a moving point, while being robust to environment disturbances, direction changes and avoiding moving objects of small sizes. By executing the following steps, the proposed technique could nonetheless be used in the following use case wherein the UAV needs to travel to a known target while avoiding unknown static obstacles all with limited energy availability.

1) The target is defined by some GNSS coordinates and it is assumed to be at a safe distance denoted by ρ of any neighboring obstacles. Obstacles are assumed to be static, except for objects (such as birds) of small size that are not moving too fast. The UAV is equipped with a GNSS sensor and a front camera.
2) Several photos are captured while remaining at the same position and rotating on itself. The shapes and sizes of all obstacles that are in the line-of-sight visible and within a horizon-distance denoted by ρ_h are then extracted as described in [61]. These estimates and their precision are converted into a 2D-map composed of a set of temporary rectangles covering all possible obstacle positions.
3) If the actual target appears outside this 2D-map or if it is contained or is too close to any temporary rectangle, a temporary target is defined according to the Bug1-algorithm [62]. If no obstacles prevent reaching the actual target, then the temporary target is defined as the actual target's closest point. If there is an obstructing obstacle, then it is defined as the closest point among the points being ρ-distant from the obstacle's contour.
4) The UAV checks if it has sufficient energy to reach the temporary target and get back to its starting position. If

[1]Most obstacles (buildings, no fly-zones) can be approximated by rectangles [16], [44], [53], and tools like visibility graphs [60] allow us to view polygonal obstacles as graphs. We can successfully represent a polygone (e.g. triangle) by a set of thin rectangles, one for each side)

it does not, it returns right away to recharge its battery.
5) The computer system inside the UAV applies the proposed technique to yield a path leading to the temporary target.
6) During the flight and until reaching the temporary target, the UAV's navigation system follows the path at a constant speed, it also uses its camera to avoid birds, while always keeping a distance no greater than ρ from the path. ρ should be set large enough to allow the navigation system to ensure this and all other adversities.
7) If the point reached is a temporary target different from the actual target, then we go back to step 2, else the mission is accomplished.

Problem Statement

The main objective of this work is to develop a robust and computationally low UAV path planning technique. We model obstacles as rectangles, and unlike other methods, we seek guidance from the interest points defined around their corners. These interest points are defined by considering the threshold distance in the uncertainty prerequisite. Contrary to existing path planning techniques, wherein every cell of the grid environment represents the vertex of a graph, our model only considers defined interest points as vertices. Fig. 1 shows how a graph-based path planning technique configures the environment compared to our proposed approach. Fig. 1 depicts a 15×16 square units environment wherein the left side of the figure shows the configuration space adopted by the graph-based technique. It divides the entire area into 240 cells that constitute the vertices of a graph. The right side of Fig. 1 shows how the proposed scheme delineates interest points around the obstacles. It selects only 20 interest points that become vertices of a graph. Permissible obstacle-avoiding edges are defined, which in collaboration with the Dijkstra algorithm formulate a path. Fig. 2 shows the uncertainty prerequisite modeling a disk, centered on the requested moving position and, whose radius is the threshold distance. The dotted line in Fig. 2 shows the desired trajectory between the start and the goal point. This trajectory is planned by including two interest points, i.e., P_1 and P_2. The UAV is assumed to be present within a disk area of radius ρ centered on each point of the trajectory. In Fig. 2, the two solid lines surrounding a dotted line shows the location of the UAV. As these lines are quite far from any obstacle, it exemplifies the underlying mechanism ensuring collision avoidance.

IV. Proposed Scheme

This section describes our proposed scheme, corresponding to items 4 and 5 described in section III. Subsection IV-A defines the way we model obstacles as rectangles and the selection criteria for interest points. Subsection IV-B introduces a suboptimal solution that finds admissible collision-avoiding trajectories while employing the Dijkstra algorithm. This section concludes with a discussion on the location of interest points defined.

A. Optimization Problem Statement

The design of the trajectory is regarded as an optimization problem, that of leaving from a starting point A at $t = 0$ and reaching the fastest way to an ending point B, while avoiding all obstacles modeled as rectangles of different widths, lengths and orientations. These rectangles are denoted $\mathcal{R}_1 \dots \mathcal{R}_R$ and defined as the set of points inside and on their borders. The first assumption we are making here is that A, B, $\mathcal{R}_r$ are all known to the UAV before its departure. We make use of a time-dependent virtual point denoted as $V(t)$ defined on $[0, T]$ where T is the time of flight. This virtual point moves at a speed no greater than v.

$$\forall t_1, t_2 \in [0, T],\, \mathrm{d}(V(t_2), V(t_1)) \leq v\,|t_2 - t_1| \tag{1}$$

where d is the usual Euclidean distance. The second assumption, illustrated in Fig. 3, states that the UAV navigation system can follow $V(t)$ by staying at a distance strictly below ρ during the time of the flight.

$$\forall t \in [0, T],\, \mathrm{d}(V(t), D(t)) < \rho \tag{2}$$

The collision of the UAV with an obstacle is modeled as:

$$\exists r \leq R, \quad \exists t \in [0, T], \quad D(t) \in \mathcal{R}_r \tag{3}$$

Using assumption 2, equation (3) is derived into a constraint on $V(t)$ ensuring collision avoidance.

$$\forall r \leq R,\, \forall t \in [0, T], \quad \mathrm{d}\left(V(t), \mathcal{R}_r\right) \geq \rho \tag{4}$$

We denote the border of each rectangle $\mathcal{R}_r$ as $\partial\mathcal{R}_r$, it is the set of points on one of the four line segments bordering $\mathcal{R}_r$. The following theorem shows why the problem statement concerns only the borders of the rectangles.

Theorem 1: Let $M(t)$ be a continuous mapping from $[0, T]$ to $\mathbb{R}^2$ and $\mathcal{R}$ a rectangle with $\partial\mathcal{R}$ as border.

$$\left.\begin{array}{ll} \mathrm{d}\left(M(0), \mathcal{R}\right) > 0 \\ \forall t \in [0, T] \quad \mathrm{d}\left(M(t), \partial\mathcal{R}\right) \geq \rho \end{array}\right\} \quad \Rightarrow \quad \forall t \in [0, T] \;\; \mathrm{d}\left(M(t), \mathcal{R}\right) \geq \rho$$

We present a sketch of the proof in appendix A. Each rectangle border $\partial\mathcal{R}_r$ is also defined by its four line segments denoted as $C_{4r-3}D_{4r-3}$, $C_{4r-2}D_{4r-2}$, $C_{4r-1}D_{4r-1}$, and $C_{4r}D_{4r}$. The collection of all line segments is denoted C_iD_i with $i \in \mathcal{I}$. In order to apply theorem 1, we add a third assumption, that the beginning position is far enough from all obstacles.

$$\forall r \leq R, \mathrm{d}(A, \mathcal{R}_r) \geq \rho \tag{5}$$

with $\rho > 0$. We considered a set of admissible trajectories denoted as $\mathcal{V}$. Fig. 4 illustrates an example of an admissible trajectory.

$$\mathcal{V} = \Big\{ V \in \left(\mathbb{R}^2\right)^{[0,T]} \;\Big|\; \begin{array}{ll} V(0) = A & \\ V(T) = B & \\ \forall t_1, t_2 \in [0, T], & \mathrm{d}(V(t_2), V(t_1)) \leq v\,|t_2 - t_1| \\ \forall i \in \mathcal{I},\, \forall t \in [0, T], & \mathrm{d}\left(V(t), \overline{C_iD_i}\right) \geq \rho \end{array} \Big\} \tag{6}$$

$\mathrm{T}_{\mathcal{V}}$ operates on $\mathcal{V}$ and yields the time of flight.

$$\mathrm{T}_{\mathcal{V}}(V) = \min\left\{t > 0 \,|V(t) = B\right\} \tag{7}$$

The optimization problem is finding:

$$V^* = \underset{V \in \mathcal{V}}{\operatorname{argmin}}\, \mathrm{T}_{\mathcal{V}}(V) \tag{8}$$

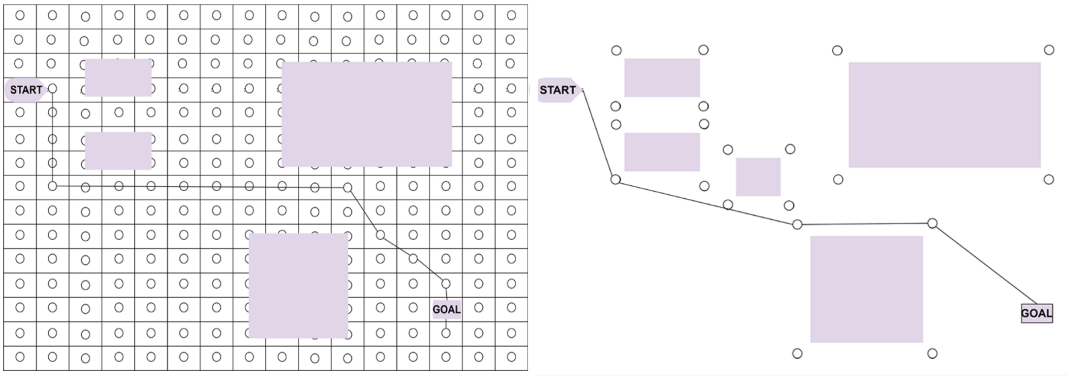

Fig. 1. Comparison of configuration space using the graph-based approach (on the left) and our proposed scheme (on the right)

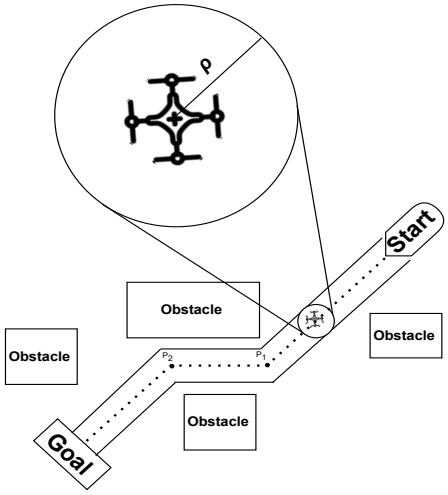

Fig. 2. Illustration of the trajectory designed by the proposed scheme with the uncertainty model

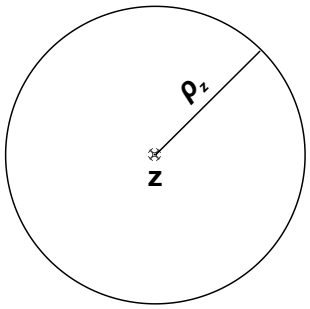

Fig. 3. Disk centered on $V(t)$ where the UAV is assumed to be according to assumption 2.

B. Suboptimal solution with the Dijkstra algorithm

To make the optimization problem more tractable, we make a fourth assumption. The trajectory is a set of connected line segments joining A and B, where the segment ends are chosen among a predefined set of points denoted as $(\mathcal{P}_j)_{j\in\mathcal{J}}$ and $V(t)$ moves at a speed v on each line segment. Denoting $\mathcal{V}' \subset \mathcal{V}$ as the mappings fulfilling this third assumption, we get a weighted graph representation of $\mathcal{V}'$ as Fig. 5 shows. This graph is denoted as $((P_j)_j, \mathcal{E}, \mathrm{T}_\mathcal{E})$.

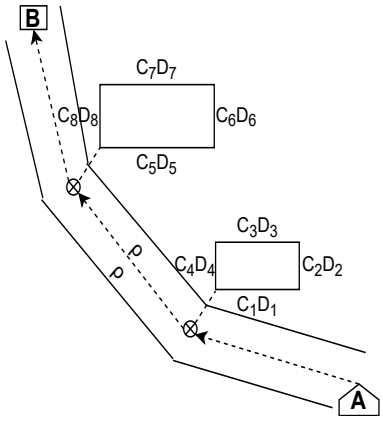

Fig. 4. $V(t)$ is on the line joining A and B. The contour surrounding this line delineates the set of all points at a distance below ρ, where the UAV adheres to assumption 2.

- $A \in \{P_j | j \in \mathcal{J}\}$ is the root.
- $B \in \{P_j | j \in \mathcal{J}\}$ is the sink.
- $(P_{j_1}, P_{j_2}) \in \mathcal{E}$ is an admissible edge connecting P_{j_1} and P_{j_2} if the line segment $\overline{P_{j_1}P_{j_2}}$ is at a distance equal to or greater than ρ of any obstacle.

$$(P_{j_1}, P_{j_2}) \in \mathcal{E} \quad \Leftrightarrow \forall i \in \mathcal{I},\, \mathrm{d}(\overline{C_i D_i}, \overline{P_{j_1}, P_{j_2}}) \geq \rho \quad (9)$$

- Each edge is given a value which is the time of travel.

$$\mathrm{T}_\mathcal{E}((P_{j_1}, P_{j_2})) = \frac{1}{v}\,\mathrm{d}\,(P_{j_1}, P_{j_2}) \quad (10)$$

The following theorem shows the equivalence between finding the suboptimal solution $\widehat{V} = \underset{V\in\mathcal{V}'}{\operatorname{argmin}}\, \mathrm{T}_\mathcal{V}(V)$ and finding the least weighted path joining A and B. The latter is precisely what the Dijkstra algorithm solves efficiently.

Theorem 2: A mapping V in $\mathcal{V}'$ is a path $P_{j_1}P_{j_2}\dots P_{j_N}$ of $((P_j)_j, \mathcal{E}, \mathrm{T}_\mathcal{E})$ joining A and B and its time of flight is the sum of all weights of the edges traversed.

$$\mathrm{T}_\mathcal{V}(V) = \sum_{n=2}^{N} \mathrm{T}_\mathcal{E}((P_{j_{n-1}}, P_{j_n})) \quad (11)$$

We present a sketch of the proof in appendix B.

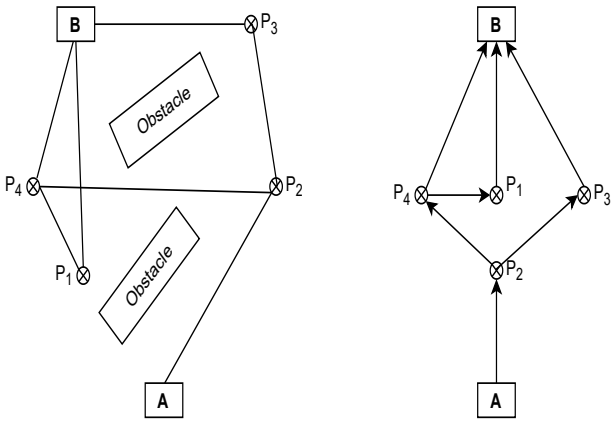

Fig. 5. Four paths joining A and B consistent with assumption 3, shown on the left as positions of $V(t)$ and on the right using a graph structure.

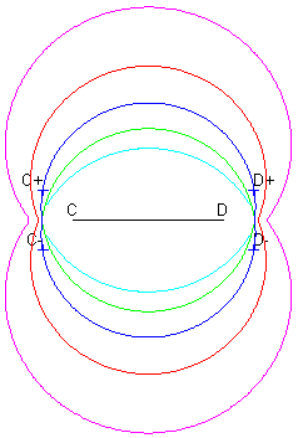

Fig. 6. Five loci of predefined points corresponding to, $\frac{3\pi}{2}-\frac{\pi}{5}$, $\frac{3\pi}{2}-\frac{\pi}{10}$, $\frac{3}{2}\pi$, $\frac{3\pi}{2}+\frac{\pi}{10}$, $\frac{3\pi}{2}+\frac{\pi}{5}$, when counting outwards. At the center is the horizontal obstacle of length $l=1$ and $\rho=0.1$. The proposed predefined points are C^-, C^+, D^-, D^+.

C. Location of the predefined points

The appropriate choice of (P_j) is crucial to the performance of the algorithm and their number is a trade-off between computational complexity and performance. Interest points (P_j) should be located near each obstacle's line-segment's end and organized to allow paths enclosing each obstacle. Considering a specific predefined point P and a unique obstacle $\overline{CD}$, not colliding line segments joining P have a range of valid angles, and the size of this range is denoted as $\Delta\Theta$.

$$\Delta\Theta = \left| \left\{ \Theta \,\middle|\, (\overrightarrow{MP}, \overrightarrow{CD}) = \theta \Rightarrow \mathrm{d}(\overline{MP}, \overline{CD}) \geq \rho \right\} \right| \quad (12)$$

Fig. 6 shows a horizontal obstacle of length $l=1$ and five different locus of predefined points, each associated with a specific value of $\Delta\Theta$: $\frac{3\pi}{2}-\frac{\pi}{5}$, $\frac{3\pi}{2}-\frac{\pi}{10}$, $\frac{3}{2}\pi$, $\frac{3\pi}{2}+\frac{\pi}{10}$, $\frac{3\pi}{2}+\frac{\pi}{5}$. It is worth noting that smaller values of $\Delta\Theta$ are associated with predefined points closer to the obstacle. Our proposition considers four predefined points for each obstacle which are sufficient to bypass the obstacle, located at a distance of $\rho\sqrt{2}$ of each segment end and having an angle of $\pm\frac{\pi}{4}$. These points, denoted as C^-, C^+, D^-, D^+ are located on the intermediate locus associated with $\Delta\theta=\frac{3\pi}{2}$ as Fig. 6 shows.

$$\begin{cases} \angle(\overrightarrow{DC}, \overrightarrow{CC^-}) = \frac{\pi}{4} & CC^- = \rho\sqrt{2} \\ \angle(\overrightarrow{DC}, \overrightarrow{CC^+}) = -\frac{\pi}{4} & CC^+ = \rho\sqrt{2} \\ \angle(\overrightarrow{CD}, \overrightarrow{CD^-}) = -\frac{\pi}{4} & CD^- = \rho\sqrt{2} \\ \angle(\overrightarrow{CD}, \overrightarrow{CD^+}) = \frac{\pi}{4} & CD^+ = \rho\sqrt{2} \end{cases} \quad (13)$$

The following theorem states that any path joining a predefined point with some appropriate angle ensures a safe distance from the obstacle.

Theorem 3: Let (C, D) be an obstacle, and C^-, C^+, D^-, D^+ be its associated predefined points and M be a given point.

$$\begin{aligned} &\angle(\overrightarrow{CD}, \overrightarrow{C^-M}) \in [\tfrac{\pi}{2}, 2\pi] \quad \Rightarrow \quad \mathrm{d}(\overline{CD}, \overline{C^-M}) \geq \rho \\ &\angle(\overrightarrow{CD}, \overrightarrow{C^+M}) \in [0, \tfrac{3\pi}{2}] \quad \Rightarrow \quad \mathrm{d}(\overline{CD}, \overline{C^+M}) \geq \rho \\ &\angle(\overrightarrow{CD}, \overrightarrow{D^-M}) \in [-\pi, \tfrac{\pi}{2}] \quad \Rightarrow \quad \mathrm{d}(\overline{CD}, \overline{D^-M}) \geq \rho \\ &\angle(\overrightarrow{CD}, \overrightarrow{D^+M}) \in [-\tfrac{\pi}{2}, \pi] \quad \Rightarrow \quad \mathrm{d}(\overline{CD}, \overline{D^+M}) \geq \rho \end{aligned} \quad (14)$$

It is worth noting that all four statements are consistent with $\Delta\Theta=\frac{3\pi}{2}$ because

$$\left|\left[\frac{\pi}{2}, 2\pi\right]\right| = \left|\left[0, \frac{3\pi}{2}\right]\right| = \left|\left[-\pi, \frac{\pi}{2}\right]\right| = \left|\left[-\frac{\pi}{2}, \pi\right]\right| = \frac{3\pi}{2}$$

We present a sketch of the proof in appendix C.

D. Computational complexity of the proposed technique

The computational complexity of the proposed technique is related to the number of obstacles, $|\mathcal{I}|$. Indeed, the graph is composed of $v=2+4|\mathcal{I}|$ vertices and at most $e=\frac{1}{2}(2+4|\mathcal{I}|)(1+4|\mathcal{I}|)$ edges. With an optimized implementation, the Disjkstra algorithm has a complexity, as calculated in [63], of $O(e+v\log(v))$, that is, $O(|\mathcal{I}|^2)$ or less. An interesting point is the low memory requirement (i.e., roughly $O(|\mathcal{I}|^2)$) as compared to when using a grid. We note that the proposed technique is often thought to be less computationally intensive because it depends on the number of obstacles instead of the number of grid points representing the area.

Furthermore, as section V-A has explained, our proposed technique does not require iterative evaluations in a hyperparameter space to meet some given constraints.

V. Simulation Testbed and Experimental Procedures

We conducted simulations under different environmental conditions by varying the number of obstacles and their dimensions, simulation area, UAV turning angles, UAV speed, start, and destination locations. We implemented the schemes we compared our proposed with using the MATLAB-R2021a's UAV Toolbox[2]. We used different grid sizes ranging from a

[2]https://www.mathworks.com/products/uav.html

TABLE II
SIMULATION PARAMETERS AND HARDWARE/SOFTWARE ENVIRONMENT

Parameter and hardware/software	Value
Grid size	400×600 – 20000×20000
Number of obstacles	4 – 19
ρ	10 m [43]
UAV model	multirotor
UAV speed limit	5 – 9 m/s [64]
Gain for heading controller	2
Roll angle limit	45°
Number of simulations	250 – 300
CPU	Intel(R) Core(TM) i5-6200U 2.30 GHz
RAM	12 GB
Operating system	Microsoft Windows 10 Pro

smaller configuration space of 400×600 to a larger one with 20000×20000 cells. Similarly, to evaluate schemes under a different number of obstacles, we varied the number of obstacles from a smaller value of 4 to a higher value of 19. We considered the multirotor UAV model to determine the trajectory tracking effectiveness of the proposed scheme. In all simulation environments, the start and destination locations are shown by solid red and green circular points, respectively. The rectangular obstacles with solid blue circular points represent interest points proposed by our path planning technique. Table II presents the simulation parameters along with the operating system parameters.

A. Setting hyper-parameters

Both, our proposed technique and the APF, each has a hyperparameter: ρ and λ. It is worth noting that the difference in the way they ought to be set.

- Our proposed technique is a hierarchical two-layer control system: the navigation system and the proposed path planning technique. The parameter ρ sets the amount of freedom granted to the navigation system and constrains on the path planning technique. Its value is independent of the location and number of obstacles and targets. It considers the size of the UAV itself, the accuracy of GNSS, the uncertainty caused by environmental disturbances up to a threshold above which the proposed technique should not be used to direct UAVs. Another factor that must be considered is the tracking of errors near turning points as section VI-B has described. It is worth pointing out that these different parameters must be taken together because the corresponding events are generally independent. For the sake of simplicity in the simulations, we used the ρ value used in [43].
- The APF and many similar algorithms optimize contradicting objectives namely, reducing the path length and moving away from obstacles. This trade-off is adjusted by a parameter denoted as λ which is a weight in a cost function that must be minimized. However, the appropriate λ-value can only be set once the locations and number of obstacles and targets are known. Instead of a universal relationship between λ and the minimal distance between the UAV and any obstacle, each relationship is generally a non-decreasing functions.

In our simulation tests, to make fair comparisons, we have conducted simulations for several λ-values and we selected the one which minimizes the path length based on the UAV being able to maintain a ρ-distance from any obstacle at all times.

B. Performance metrics

We used the following performance metrics in our performance evaluation tests:

- Total distance traveled by a UAV from a start point to the endpoint.
- Trajectory tracking error with the variation in UAV's turning angles.
- Trajectory tracking error with the variation in UAV's moving speed.
- Algorithm's computational load by observing the program's execution time.

VI. RESULTS AND DISCUSSIONS

This section presents experimental results for the proposed pathfinding technique along with the artificial potential field method. We compare our proposed scheme with the APF method, which is known to have a precise mathematical model with a low computational load and is suitable for real-time applications [27]. Like most other previously proposed approaches [44], [45], [53], this APF method does not consider a specific uncertainty model. Instead, it uses a free parameter to balance the trade-off between obstacles and reducing the path length. We have used a specific parameter value and considered only the experiments for which the APF technique is successful.

A. Total distance traveled by the UAV

A path planning technique should keep a balance between the length of a planned trajectory and a collision avoidance algorithm. A path planning algorithm looking for the shortest path may cause a collision with an obstacle. In contrast, when a collision avoidance algorithm is used, this will result in longer trajectories. In this context, to evaluate the performance of the proposed scheme, we performed 2600 experiments with different numbers of obstacles varying in sizes, orientations and locations. Fig. 7 illustrates trajectories designed by the compared schemes under different simulation scenarios. The red contour surrounding the desired proposed black line trajectory is the region within which the UAV is expected to take during its flight. It can be inferred from the figure that APF takes a longer curved paths compared to the proposed method that uses pre-defined interest points to get a linear, shorter path to reach the destination.

The experiments concern complex environments with obstacles of varying sizes, locations, and orientations significantly obstructing the direct path. In each experiment, the ending point was chosen to be far enough from any obstacle so that when using the APF technique the field has a unique global minimum. In many experiments the APF technique yields a path leaving the known environment. Among the remaining

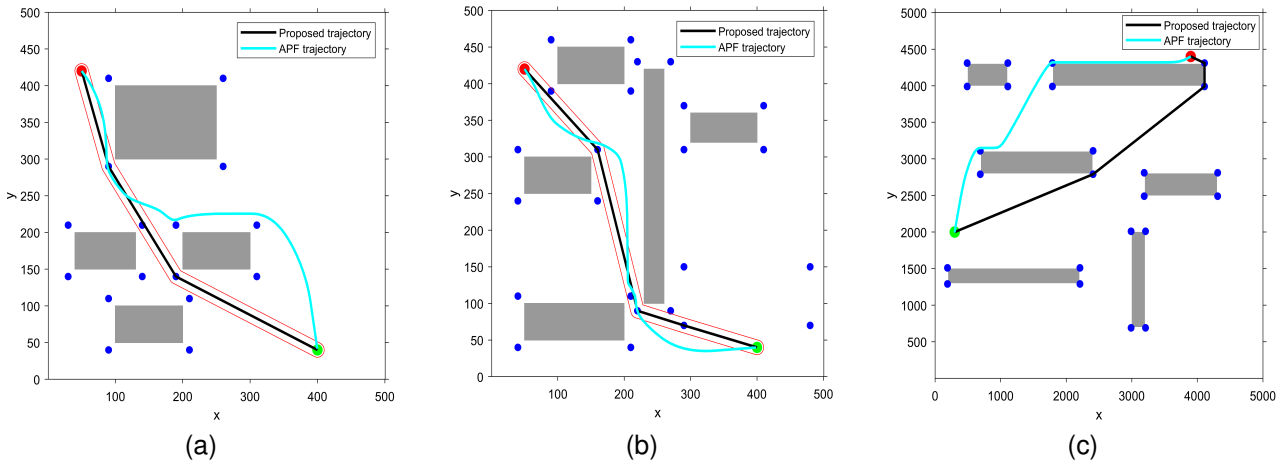

Fig. 7. Trajectories for the schemes compared (a) 4 obstacles, 500 × 500 grid size (b) 5 obstacles, 500 × 500 grid size (c) 6 obstacles, 5000 × 5000 grid size

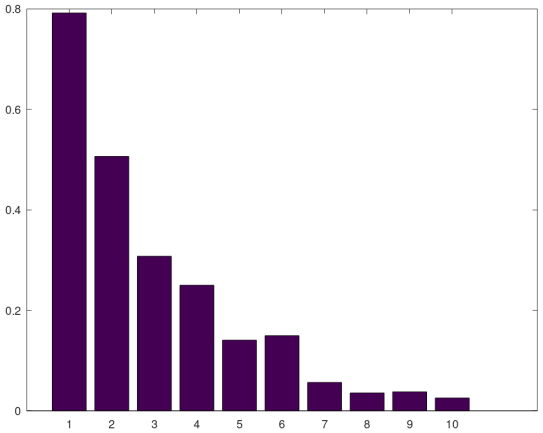

Fig. 8. Measured probability in complex environment for the APF to reach the target.

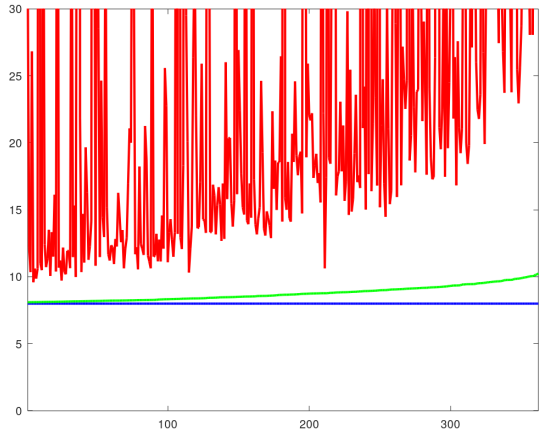

Fig. 9. The flat blue line is the distance between the starting point and the target. The green slowly increasing line is the length of the path yielded by our proposed method. The irregular red curve is the length of the path yielded with the APF method. The experiments are ordered so that the intermediate green line is non-decreasing.

experiments, for some implementation choices and parameter values, we observed numerically instable trajectories unable to find the reaching point. Fig. 8 displays the experimental probabilities for the APF to succeed in finding the reaching point as a function of the number of obstacles. In all these experiments, our method was able to find a target avoiding all obstacles. Fig. 9 concerns the experiments for which the APF implementation succeeded in reaching the target. It shows with a straight horizontal line the Shortest Direct Distance (SDD) and in a green slightly increasing curve the length of the path yielded by our proposed method. The upper irregular red curve shows the lengths of the path yielded with the APF method. For clarity reasons, we ordered the experiments in ascending order of the path length of our proposed method, because we think that it gives an appropriate indication of the complexity of the environment. We see an important reduction in the distance traveled using our method as compared to using APF.

~~This important reduction in the distance traveled implies a corresponding reduction in energy consumption.~~

An important issue is also to compare the energy consumption of the two methods. And many factors contribute significantly to energy loss: navigation (distance travelling at a given speed, manoeuvring a bend and withstanding some wind), computing (memory- and CPU-processing), and communication (downlink and uplink). The following more detailed analysis of this energy consumption shows that our method reduces also the consumed energy.

In [71], 188 quadcopters have been tested across a range of speeds with small packages (0.5kg). It was experimentally found that most of the power is consumed for hovering while ground speed and wind strength impact little the power consumed. A striking consequence is that the use of a higher speed to reach the destination reduces the energy consumption. It has been measured that on

average the consumption is of 80 Joules per meter. To have a rough estimation of the energy consumed while travelling we multiplied the average length of the path by its energy consumption per meter. For the experiments shown in figure 9, we get a consumption of 701.2 Joules for our proposed method and of 3155.2 Joules for the APF method.

In [73] and [72], we can find the power consumption of floating point operations and data movements in 2008 and in 2012 for high-performance computing. Both articles explain that power consumption has now become a dominant constraint as it limits the possibility to increase the clock frequency, and this power consumption is mainly caused by data movements. The power consumption of a floating point operation is below 10^{-10} Joules, that of reading a byte from memory is below 10^{-8} Joules. To have a rough estimation of this energy consumption, we first computed the number of operations Matlab can do in one second on our computer. We then multiplied the average execution time by that number of operations and by the energy consumed when reading four bytes from memory. For the experiments shown in figure 9, we get a consumption of 0.8 Joules for our proposed method and of 2.4 Joules for the APF method.

From a networking viewpoint, step 5 of the intended use case requires as only exchange of information, the reception at the beginning of the locations of A, B, and the locations, orientations, sizes of all obstacles. In [74], we can find an old estimation of the per-byte transmission time (1μs) and the power consumed when receiving messages (0.9W), when using IEEE 802.11b. We derive a rough estimation of the energy consumed when receiving the locations of A, B and the obstacles modeled as rectangles, we first get a number of bytes to transmit: preamble and header of the physical layer (24 bytes in 802.11b), MAC header (38 bytes), IP header (20 bytes) and payload (two times the number of locations and six times the number of obstacles, a total of 124 bytes for experiments with 20 obstacles). And then we multiply that number by the power consumed and the per-byte transmission time. The received messages are the same for both methods and their energy cost is of 2×10^{-4} Joules.

B. Trajectory tracking error versus UAV's turning angles

This subsection demonstrates our claim for designing collision-free trajectories for various UAV turning angles. Fig. 10 shows the scenario we considered which includes five obstacles and the UAV speed is 7 m/s. The dashed arrow in Fig. 10 indicates the interest point around which we considered different turning angles. Fig. 11 shows zoomed-in parts for various UAV turning angles around the vertex represented by the dashed arrow. We observe, from the figure, that the tracked trajectory points remain within the defined limits, i.e., within the red contour. As can be inferred from Fig. 11, the UAV trajectory deviates widely when the UAV turning angle increases in order to stay within its dynamic constraints. Fig. 12 presents the trajectory tracking errors for various UAV's turning angles. This figure also substantiates that the tracking error increases as we move from 0 degrees to 75 degrees. However, it is worth mentioning that the trajectory deviation remains within limits and validates our collision-free trajectory design claim.

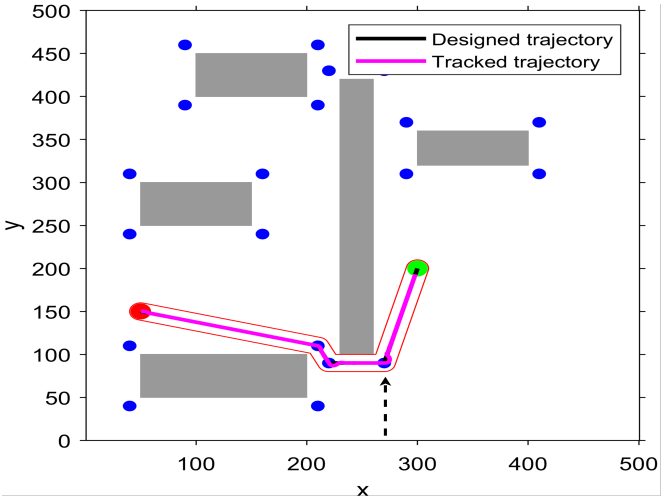

Fig. 10. Simulation scenario with 75° turning angle

C. Trajectory tracking error versus UAV's moving speed

This section analyzes the effect of UAV speed on trajectory tracking. As Fig. 13 shows, we consider a simulation scenario with five obstacles and we vary the UAV speed. Fig. 14 presents the trajectory tracking errors for different UAV speeds. The tracking error for 5 m/s is the lowest among all and increases when the UAV speed approaches 9 m/s because the UAV's control system output stabilizes faster at lower speed values. The tracking errors for the various UAV speed tests stay within the defined contour threshold and validate collision-free trajectory tracking for the proposed scheme.

D. Computational Load Comparison

We estimate the computational load associated with the proposed scheme corresponding to items 4 and 5 described in section III.

To this end, we simulated different scenarios by varying the distance between the start and the destination location and the simulation grid size. Fig. 15 presents the computational load results obtained with the path planning techniques in terms of execution times under varying simulated grid sizes. Both schemes incur comparable load until 6000×6000 grid size, after which APF incurs higher execution time. As Fig. 15 shows, in terms of execution time, the proposed method is not affected by the variation in the grid size. This is because APF considers the entire environment during the path planning process whereas the proposed scheme only considers the interest points. Fig. 16 plots the execution time with respect to the distance between the start and destination locations. As Table III shows, this experiment uses different destination points while keeping the same start position. Fig. 16 plots

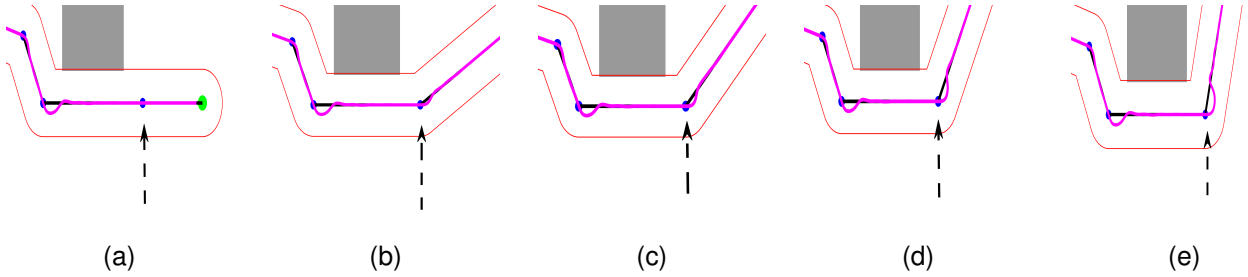

Fig. 11. Zoomed-in plots with arrow pointing toward vertices with angles (a) 0° (b) 30° (c) 45° (d) 60° (e) 75°

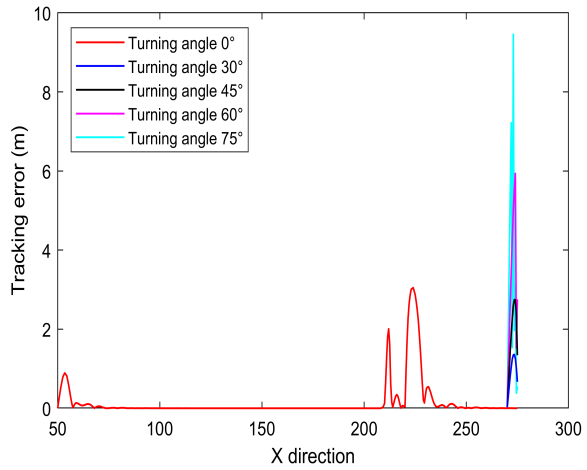

Fig. 12. Tracking error for different turning angles

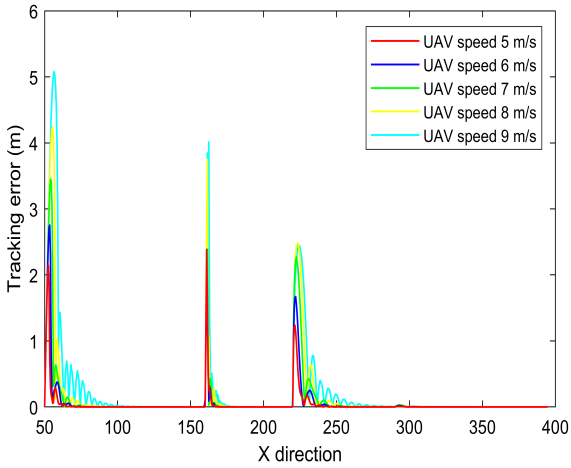

Fig. 14. Tracking error for different UAV speeds

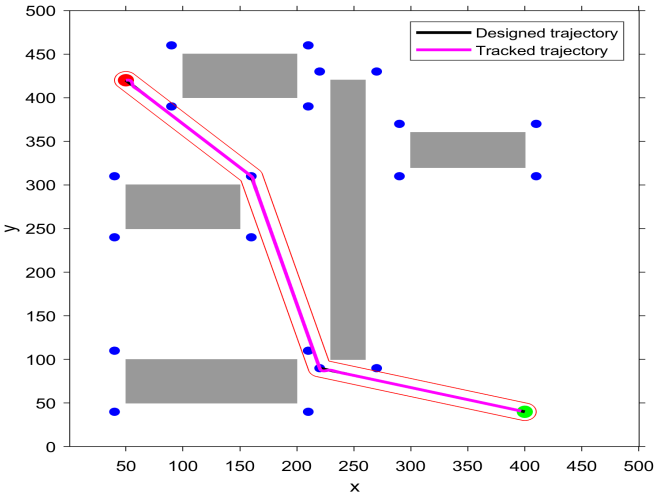

Fig. 13. Scenario considered to determine tracking error under different UAV speeds

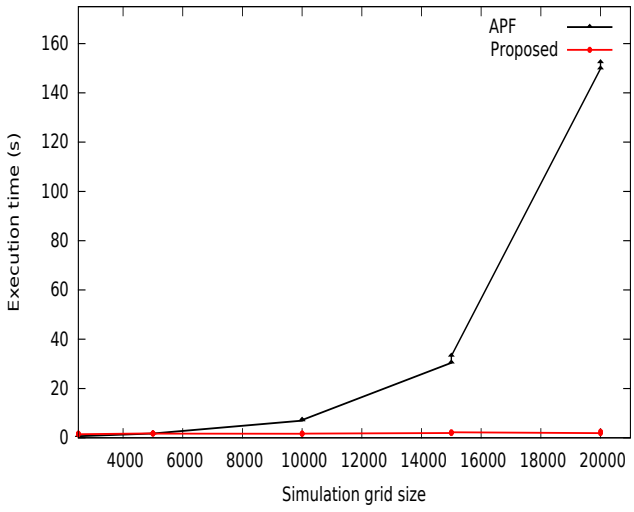

Fig. 15. Impact of simulation grid size on execution time [19 obstacles]

the average results for each start and destination combination. The proposed scheme has an execution time with a mean value of 2.22 and a standard deviation of 0.07. APF, on the other hand, has a mean value of 163.96 and a standard deviation of 4.67. From the figure, we observe that although both schemes are unaffected by the variation in the distance, but it is worth mentioning here that our proposed method outperforms APF in terms of execution time. There could be various reasons that explain the high execution time of APF: high memory consumption, implementation choices assuming that there are few moving obstacles of small size, and trade-off when selecting the parameter values between performance and numerical instabilities requiring a larger grid.

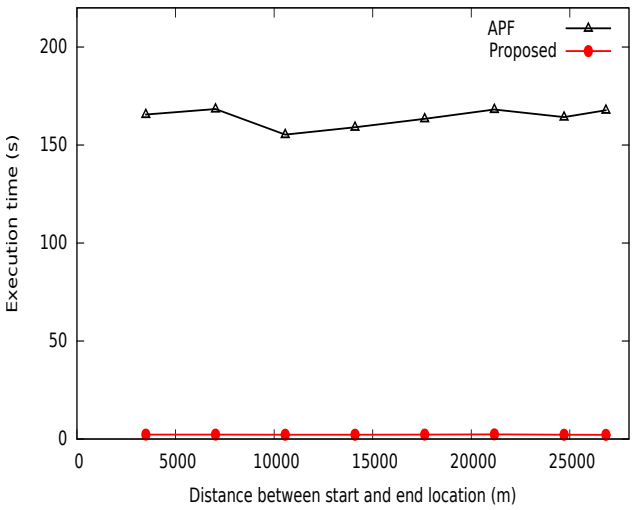

Fig. 16. Impact of the distance between the start and destination location on execution time [grid size 20000 × 20000, 19 obstacles]

TABLE III
DESTINATION POINTS AND DISTANCE BETWEEN START AND DESTINATION POINTS (START POINT [10 , 50])

Destination point	Distance between start and destination point (m)
[2500 , 2500]	3493
[5000 , 5000]	7028
[7500 , 7500]	10564
[10000 , 10000]	14100
[12500 , 12500]	17635
[15000 , 15000]	21171
[17500 , 17500]	24706
[19000 , 19000]	26828

VII. Conclusion

A collision-free path planning is of paramount importance to ensure the safety of UAVs and humans on the ground. Our proposed UAV path planning technique works by taking advantage of interest points defined around the rectangular obstacles. We used the disk-based uncertainty model to eliminate the chances of a collision by considering the environmental disturbances and the tracking errors that may appear at turning points. This uncertainty model which is based on a threshold distance provides an extra layer of safety to the underlying control system, i.e., to act as a guard assuming the control system does not exceed its defined overshoot limits. Moreover, due to the low computational demand regardless of the deployed environment dimensions, our approach becomes promising for real-time path planning.

Regarding our future work, one idea is to replace the Dijkstra algorithm used in our proposed technique with some other graph algorithm such as A* in order to evaluate path planning computational complexity under different scenarios. Simulations have shown that given a turning point of a given angle, increased speed induces higher tracking errors yielding the necessity to use a higher value for ρ which constrains the selection of the path. To reduce these constraints, we are considering the possibility of assigning a reduced speed in the neighborhood of turning points, a first step towards integrating UAV kinematics into the path planning method. This work is also applicable to scenarios where we have multiple destination locations, with each location as a new interest point. In our future work, we plan to develop a UAV-based network comprising roadside units to provide new interest points at run-time. Our proposed technique cannot be directly applied to unconstrained moving objects. Nonetheless we propose in future work to adapt our technique to obstacles and targets whose complete motion is known prior to departure, replacing the 2D geometry with a space-time geometry: moving obstacles are represented by oblique prisms, interest points and vertices by line segments, edges by polygons, and weights by time-dependent functions.

Acknowledgments

We thank the anonymous reviewers for their valuable comments which helped us improve the quality and presentation of this paper.

Appendix A
Sketch of Proof of Theorem 1

Let us first prove theorem 1 for a specific rectangle centered and of size 2×2 denoted $\mathcal{R}^*$ with $M(t)$ a continuous mapping from $[0,T]$ to $\mathbb{R}^2$ such that $\mathrm{d}\left(M(0),\mathcal{R}\right)>0$ and for $t\in[0,T]$, $\mathrm{d}\left(M(t),\mathcal{R}\right)\geq\rho$. We define $t\mapsto f(t)$ on $[0,T]$ as

$$\begin{cases} f(t)=\mathrm{d}(M(t),\mathcal{R}^*) & \text{if} \quad \mathrm{d}(M(t),\mathcal{R}^*)>0 \\ f(t)=-\mathrm{d}(M(t),\partial\mathcal{R}^*) & \text{if} \quad \mathrm{d}(M(t),\mathcal{R}^*)<\mathrm{d}(M(t),\partial\mathcal{R}^*) \\ f(t)=0 & \text{if} \quad \mathrm{d}(M(t),\partial\mathcal{R}^*)=0 \end{cases}$$

It is the following analytic expression of both distances which shows that f is well defined, continuous and that the first condition happens when $M(t)$ is outside the rectangle, the second condition when $M(t)$ is inside the rectangle and the third condition when $M(t)$ is on its border.

$$\begin{cases} \xi(x)=(|\,|x|-1|)\,\mathbf{1}_{\mathbb{R}\setminus[0,1]}(x) \qquad \eta(x)=(|\,|x|-1|)\,\mathbf{1}_{[0,1]}(x) \\ \mathrm{d}\left(M(t),\mathcal{R}^*\right)=\sqrt{\xi(x)^2+\xi(y)^2} \\ \mathrm{d}\left(M(t),\partial\mathcal{R}^*\right)=\sqrt{\xi(x)^2+\xi(y)^2}+\min\left(\eta(x),\eta(y)\right) \end{cases}$$

where x and y are the coordinates of M and $\mathbf{1}_{\mathcal{S}}(x)=0$ if $x\notin\mathcal{S}$ and $\mathbf{1}_{\mathcal{S}}(x)=1$ if $x\in\mathcal{S}$, $\mathcal{S}$ being any set. Based on the first assumption, we have $f(0)>0$ and if the implication we are proving was false, then there would exist $t\in[0,T]$ such that $f(t)\leq 0$ which would imply the existence of $t'\in[0,T]$ with $f(t')=0$ meaning that $\mathrm{d}(M(t),\partial\mathcal{R}^*)=0$ and contradicting the second assumption. We extend this proof to any rectangle $\mathcal{R}$ by considering a transformation $\mathcal{T}$ that is a combination of translation, rotation and rescaling, whose inverse is denoted $\mathcal{T}^{-1}$. It fulfills the following conditions with $\tau>0$.

$$\begin{cases} M\in\mathcal{R}\;\Leftrightarrow\;\mathcal{T}(M)\in\mathcal{R}^* \\ M\in\partial\mathcal{R}\;\Leftrightarrow\;\mathcal{T}(M)\in\partial\mathcal{R}^* \\ \mathrm{d}(\mathcal{T}(M),\mathcal{T}(M'))=\tau\,\mathrm{d}(M,M') \\ M(t)\text{continuous}\;\Leftrightarrow\;\mathcal{T}(M(t))\text{continuous} \end{cases}$$

These conditions prove that

$$\mathrm{d}(M,\mathcal{R})=\min_{M'\in\mathcal{R}}\mathrm{d}(M,M')=\min_{\mathcal{T}(M')\in\mathcal{R}^*}\mathrm{d}(M,M')=\frac{1}{\tau}\min_{\mathcal{T}(M')\in\mathcal{R}^*}\mathrm{d}(\mathcal{T}(M),\mathcal{T}(M'))=\frac{1}{\tau}\,\mathrm{d}(\mathcal{T}(M),\mathcal{R}^*)$$

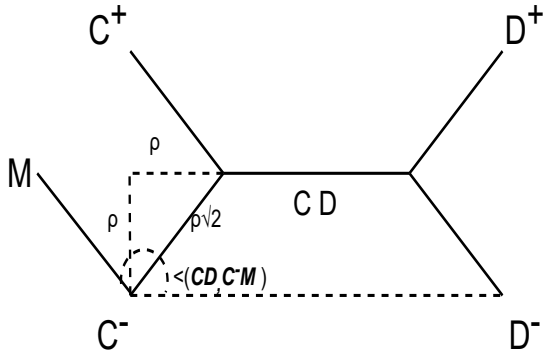

Fig. 17. Graphical construction illustrating theorem 3. As $\angle(CD, C^-M) = \frac{3\pi}{4}$, M is on the left of a half-plane delineated by line (C^-C^+) located at a distance of ρ of the obstacle CD.

and in the same way that $\mathrm{d}(M, \partial\mathcal{R}) = \frac{1}{\tau}\,\mathrm{d}(\mathcal{T}(M), \partial\mathcal{R}^*)$. Hence the theorem proven for $\mathcal{R}^*$ is also true for $\mathcal{R}$.

APPENDIX B
SKETCH OF PROOF OF THEOREM 2

The proof is based on these four statements.

- The weighted partial sums of a given path are time stamps of the corresponding mapping $V \in \mathcal{V}'$.

$$t_n = \frac{1}{v}\sum_{k=2}^{n} \mathrm{d}(P_{j_{k-1}}, P_{j_k}) \tag{15}$$

- The weight of a path is the time of flight of a mapping in $V \in \mathcal{V}'$.

$$T_{\mathcal{V}}(V) = \frac{1}{v}\sum_{k=2}^{N} \mathrm{d}(P_{j_{k-1}}, P_{j_k}) \tag{16}$$

- Each traversed edge is a line segment of the trajectory traveled at speed v.

$$n \in \{2, \ldots, N\},\ t \in [t_{n-1}, t_n] \ \Rightarrow\ \overrightarrow{P_{j_{n-1}}V(t)} = \frac{t - t_{n-1}}{t_n - t_{n-1}}\overrightarrow{P_{j_{n-1}}P_{j_n}} \tag{17}$$

- Based on equation (9), edges traversed by the path define line segments in the trajectory that remain at a safe distance from any obstacle.

$$\forall n \in \{2, \ldots, N\}, \quad (P_{j_{n-1}}, P_{j_n}) \in \mathcal{E} \ \Leftrightarrow\ \forall i \in \mathcal{I},\ \mathrm{d}(V(t), \overline{C_iD_i}) \geq \rho \tag{18}$$

APPENDIX C
SKETCH OF PROOF OF THEOREM 3

As the four statements are similar, we exhibit only the proof of the first statement and assume that M is located such that $\theta = \angle(\overrightarrow{CD}, \overrightarrow{C^-M}) \in [\frac{\pi}{2}, 2\pi]$. Figure 17 illustrates the two following arguments.

- If $\theta \in [\frac{\pi}{2}, \frac{3\pi}{2}]$, then M is located in the left half-plane delineated by line (C^-C^+) which is located at a distance of ρ from the obstacle $\overline{CD}$. As this property is shared too by C^- and using convexity, we get that $\mathrm{d}(\overline{CD}, \overline{C^-M}) \geq \rho$.
- If $\theta \in [\frac{3\pi}{2}, 2\pi]$, then M is located in the left half-plane delineated by line (C^-D^-) which is located at a distance of ρ from the obstacle $\overline{CD}$. As this property is shared too by C^- and using convexity, we get also that $\mathrm{d}(\overline{CD}, \overline{C^-M}) \geq \rho$.

REFERENCES

[1] Oubbati. O. S, Lakas. A, Lorenz. P, Atiquzzaman. M, Jamalipour. A. 2019. "Leveraging communicating UAVs for emergency vehicle guidance in urban areas," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, pp. 1070–1082.

[2] D. Yang, Q. Wu, Y. Zeng, R. Zhang. 2018. "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 6721–6726.

[3] Y. Wan, J. Yan, Z. Lin, V. Sheth, S. K. Das. 2018. "On the structural perspective of computational effectiveness for quantized consensus in layered UAV networks," *IEEE Transactions on Control of Network Systems*, vol. 6, pp. 276–288.

[4] S. Waharte, N. Trigoni. 2010, September. "Supporting search and rescue operations with UAVs," In *Proceedings of International Conference on Emerging Security Technologies*, Canterbury, UK, pp. 142–147.

[5] N. Bashir, S. Boudjit, M. Y. Saidi. 2021, January. "A Distributed Anticipatory Life-Enhancing Recovery Approach for Unmanned Aerial Vehicular Networks," In *Proceedings of IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, pp. 1–7.

[6] "How drones can be used to combat COVID-19 ," Accessed: Jul. 22, 2021, Available: https://www.unicef.org/supply/documents/how-drones-can-be-used-combat-covid-19.

[7] "Innovating to Fight COVID-19: Four Ways Drones are Contributing," Accessed: Apr. 15, 2020, Available: https://enterprise.dji.com/fr/news/detail/fight-covid-19-with-drones.

[8] M. Radmanesh, M. Kumar, P. H. Guentert, M. Sarim. 2018. "Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study," *Unmanned systems*, vol. 6, pp. 95–118.

[9] I. K. Ha, Y. Z. Cho. 2018. "A probabilistic target search algorithm based on hierarchical collaboration for improving rapidity of drones," *Multidisciplinary Digital Publishing Institute*, vol. 18, p. 2535.

[10] R. D. Arnold, H. Yamaguchi, T. Tanaka. 2018. "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *Journal of International Humanitarian Action*, vol. 3, pp. 1–18.

[11] L. Ruetten, P. A. Regis, D. Feil-Seifer, S. Sengupta. 2020, January. "Area-optimized UAV swarm network for search and rescue operations," In *Proceedings of 10th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, pp. 613–618.

[12] B. Zhu, E. Bedeer, R. Barton. 2021. "UAV Trajectory Planning in Wireless Sensor Networks for Energy Consumption Minimization by Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no 9, p.9540–9554.

[13] O. Oubbati, A. Lakas M. Guizani. 2022. "Multi-Agent Deep Reinforcement Learning for Wireless-Powered UAV Networks," *IEEE Internet Of Things Journal*, 16 pages. doi: 10.1109/JIOT.2022.3150616.

[14] H. Chang, Y. Chen, B. Zhang, D. Doermann. 2022. "Multi-UAV Mobile Edge Computing and Path Planning Platform based on Reinforcement Learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no 3, p. 489–498.

[15] A. E. Holton, S. Lawson, C. Love. 2015. "Unmanned Aerial Vehicles: Opportunities, barriers, and the future of "drone journalism," *Journalism practice*, vol. 9, pp. 634–650.

[16] M. Naazare, D. Ramos, J. Wildt, D. Schulz. 2019, September. "Application of graph-based path planning for UAVs to avoid restricted areas," In *Proceedings of IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Würzburg, Germany, pp. 139–144.

[17] R. L. Galvez, G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, E. P. Dadios, A. A. Bandala, R. R. P. Vicerra, A. H. Fernando. 2017, November. "Obstacle avoidance algorithm for swarm of quadrotor unmanned aerial vehicle using artificial potential fields," In *Proceedings of TENCON 2017 IEEE Region 10 Conference*, Penang, Malaysia, pp. 2307–2312.

[18] S. A. Bortoff. 2000, June. "Path planning for UAVs," In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, Chicago, IL, USA, pp. 364–368.

[19] H. A. Satai, M. M. A. Zahra, Z. I. Rasool, R. S. Abd-Ali, C. I. Pruncu. 2021. "Bézier curves-based optimal trajectory design for multirotor UAVs with any-angle pathfinding algorithms," *Sensors*, vol. 21, p. 2460.

[20] V. San Juan, M. Santos, J. M. Andújar, 2018. "Intelligent UAV map generation and discrete path planning for search and rescue operations," *Complexity*, vol. 2018, p. 17.

[21] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, England, 1989.

[22] R. S. Sutton and A. G. Barto. *Reinforcement Learning An Introduction*. The MIT Press, Cambridge Massachusetts London England, second edition edition, 2018.

[23] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath. 2017. "A brief Survey of Deep Reinforcement Learning," *IEEE Signal Processing Magazine*, vol. 34, 16 pages.

[24] O. Oubbati, M. Atiquzzaman, A. Baz, H. Alhakami, J. Ben-Othman. 2021. "Dispatch of UAVs for Urban Vehicular Networks: A Deep Reinforcement Learning Approach," *IEEE Transactions On Vehicular Technology*, vol. 70, no 12, pp. 13174–13189.

[25] D. P. Bertsekas. *Dynamic Programming and Optimal Control Volume I*. Athena Scientific, Belmont, Massachusetts Institute of Technology, third edition edition, 2005.

[26] C. Goerzen, Z. Kong, B. Mettler. 2010. "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 65–100.

[27] G. Zhang, L.T. Hsu. 2019. "A new path planning algorithm using a GNSS localization error map for UAVs in an urban area," *Journal of Intelligent & Robotic Systems*, vol. 94, pp. 219–235.

[28] S. Waharte, N. Trigoni, S. Julier. 2009, June. "Coordinated search with a swarm of UAVs," In *Proceedings of 6th IEEE annual communications society conference on sensor, mesh and ad hoc communications and networks workshops*, Rome, Italy, pp. 1–3.

[29] M. Kothari, I. Postlethwaite. 2013. "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, pp. 231–253.

[30] J. Redding , J. Amin, J. Boskovic, Y. Kang, K. Hedrick, J. Howlett, S. Poll. 2007, August. "A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters," In *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, p. 6413.

[31] S. Siebert, J. Teizer. 2014. "Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system," *Automation in construction*, vol. 41, pp. 1–14.

[32] Z. Yu, K. Nonami, J. Shin, D. Celestino. 2007. "3D vision based landing control of a small scale autonomous helicopter," *International Journal of Advanced Robotic Systems*, vol. 4, p. 7.

[33] G. Zhang, L. T. Hsu. 2018. "Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system," *Aerospace science and technology*, vol. 80, pp. 368–380.

[34] Y. Zhao, Z. Zheng, Y. Liu. 2018. "Survey on computational-intelligence-based UAV path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64.

[35] L. De Filippis, G. Guglieri, F. Quagliotti. 2012. "Path planning strategies for UAVS in 3D environments," *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 247–264.

[36] S. Goudarzi, N. Kama, M. H. Anisi, S. Zeadally, S. Mumtaz. 2019. "Data collection using unmanned aerial vehicles for internet of things platforms," *Computers & Electrical Engineering*, vol. 75, pp. 1–15.

[37] P. Vadakkepat, K. C. Tan, W. Ming-Liang. 2000, July. "Evolutionary artificial potential fields and their application in real time robot path planning," In *Proceedings of the 2000 congress on evolutionary computation*, La Jolla, CA, USA, pp. 256–263.

[38] O. Khatib. 1986. "Real-time obstacle avoidance for manipulators and mobile robots," *Autonomous robot vehicles*, Springer, pp. 396–404.

[39] L. Lifen, S. Ruoxin, L. Shuandao, W. Jiang. 2016, August. "Path planning for UAVS based on improved artificial potential field method through changing the repulsive potential function," In *Proceedings of Chinese Guidance, Navigation and Control Conference*, Nanjing, China, pp. 2011–2015.

[40] X. Chen, J. Zhang. 2013, August. "The three-dimension path planning of UAV based on improved artificial potential field in dynamic environment," In *Proceedings of 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China , pp. 144–147.

[41] Y. Yao, X. S. Zhou, K. L. Zhang, D. Dong. 2010. "Dynamic trajectory planning for unmanned aerial vehicle based on sparse A* search and improved artificial potential field," *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, vol. 27, pp. 953–959.

[42] Z. Yingkun. 2018, June. "Flight path planning of agriculture UAV based on improved artificial potential field method," In *Proceedings of Chinese Control And Decision Conference*, Shenyang, China , pp. 1526–1530.

[43] E. Oland, R. Kristiansen. 2013, March. "Collision and terrain avoidance for UAVs using the potential field method," In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1–7.

[44] L. Lifen, S. Ruoxin, L. Shuandao, W. Jiang. 2016, August. "Path planning for UAVS based on improved artificial potential field method through changing the repulsive potential function," In *Proceedings of IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, pp. 2011–2015.

[45] Y. B. Chen, G. C. Luo, Y. S. Mei, J. Q. Yu, X. L. Su. 2016. "UAV path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, pp. 1407–1420.

[46] C. T. Recchiuto, A. Sgorbissa. 2018. "Post-disaster assessment with unmanned aerial vehicles: A survey on practical implementations and research approaches," *Journal of Field Robotics*, vol. 35, pp. 459–490.

[47] F. Yan, Y. S. Liu, J. Z. Xiao. 2013. "Path planning in complex 3D environments using a probabilistic roadmap method," *International Journal of Automation and computing*, vol. 10, pp. 525–533.

[48] M. Kothari, I. Postlethwaite, D. W. Gu. 2010. "A Suboptimal Path Planning Algorithm Using Rapidly-exploring Random Trees," *International Journal of Aerospace Innovations*, vol. 2.

[49] M. W. Achtelik, S. Weiss, M. Chli, R. Siegwart. 2013, May. "Path planning for motion dependent state estimation on micro aerial vehicles," In *Proceedings of IEEE international conference on robotics and automation*, Karlsruhe, Germany, pp. 3926–3932.

[50] M. Kothari, I. Postlethwaite, D. W. Gu. 2009, December. "Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees," In *Proceedings of IEEE on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Shanghai, China, pp. 3069–3074.

[51] Sun, Q., Li, M., Wang, T. and Zhao, C., 2018, June. 'UAV path planning based on improved rapidly-exploring random tree," In *Proceedings of IEEE on Chinese Control and Decision Conference*, Shenyang, China, pp. 6420–6424.

[52] E. W. Dijkstra. 1959. "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, pp. 269–271.

[53] P. Maini, P. B. Sujit. 2016, June. "Path planning for a UAV with kinematic constraints in the presence of polygonal obstacles," In *Proceedings of IEEE international conference on unmanned aircraft systems (ICUAS)*, Arlington, VA, USA, pp. 62–67.

[54] F. L. L. Medeiros, J. D. S. Da Silva. 2010, October. "A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation," In *Brazilian Symposium on Artificial Intelligence*, Berlin, Heidelberg, pp. 213–222.

[55] Y. Qu, Y. Zhang, Y. Zhang. 2018. "A global path planning algorithm for fixed-wing UAVs," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 691–707.

[56] X. Chen, X. Chen. 2014, May. "The UAV dynamic path planning algorithm research based on Voronoi diagram," In *26th IEEE chinese control and decision conference*, Changsha, China, pp. 1069–1071.

[57] P. E. Hart, N. J. Nilsson, B. Raphael. 1968. "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107.

[58] K. Yang, S. Sukkarieh. 2008, September. "3D smooth path planning for a UAV in cluttered natural environments," In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 794–800.

[59] W. Chen, X. Wu, Y. Lu. 2015. "An improved path planning method based on artificial potential field for a mobile robot," *Cybernetics and information technologies*, vol. 15, pp. 181–191.

[60] M. De Berg, M. Van Kreveld, M. Overmars, O. C. Schwarzkopf. 2000. "Visibility graphs," *Computational geometry*, pp. 307–317.

[61] H. F. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM) : Part I the essential algorithms. volume 13, pages 99–110, 2006.

[62] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun.

[63] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, University of Illinois, 2018.
Principles of Robot Motion-Theory, Algorithms, and Implementation. The MIT Press, Cambridge, Massachusetts, 2005.

[64] M. Doole, J. Ellerbroek, V. L. Knoop, J. M. Hoekstra. 2021. "Constrained Urban Airspace Design for Large-Scale Drone-Based Delivery Traffic," *Aerospace*, vol. 15, p. 38.

[65] Yaoming ZHOU, Yu SU, Anhuan XIE, and Lingyu KONG. A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of uav. *Chinese Journal of Aeronautics*, 34(9):199–209, 2021.

[66] Yu-Hsin Hsu and Rung-Hung Gau. Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks. *IEEE Transactions on Mobile Computing*, 21(1):306–320, 2022.

[67] Moataz Samir, Dariush Ebrahimi, Chadi Assi, Sanaa Sharafeddine, and Ali Ghrayeb. Trajectory planning of multiple dronecells in vehicular networks: A reinforcement learning approach. *IEEE Networking Letters*, 2(1):14–18, 2020.

[68] Moataz Samir, Dariush Ebrahimi, Chadi Assi, Sanaa Sharafeddine, and Ali Ghrayeb. Leveraging uavs for coverage in cell-free vehicular networks: A deep reinforcement learning approach. *IEEE Transactions on Mobile Computing*, 20(9):2835–2847, 2021.

[69] Wei Jiang, Yongxi Lyu, Yongfeng Li, Yicong Guo, and Weiguo Zhang. UAV path planning and collision avoidance in 3d environments based on pompd and improved grey wolf optimizer. *Aerospace Science and Technology*, 121:107314, 2022.

[70] Guan-Ting Tu and Jih-Gau Juang. Uav path planning and obstacle avoidance based on reinforcement learning in 3d environments. *Actuators*, 12(2), 2023.

[71] Thiago A. Rodrigues, Jay Patrikar, Natalia L. Oliveira, H. Scott Matthews, Sebastian Scherer, and Constantine Samaras. Drone flight data reveal energy and greenhouse gas emissions savings for very small package delivery. *Patterns*, 3(100569):1–11, 2022.

[72] Robert Leland, Richard Murphy, Bruce Hendrickson, Katherine Yelick, John Johnson, and Jonathan Berry. Large-scale data analytics and its relationship to simulation. Technical report, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, March 2014.

[73] Peter Kogge and John Shalf. Exascale computing trends: Adjusting to the "new normal"' for computer architecture. *Computing in Science & Engineering*, 15(6):16–26, 2013.

[74] Daniel de O. Cunha, Luís Henrique M. K. Costa, and Otto Carlos M. B. Duarte. Analyzing the energy consumption of IEEE 802.11 ad hoc networks. In Elizabeth M. Belding-Royer, Khaldoun Al Agha, and Guy Pujolle, editors, *Mobile and Wireless Communication Networks*, pages 473–484, Boston, MA, 2005. Springer US.

Nouman Bashir received his M.Sc. degree in Electronic Engineering from GIK Institute of Engineering Sciences and Technology, Pakistan, in 2017. He is currently pursuing his Ph.D. at the L2TI laboratory of Université Sorbonne Paris Nord. From 2015 to 2017, he was with the Faculty of Electrical Engineering at GIKI institute, where he served as a Graduate Assistant. During the same period, he was also a member of the TeleCoN Research Lab at GIK Institute. His research interests include Unmanned Aerial Vehicular(UAV) networks, wireless sensor networks, protocols and architecture design in mobile ad hoc networks.

Saadi Boudjit received his Ph.D degree in computer science from the National Institute for Research in Computer Science and Control (INRIA Paris) in 2006 and was a research fellow with Telecom Paris Tech (2006-2007). He is currently Associate Professor and member of the L2TI laboratory at the university of Paris 13 (since 2007). Dr. Boudjit is the initiator and chair of ACM MobileHealth workshop which aims at providing a forum for the interaction of multiple areas related to pervasive wireless healthcare systems. He also acted and still acts asTPC member of several IFIP, ACM and IEEE conferences and workshops (HealthCom, MobileHealth, ICC, Globecom, CAMAD, WCNC, WONS, Wireless Days, GIIS, ...). His research interests include wireless networks, parallel and distributed computing, protocols and architecture design in mobile ad hoc networks, wireless sensor networks, vehicular adhoc networks, and protocols and architecture design for eHealth systems.

Gabriel Dauphin received the engineer's degree from Mines Paris Tech, Paris, France, in 1996, and the Ph.D. degree in signal and image processing from Télécom Paris Tech University, Paris, in 2001.Since 2002, he has been an Assistant Professor with the Laboratory of Information Processing and Transmission (L2TI), University Paris 13, Villetaneuse,France. His research interests include gaterecognition, stereoscopic image compression, machinelearning, and remote sensing.

Sherali Zeadally earned his bachelor's degree in computer science from the University of Cambridge, England. He also received a doctoral degree in computer science from the University of Buckingham, England, followed by postdoctoral research at the University of Southern California, Los Angeles, CA. He is a Professor in the College of Communication and Information at the University of Kentucky. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England.