A Connectivity Aware Path Planning for a fleet of UAVs in an Urban Environment

Nouman Bashir, Saadi Boudjit, and Gabriel Dauphin

***Abstract*—Unmanned Aerial Vehicles (UAVs) are known for their highly dynamic nature, as a result of which their applications demand high design consideration in urban areas. It is imperative to have trajectories that avoid UAV-to-UAV and UAV-to-obstacle collision to ensure the safety of a fleet and people on the ground. Moreover, many applications, like temporary network provision, require continuous backhaul fleet connectivity. This work simultaneously addresses UAVs' path planning and routing issues to propose connectivity-aware path planning for a fleet of UAVs in an urban environment. The proposed scheme is a graph-based offline path planning with fleet line formation that ensures continuous backhaul connectivity. This feature allows any UAV to play the role of leader and guide the entire fleet according to a desired speed. Thanks to the continuous backhaul connectivity, the Base Station (BS) can disseminate commands to the connected fleet as required. Fleet line formation acts as a backbone network and allows additional UAVs or ground users to become a part of this network. The proposed approach is implemented in MATLAB's UAV Toolbox and evaluated in a network simulator. The simulation results demonstrate that the proposed scheme provides collision-free trajectories while ensuring continuous BS connectivity.**

***Index Terms*—UAVs, backhaul connectivity, path planning, UAVs routing, UAVs applications, obstacle avoidance.**

I. Introduction

Maneuvering capabilities, autonomous behaviors, and communication potential of Unmanned Aerial Vehicles (UAVs) are key enablers for intelligent transportation systems. The technology of UAVs is continuously evolving and resulting in more robust and cost-effective solutions [1]. We are here concerned with rotorcraft UAVs and these have been involved recently in a plethora of applications such as surveillance [2], [3], disaster management [4]–[6], search and rescue operations [7], [8], temporary network connectivity [9]–[12], and many other applications [13]. In such applications, specifically for an urban environment, it becomes imperative to design collision-free trajectories while ensuring reliable end-to-end link connectivity to the Base Station (BS). Such path planning techniques are called connectivity-aware coverage [14].

A single UAV with limited power resources and communication range fails to perform in applications where rapidity or real-time aspect is required [15]. A fleet of UAVs has proven to be more scalable and robust than single UAV-based systems, specifically in missions having complex tasks [14], [16]. Deployment of small and lightweight UAVs reduces cost and improves mobility by allowing passage from confined areas [17].

The formation of a UAV fleet is a challenging task due to the unique dynamics associated with UAVs [18]. Path planning and data routing for a fleet of UAVs are influenced by these characteristics and demand careful consideration. Having a fleet of UAVs in a mission increases the chances of collisions [19], [20], moreover high relative mobility results in frequent link failures [21]. Various formation control architectures are available in the literature, such as leader-follower, behavioral-based and virtual structure [22]. The leader-follower approach is the popular one wherein a leader UAV moves along a predefined trajectory, while other follower UAVs follow the leader and keep a safe relative distance between the leader and other UAVs [23]. In the virtual structure approach, UAVs maintain a reference formation shape by keeping a rigid geometric relation with other each other. Each UAV in this method minimizes the error between the defined virtual position and actual position to maintain a reference shape [24]. In [36] the target is a line formation. The behavior-based technique comes up with a predefined behavior for each UAV using a hybrid vector-weighted formation control function [25].

To control the formation of UAVs, a continuous exchange of information among UAVs, such as the current locations and velocities, is necessary [26], [27]. Considering the dynamic nature of UAVs, the absence of state information due to communication problems may lead to severe consequences for the UAVs and people on the ground [28]. On the other hand in the context of low bandwidth capacities, a UAV formation control demanding a high communication load leaves reduced communication resources for the remaining tasks. This could jeopardize its use in applications requiring high Quality-of-Service (QoS) such as Search and Rescue Operation during a natural disaster. Collision-free path planning is of paramount importance to introduce autonomous operating capability into a fleet of UAVs [29]. In terms of time-domain, UAV path planning methods fall into online and offline categories [30]. In the first approach, UAVs plan their path in real-time and react to environmental changes, while in the second approach, UAVs are given a path plan before the commencement of a mission. Online methods demand the availability of high computing resources at each UAV or at least at leader UAV followed by dissemination of the designed path to other UAVs. Offline methods, on the other hand, fail to tackle any unforeseen scenarios.

Designing routing protocols for a dynamic and resource-constrained UAV network poses serious challenges. Path planning and formation control should go hand in hand as conven-

N. Bashir is with NXP Semiconductors, Caen 14118, France.
S. Boudjit and G. Dauphin are with L2TI, Institut Galilée, Université Sorbonne Paris Nord, Villetaneuse 93430, France.
E-mail: boudjit@univ-paris13.fr

tional ad hoc routing protocols are not capable enough to cope with such a rapidly changing topology of UAV networks. A UAV network may have different connectivity requirements based on a specific application. These requirements include always connected, periodic, and delay-tolerant connectivity. In an always-connected network, all UAVs have BS connectivity all the time. In a periodic approach, UAVs get an opportunity to exchange information with each other at relay points. Delay-tolerant UAV networks are devoid of continuous connectivity, and the exchange of information becomes possible only in the proximity of BS. Having just fleet connectivity is sufficient to establish cooperative relationships among UAVs, but having no connection with BS results in the lack of real-time aspect.

This paper simultaneously addresses path planning and routing issues for a fleet of UAVs to provide collision-free trajectories, continuous monitoring, and end-to-end link continuity with BS.

We propose a four layer formation control of UAVs. The organisation in layers, of the proposed control strategy, bears some resemblance with the multiple closed loops having disparate time-constant dynamics, as mentioned in [39]. The first layer, inside each UAV, receives from the second layer the position of the virtual moving target. This first layer is also called the navigation's system, and given some decent environmental conditions, it ensures that the UAV remains at a safe distance from its virtual moving target. It should take advantage of knowing ahead of time the position of the virtual target and hence adapt smoothly to sharp bends. It makes sure that it remains at all times within a safe distance from this target. An upper bound of that distance is denoted here ρ_c. The second layer inside each UAV receives from the third layer the path and the speed of the formation, it computes the position of the virtual moving target. This second layer sends back to the third layer, the information regarding the intended application and possibly the presence of unknown obstacles. The third layer inside the ground controller receives form the fourth layer the starting and finishing points, the set of obstacles and the speed of the formation. The fourth layer in the ground controller depends specifically on the application considered. It collects the information transmitted by the UAVs, defines and modifies the objectives. It may give different missions to different groups of UAVs and induce some groups of UAVs to see as obstacles the path used by other UAVs.

In this paper we give a short description of the fourth layer to show how our proposal could be used in some applications. We do not describe the first layer, while its design is straightforward when ρ_c is set loose, it remains a challenging research issue when set tightly.

We describe precisely the second and third layer which include a graph-based path planning method with obstacles modeled as line segments. Four points defined around each obstruction represent nodes of the graph that help in going around those obstructions. Traceable non-colliding edges which do not yield sharp turns are defined afterward. The proposed technique iteratively applies the Dijkstra algorithm and removes any non-compliant edges until it finds a valid path.

Our proposal is a centralized off-line formation control, where UAVs act as a virtual structure moving along a path, possibly bending itself. Nonetheless we think that control strategy could be used in real time and adapt to moving obstacles not well known. While the ground control is expected to know the main obstacles, it may collect supplementary information from each UAV. It may regularly update the path by sending messages containing the new points and the time at which this new path comes into effect. Besides each UAV may adapt locally to small obstacles, to the extent that it remains within a distance ρ_c of the virtual moving target. Actually it may prove to be more convenient for real time applications, because of the reduced needs for UAVs, to, communicate, keep a close watch on the movements of the other UAVs, and process a large amount of information. In the following of this paper, obstacles are understood as being stationary and having locations and sizes known prior to the computation of the path.

The main contributions of our work are as follows:

- We consider a connectivity-aware path planning model for a fleet of UAVs with line formation. The proposed path planning ensures a collision-free trajectory starting from the departure position to the landing position. The model also takes into account a safe distance margin during the path planning to encompass uncertainties that may arise due to environmental disturbances. Because it is a centralized offline method, it is straightforward to prove that there cannot be any collision.
- The proposed technique separates the third layer (process of finding the optimal path) from the second layer (sending appropriate information to each UAV's navigation system). The second layer needs no exchange of information among UAVs, it is completely decentralized. This separation results from the choice of requiring UAVs to follow some spatial constraints.
- The proposed technique can adapt to multiple monitoring objectives and to a scalable use of multiple UAVs, without causing stability issues. This capacity again derives from the use of spatial constraints on UAVs. These modifications would be implemented in the fourth layer.
- The proposed scheme is implemented and evaluated in MATLAB and in a network simulator, respectively. The efficacy of this scheme remains satisfactory in terms of backhaul and fleet connectivity, collision avoidance, and fleet scalability with the variations in fleet speed, leader's location, number and nature of data originating nodes. This good networking behavior comes as no surprise, it is a consequence of the choice of a centralized offline proposition.

We organized the rest of the paper as follows. Section II presents the related work in the field of UAVs covering path planning and routing issues. Section III describes the system considered. Section IV provides in detail the proposed path planning method for a fleet of UAVs. Section V presents simulation testbed and experimental procedures. Results along with the discussion are presented in section VI. Section VII concludes the paper with possible future directions.

II. Related Work

Path planning with specific routing requirements for a fleet of UAVs becomes a challenging task, specifically with dynamic UAV topology and fleet deployment in an urban environment having obstacles and no-fly zones. This section reviews related work on path planning, routing protocol designing, and connectivity-aware path planning for UAV networks.

Zuo et al. [39] give a thorough review on the control issues regarding the UAVs, it is mainly focused on the fixed wings but it concerns also those having vertical take-off capabilities. It raises some important issues for rotorcraft UAVs: size of the stabilized flight enveloppe, robustness towards unknown dynamics (e.g. transport of unknown payload), design of observers for under-actuated UAVs. This review describes a large number of control techniques. [42] and [43] exemplify control techniques aiming at maintaining a short distance with a moving target: by adding an adaptive controller to an off-the-shelf module, and by using model-driven command inputs based on a created Lyapunov function.

Zhang et al. [31] propose a new UAV path planning approach using a navigational system-based localization error map in an urban environment. It avoids hazardous areas by reducing the effect of multipath and non-line-of-sight signal reception through predicting position errors in different areas. This approach uses a 3D building model, broadcast almanac, and ray-tracing simulation to generate the position error map and utilizes a modified A* algorithm to generate feasible trajectories. Additional work, for error map generation and processing, demands a high computational load before starting the flight.

Yin et al. [32] propose a multi-objective UAV path planning approach for a dynamic urban environment. This approach explores feasible paths while ensuring the safety of a UAV and guarantees travel time. It uses two types of safety index maps to tackle static and dynamic obstacles. The offline search uses a static safety map to avoid static obstacles and reduce travel time. The online search uses a dynamic safety map to go around unexpected obstacles quickly. The computational complexity of generating a static safety index map is high in this path planning scheme.

Chen et al. [24] come up with a path planning technique for multi-UAV formation in a known and realistic environment using a modified Artificial Potential Field (APF) method. This modification includes additional control force with its solution provided by the optimal control method. This approach introduces path planning and particle dynamics models for a single UAV. It formulates a path planning for a UAV formation by modeling it as a virtual velocity rigid body and a virtual target point. This approach requires a precise definition of repulsive potential to avoid a virtual point from entering into obstacle areas.

Filippis et al. [30] propose a UAV path planning for a 3D urban environment. This method uses a graph-based Theta* search algorithm to reduce the path length by including a lesser number of node points. Moreover, the Theta* algorithm used, results in smooth trajectories having fewer unnecessary altitude changes. This method reduces the searching time by using an effective nodal expansion technique where obstacles result in the blockage of a path. This method has higher computational complexity as compared to the A* algorithm.

Yoon et al. [33] propose an adaptive UAVs path planning to deliver delay-sensitive information during a natural disaster situation. The main objective of this technique is to find optimal UAV paths and serve the maximum number of nodes within a specific packet deadline. A distributed path planning mechanism determines the next visiting point constrained by delivery and packet deadline time. A task division method reduces the overall travel time by collaboratively distributing tasks among different UAVs. This approach lacks a real-time aspect and does not incorporate obstacles into its model.

Fabra et al. [28] propose a coordination protocol for maintaining a swarm of UAVs during a mission. In this centralized approach, a primary UAV synchronizes all other UAVs at intermediate points during a mission. This method attains a high level of swarm cohesion and a lower level of synchronization delays even with lossy communication channels. This technique does not consider obstacles, and BS cannot feed waypoints in real-time. It is imperative to have a reliable routing specifically for applications like disaster management, rescue operations, and battlefields.

Toorchi et al. [34] propose a skeleton-based intelligent routing protocol for dynamic networks of UAVs. It reduces routing complexity by exploiting the hierarchical and geometric structure of the swarm formation. It uses an adaptive leaf-like pipe acts as a central framework for routing purposes. During a change in UAV formation, UAVs move according to formation morphing technique to have minimum impact on the geometric addresses. The absence of path planning and BS-connectivity are shortcomings.

Hayat et al. [14] propose multiple objective path planning for Search And Rescue (SAR) operations with QoS requirements. This path planning approach considers two adaptive strategies. In the first one, search, inform, and monitoring tasks are optimized simultaneously. In the second case, search and inform are optimized initially, followed by monitoring optimization to get optimum positions. The SAR mission starts by detecting a static target in the shortest possible time. A UAV, after the detection of a target, carries the location information to the BS. This scheme optimizes coverage and connectivity but lacks in responding quickly, specifically when UAVs are deployed far enough from BS and require more time to form a connected network.

Yang et al. [38] propose a control strategy able to follow a dynamic target in a military context. The UAVs dynamics are precisely taken into account to both avoid obstacles, minimize energy and track the target. Because of the specific context, the issue is not to avoid collisions among UAVs but to quickly reach the target. The control strategy is mainly driven by artificial potential fields, fuzzy control is used to avoid local minima and instabilities.

Liu et al. [40] introduce a subtle objective, that of monitoring a moving target that might take advantage of some obstacles to escape monitoring. Their proposal is to combine, prediction of the target's movements and cooperation of sev-

TABLE I
SUMMARY OF VARIOUS PATH PLANNING AND ROUTING TECHNIQUES FOR UAVS

Reference	Research objective	Approach followed	Strengths	Weaknesses
Zhang et al. [31]	Reduce multipath effect in urban path planning	Use of predicted positions error map to avoid hazardous areas	Safely operable in urban environment with low altitude	Generation of error prediction maps High computational complexity of error maps Path may have sharp turning angles
Yin et al. [32]	Reduce travel time and avoid dynamic obstacles in urban environment	Use of two safety maps a static map to reduce travel time and a dynamic map to avoid unexpected obstacles	Safety of UAV and travel time guaranteed	High complexity of generating a static safety index map
Chen et al. [24]	Path planning for multiple-UAVs formation in realistic and known environment	Modified artificial potential field method with additional control force	Effective maintenance of formation and path following	Requirement of a precise definition of repulsive potential
Filippis et al. [30]	Shorter and smooth paths in 3D urban environment	Having less number of nodes and avoiding unnecessary altitude changes	Low searching time in case of re-planning	Higher computational complexity
Yoon et al. [33]	Optimal path and serve maximum nodes within packet specific deadline	Distributed path planning constrained by delivery and packet deadline time	Optimal paths with overall travel time and lower packet delivery delays	Lacking real-time aspect and does not consider obstacles
Fabra et al. [28]	Increase coverage area and accelerate mission completion time	Use of swarm synchronization of all UAVs at intermediate points	Swarm cohesion Lower synchronization delays even with lossy communication channels	No real-time aspect No obstacle consideration
Toorchi et al. [34]	Reduce routing complexity for dynamic UAVs networks	Skeleton-based (swarm structure) intelligent routing protocol	Higher throughput Adaptable to changes in formation	Absence of path planning No connectivity with BS (No real-time aspect)
Hayat et al. [14]	Optimize coverage and connectivity in search and rescue missions	Joint optimization of search, inform and monitoring tasks Incorporation of communication in the path design process	Tunable to application requirements to prioritize coverage over connectivity or the other way around	Lacking in response time, specifically when UAVs are deployed far enough from BS
Yang et al. [38]	Tracking of dynamic targets	Use of artificial potential field, fuzzy control and choice of the best control strategy	Fast tracking, high precision, strong stability and avoid chattering	Collision avoidance among UAVs is not considered
Liu et al. [40]	Tracking of moving targets while avoiding obstacles	Use of fuzzy constraints, a cost function modeling the monitoring ability and a two-state control strategy for the formation	High monitoring ability, with true cooperation between UAVS	Experiments show that it is not collision-free by design
Wu et al. [41]	Path planning, energy consumption and quality of service	Q-learning and genetic algorithm	Energy consumption at turning points	Collision avoidance is not considered
Our proposed method	Connectivity-aware collision-free path planning to provide real-time aspect for delay sensitive applications	Path planning ensuring fleet connectivity and all-time connectivity with at least one BS	Collision-free paths and robust to sensor uncertainties Real-time communication Scalability Continuous coverage over the designed trajectory	Lacks consideration of UAV dynamics, but path planning considers a parameter to compensate for uncertainties that may arise due to environmental disturbances

eral UAVs, to maintain a constant monitoring. It uses a cost function modeling the monitoring capacity, fuzzy constraints to model the effects of obstacles and a two-state control strategy to adapt the formation.

Wu et al. [41] consider a set of UAVs in an urban area with specific missions and aiming at maximizing a quality of service while minimizing energy consumption. Their proposed technique yields a path to a given approximate location, and using Q-learning, determines an optimal location. It interestingly takes into account the energy consumption of turning points contained in the path. It is also embedded in a proposed three-layer architecture.

Table I summarizes all path planning and routing techniques discussed in this section.

III. SYSTEM DESCRIPTION

We first describe the considered use cases (section III-A), the objectives reached by our proposal (section III-B) and finally how these objectives fit with the use cases (section III-C).

A. Mission Outline

We consider an Inform and Monitoring (IM) mission with a fleet of UAVs along with the deployment of multiple BSs in an urban environment. Before the mission starts, the proposed path planner is informed about the location of obstacles, departing and landing points and angles, the communication distance, the uncertainty of the navigation system in terms of safe margin distance, speed of UAVs, location of intermediate BSs to visit during the mission. The connectivity-aware path planner estimates the number of UAVs required using the information given and informs each UAV accordingly about its mission-specific collision-free trajectory. The tracking of the designed trajectory by a fleet of UAVs mainly depends upon the specific application. The proposed scheme considers the following application scenarios:

1) Provision of situational information, continuous monitoring, and temporary network connectivity at a natural disaster location quickly. Moreover, the fleet on their

way will inform all vehicles on the roads about the disaster and keep them away from this specific path to help the rescue team interfere quickly.

2) Continuous tracking and monitoring of a public rally. The first UAV can lead the fleet according to the variation in the speed of the rally.

Both scenarios could take place in rural or urban landscapes, provided that the environmental conditions remain favorable (many UAVs cannot sustain moderate wind gusts or a small rain) and that the UAV regulations are being observed.

B. Problem Statement

Before formulating the objectives and the requirements of our proposal, we first illustrate the concepts with the following four figures.

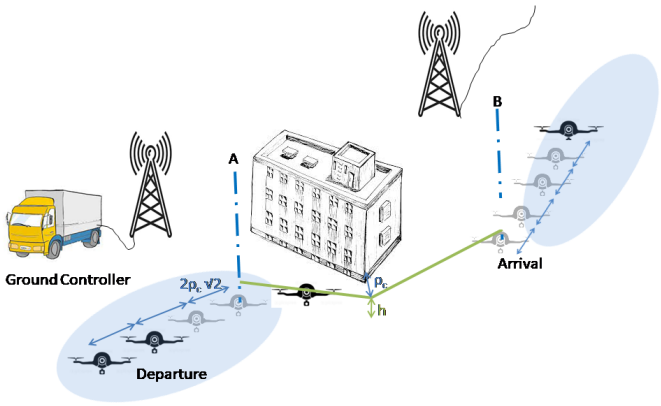


Fig. 1. General layout illustrating the problem statement with four UAVs and an obstacle figured as a building.

Figure 1 illustrates the problem statement. The obstacles are figured here with a building in the middle. UAVs are to travel from a departure point denoted A and indicated with a blue vertical dash dotted line, to an arrival point, denoted B and indicated in the same way. There is already a base station at B. The green line is the path joining point A to point B while avoiding the building. A lorry on the left brings, the UAVs, a base station, and a ground controller communicating to all UAVs by means of any of the two base stations. At first, the four UAVs are put in a line close to each other, starting from point A. Then the first UAV takes off, travels along the green line beyond point B and reaches its specific parking slot where it lands. Figure 1 captures a later moment when the second UAV is traveling along the green line. Light shaded UAVs indicate the position that UAVs had at the beginning of the experiment or positions that they will have at the end of the experiment.

Figure 2 shows a 2D-map representing figure 1: the building is represented as a red rectangle in the middle, it constitutes an obstacle. The lorry and its base station is represented as point A, it is the beginning point. The green line on the left of A shows the area where the different UAVs are lying before

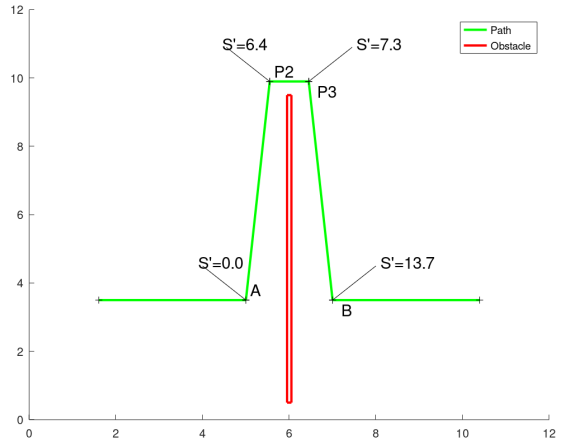


Fig. 2. 2D-map modeling figure 1 with an obstacle figuring the building in the middle and the path joining the lorry to the right base station.

departure. The second base station is represented as point B. P_2 and P_3 are two interest points derived from the obstacle. We can see on this path, that the distance of the obstacle with the path is the same with line segment P_2P_3 that is ρ_c. This ensures that any drone following the path from a distance no greater than ρ_c cannot collide with the obstacle. For the sake of clarity, we introduce S' measuring the length traveled for all UAVs, beginning at A, so that for any of the four UAVs, given the path and a specific value of S', we have the exact position of the UAV. Note that it is different from $\mathcal{S}$ defined later on, in equation (2), whose value is related to time and refers to different positions depending on the UAV considered.

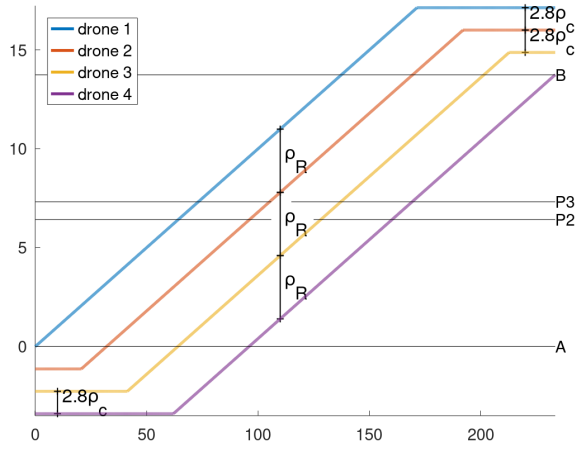


Fig. 3. Movement of the four UAVs shown in figure 1 and described as four time-functions in the 1D-space of the distance traveled beyond A denoted S'.

In figure 3, the movement of the four UAVs shown in figure 1 are here represented as four functions that each maps the elapsed time since the beginning of the experiment denoted t into the distance traveled beyond A denoted S'. Because each value of S' refers to a specific point in the

path, points $\mathsf{A}, \mathsf{P}_2, \mathsf{P}_3$ and B are shown as horizontal black lines. Departure and arrival are the areas corresponding to S' negative and to S' greater than 13.7 (i.e. the value of S' at B). In those two areas, the small vertical arrows at the bottom left and at the top right indicate the difference in traveled distance between two successive UAVs, it is equal to $2\sqrt{2}\rho_c$. When UAVs are traveling between A and B, that is between the two horizontal lines, A and B, this difference in traveled distance is increased up to ρ_R. It is the greatest distance for which communication is still working. ρ_R is indicated with the three vertical arrows at the center.

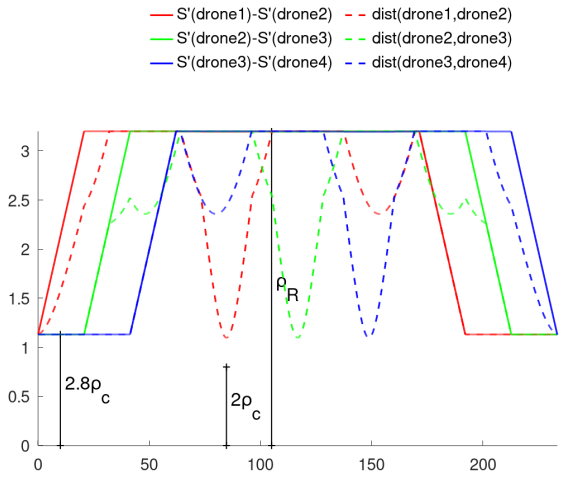


Fig. 4. Relative positions of successive UAVs shown by their difference in their S' values (plain lines) and by their Euclidean distance (dotted lines) as functions of time.

Figure 4 shows, with plain lines, the time evolution of the difference in distance traveled between the first and the second UAV, the second and the third UAV, the third and the fourth UAV. It also shows, with dotted lines, the time evolution of the distance between each of these pairs of UAVs. We can see that each of these pairs of UAVs are connected, first because the difference in their traveled distances is below ρ_R, as indicated by the long vertical arrow in the middle. Secondly, the triangle inequality ensures that the distance between them is less than their difference in traveled distances as all dotted lines are below the plain line of the same color. We also see that collisions are avoided between any two successive UAVs, because, first, the difference in their traveled distances is always greater than $2\sqrt{2}\rho_c$ as indicated by the short vertical arrow on the bottom left. Secondly, thanks to some angular constraints on the path, this ensures that the distance between each pair members is greater than $2\rho_c$, as indicated by the vertical arrow which marks the lower bound of all dotted lines.

Our proposal ensures the following four objectives.

1) Collisions are avoided between any two UAVs.

$$\forall t, n_1 \neq n_2,\ \mathsf{R}_{n_1}(t) \neq \mathsf{R}_{n_2}(t) \tag{1}$$

where $\mathsf{R}_n(t)$ stands for the location of the n^{th}-UAV at time t.

2) Collisions are avoided between any UAV and any stationary obstacle.

$$\forall t, n, i,\ \mathsf{R}_n(t) \notin \overline{\mathsf{C}_i\mathsf{D}_i} \tag{2}$$

where $\overline{\mathsf{C}_i\mathsf{D}_i}$ are the line segments modeling obstacles.

3) All UAVs may communicate at all times with A or B which are base stations.

$$\mathsf{R}_n(t) \overline{\bowtie} \mathsf{A} \text{ or } \mathsf{R}_n(t) \overline{\bowtie} \mathsf{B} \tag{3}$$

where $\overline{\bowtie}$ denotes the ability to do UAV-network communication.

4) The speed at which all UAVs are on average travelling can be set freely.

$$\forall n,\ \left| \frac{d}{dt} \mathsf{V}_n(t) \right| \overset{a.e.}{=} v(t) \tag{4}$$

where $\mathsf{V}_n(t)$ is the n^{th} virtual UAV that the n^{th} UAV follows closely. This equation holds only once the n^{th}-UAV has left its starting point and as long as it has not reached its final point.

In our proposal, these objectives derive from precise and constraining rules to which all UAVs are to abide. These rules are displayed in the form of four assumptions.

1) The navigation system of the n^{th}-UAV $\mathsf{R}_n(t)$ is able to follow $\mathsf{V}_n(t)$ within a safe margin ρ_c.

$$\forall n,\ \mathsf{R}_n(t)\mathsf{V}_n(t) < \rho_c \tag{5}$$

where $\mathsf{R}_n(t)\mathsf{V}_n(t)$ is the Euclidean distance between $\mathsf{R}_n(t)$ and $\mathsf{V}_n(t)$.

2) The prescribed speed $v(t)$, at time t, is not known to any UAV before time t. It should not be greater than a maximum speed denoted $v_{\max}$ (i.e. $v_{\max}$ has to be smaller than the technical speed threshold as it should be followed by the UAV's navigation system possibly on a long period of time).

$$v(t) \leq v_{\max} \tag{6}$$

To simplify the mathematical technicalities, $v(t)$ is assumed to be a finite linear combination of indicator functions having a positive limit as $t \to +\infty$. The supports of these indicator functions are left-closed and right-open intervals.

3) Each UAV is able to communicate with another UAV when their mutual distance is below ρ_r. Their ability to directly communicate with one another is denoted $\bowtie$.

$$\mathsf{R}_{n_1}(t)\mathsf{R}_{n_2}(t) \leq \rho_R \quad \Leftrightarrow \quad \mathsf{R}_{n_1}(t) \bowtie \mathsf{R}_{n_2}(t) \tag{7}$$

The departure and terminal point, denoted A and B, are also base stations.

$$\begin{aligned} \mathsf{R}_n(t)\mathsf{A} \leq \rho_R \quad &\Leftrightarrow \quad \mathsf{R}_n(t) \bowtie \mathsf{A} \\ \mathsf{R}_n(t)\mathsf{B} \leq \rho_R \quad &\Leftrightarrow \quad \mathsf{R}_n(t) \bowtie \mathsf{B} \end{aligned} \tag{8}$$

ρ_R is generally much greater than ρ_c, we assume here that at least

$$\rho_R \geq 2(1+\sqrt{2})\rho_c \tag{9}$$

4) The line segments $\overline{\mathsf{C}_i\mathsf{D}_i}$, modeling the obstacles, are motionless and known prior to departure .

C. Adapting our proposal to each specific use cases

1) Natural Disaster: We would like to show here how our proposal could help implementing in practise the two use cases presented in section III-A.

As for the first use case, the following tasks are being drawn up.

- Bring a lorry at a relevant location, shown here at the center left. This lorry should contain the UAVs, a ground control, an equipment listing all the obstacles in the area, a base station and a recharging device.
- Select the static points that UAVs should be monitoring, either to retrieve information or to prevent residents from coming in.
- Solve using a classical heuristic the Travel Salesman Problem and order the different points.
- Use our proposal to bring a UAV at the first point, while remaining connected. Our proposal has to be slightly modified because there is no base station at the destination point: a higher number of UAVs are required to maintain the connection with the base station. The UAVs' movement is stopped when the first UAV joins the first point. Add the path being used as supplementary obstacles for later use to find paths joining the other points.
- Use our proposal to bring the first UAV from the first point to the second and to bring a second UAV to the first point. Again each UAVs' movement is stopped when a point is reached until the UAV reaching this point leaves for the next point. Add the second path being used as supplementary obstacles for later use to find paths joining the remaining points.
- Repeat the previous step until there is a UAV at each point.
- Each time there is a risk that a UAV becomes discharged, the same process is repeated this time, to bring the UAV at risk to the initial point, UAVs are expected to follow the different paths joining the different points, in the same order.
- When the UAVs are no longer needed, they are moved back to the lorry in the same manner.

The second use case could be implemented with the following tasks.

- Bring a lorry near the starting point of the rally denoted as A and shown here at the center left. This lorry should contain the UAVs, a ground control, an equipment listing all the obstacles along the rally, a base station and a recharging device. It is assumed here that there is a base station and a recharging device where the rally ends, denoted B.
- Use our proposal to map a path joining A to B. And use our proposal to bring a UAV at the beginning of the rally along this path. ρ_R is now set not only to preserve connectivity with base stations but also as an upper bound of the monitoring capacities of each UAV. It is assumed here that to efficiently monitor the rally a greater number of UAVs are required. The path is added as supplementary obstacles for later use to find a return path.
- The speed of the whole formation is set by the first UAV along with the speed of the front of the rally. To ensure that all UAVs change their speed at the same time, the broadcast messages contain, both the speed and the time at which it comes into effect.
- When recharging is needed for a given UAV, it broadcasts the corresponding signal and the time at which it takes effect. This signal indicates that all previous UAVs are to join B, the remaining UAVs are to follow to maintain connectivity. Our proposal is used to map a return path joining B to A.

IV. Proposed Algorithm

A. Kinematics of UAVs along a given path

We consider in this section being provided with a path $\mathbf{P} \in \mathbb{P}$ defined in the following definition. We propose here a precise time and space description of the location of each UAV and give theorems showing that this description meets the objectives listed in section III-B when $\mathbf{P}$ complies with some prerequisites.

Definition 1. *$\mathbb{P}$ is a set of paths, each defined as array of distinct points.*

$$\begin{aligned}\mathbf{P} = [\mathsf{P}_1 \dots \mathsf{P}_K] \in \mathbb{P} \quad \Leftrightarrow \quad \forall k \in \{1 \dots K\},\\ \mathsf{P}_k \in \mathbb{M} \text{ and } [k \neq k' \Rightarrow \mathsf{P}_k \neq \mathsf{P}_{k'}]\end{aligned} \tag{10}$$

where $\mathbb{M}$ is a set of points in a 2D-space.

We consider that the provided path is *complete* when it joins A and B. The set of all complete paths is denoted $\mathbb{P}_c$.

$$\mathbb{P}_c = \{[\mathsf{P}_1 \dots \mathsf{P}_K] \in \mathbb{P} \,|\, \mathsf{P}_1 = \mathsf{A} \text{ and } \mathsf{P}_K = \mathsf{B}\} \tag{11}$$

The length of a path is

$$\mathcal{L}(\mathbf{P}) = \sum_{k=1}^{K-1} \mathsf{P}_k\mathsf{P}_{k+1} \tag{12}$$

To describe how UAVs are displayed in the departure area and in the arrival area, we define two rays starting at A and B and having an angle of $\pi + \Theta_\mathsf{A}$ and Θ_B with $\overrightarrow{\mathsf{AB}}$.

$$\begin{aligned}\Delta_\mathsf{A} &= \left\{ \mathsf{M} \,\middle|\, \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{AM}}) = \pi + \Theta_\mathsf{A} \right\} \text{ and}\\ \Delta_\mathsf{B} &= \left\{ \mathsf{M} \,\middle|\, \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{BM}}) = \Theta_\mathsf{B} \right\}\end{aligned} \tag{13}$$

These rays should be far enough from any obstacles.

$$\forall i,\ \mathcal{D}(\Delta_\mathsf{A}, \overline{\mathsf{C}_i\mathsf{D}_i}) \geq \rho_c \text{ and } \mathcal{D}(\Delta_\mathsf{B}, \overline{\mathsf{C}_i\mathsf{D}_i}) \geq \rho_c \tag{14}$$

These rays should also be far enough from each others.

$$\mathcal{D}(\Delta_\mathsf{A}, \Delta_\mathsf{B}) \geq 2\rho_c \tag{15}$$

To avoid collisions during departure and arrival, UAVs are displayed along Δ_A at departure and Δ_B at arrival. The distance between each consecutive time slots is $2\rho_c\sqrt{2}$.

The orientation's motion in Δ_A and Δ_B are described with two unitary vectors $\overrightarrow{e_0}$ and $\overrightarrow{e_K}$.

$$\angle\left(\overrightarrow{\mathsf{AB}}, \overrightarrow{e_0}\right) = \Theta_\mathsf{A} \text{ and } \angle\left(\overrightarrow{\mathsf{AB}}, \overrightarrow{e_K}\right) = \Theta_\mathsf{B} \tag{16}$$

The exact starting and finishing locations of each virtual UAV are N points denoted $\mathsf{P}_{0,n}$ and $\mathsf{P}_{K+1,n}$.

$$\overrightarrow{\mathsf{AP}_{0,n}} = -(n-1)\overrightarrow{e_0} \text{ and } \overrightarrow{\mathsf{BP}_{K+1,n}} = (N-n)\overrightarrow{e_K} \quad (17)$$

To ease notations, an unneeded index n is added to P_k.

$$\forall k \in \{1 \dots K\}, \quad \mathsf{P}_{k,n} = \mathsf{P}_k \quad (18)$$

Once $\mathbf{P}$ is defined, the exact location of $\mathsf{V}_n(t)$ can be derived from the distance traveled. The following definition computes $\mathcal{S}(t)$ a distance, s, that a non-stopping UAV would have traveled at time t.

Definition 2. *$\mathcal{S}(t)$ maps time into traveled distance.*

$$\mathcal{S}(t) = \int_0^t v(\tau)\, d\tau \quad (19)$$

Describing the kinematics of each UAV requires the computation of an inverse mapping, yielding the time t by which this non-stopping UAV would have traveled a given distance s. Theorem 1 provides such an inverse mapping and gives an explicit definition. Its proof is available in [44].

Theorem 1. *Let $v(t)$ follow assumption 2, there exists $t = \Phi[s]$ with the following property:*

$$\begin{cases} \mathcal{S}(\Phi[s]) = s & \\ \frac{d}{ds}\Phi[s] = \frac{1}{v(\Phi[s])} & \text{if} \quad v(\Phi[s]) \neq 0 \\ \lim\limits_{s' \to s^-} \Phi[s'] < \Phi[s] & \text{if} \quad v(\Phi[s]) = 0 \end{cases} \quad (20)$$

Moreover at time t, $\Phi[s]$ is known to all UAVs when $\Phi[s] \leq t$

With the following definition, we get a second inverse mapping of $\mathcal{S}(t)$ that uses only the information available to all UAVs at time t.

Definition 3.

$$\Phi_t[s] = \begin{cases} \Phi[s] & \text{if} \quad \Phi[s] \leq t \\ +\infty & \text{if} \quad \text{not} \end{cases} \quad (21)$$

To ease notations, we will be using $\Phi[s]$ instead of $\Phi_t[s]$.

Note that in the case of the no speed-change scenario, $\mathcal{S}(t)$ and $\Phi[s]$ are linear:

$$\mathcal{S}(t) = vt \text{ and } \Phi[s] = \frac{s}{v} \quad (22)$$

with $v = v(t)$ being the constant speed.

We define two time-varying lags, they are expressed in terms of distance traveled: $\Delta\mathcal{S}_c$, meant to avoid collision and $\Delta\mathcal{S}_r$, meant to maintain radio-connection.

$$\Delta\mathcal{S}_c = 2\rho_c\sqrt{2} \text{ and } \Delta\mathcal{S}_r = \rho_R - 2\rho_c \quad (23)$$

This travel is described for each UAV by durations Δt_k and distances $L_{k,n}$ that have or could have been traveled. $L_{k,n}$ are displayed later on.

$$\begin{aligned} &\forall k \in \{-1 \dots K+2\}, \forall n \in \{1 \dots N\} \\ &\Delta t_{k,n} = \Phi\left[L_{k+1,n}\right] - \Phi\left[L_{k,n}\right] \text{ and } L_{-1,n} = 0 \end{aligned} \quad (24)$$

For each UAV, its travel is composed of five phases as illustrated in figure 5 (for the no speed-change scenario). In

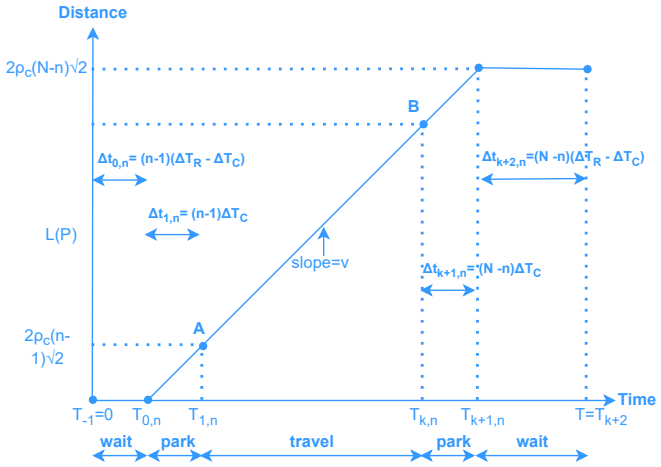


Fig. 5. Line chart of the distance traversed by the n^{th} UAV for the no speed-change scenario.

its first phase, the UAV remains at its parking slot waiting that it becomes useful to maintain connectivity between the previous UAV and the remaining UAVs. In its second phase, it travels through the parking area to enter the path joining A and B. In its third phase, it travels along that path. In its fourth phase, it travels in the arrival parking area and reaches its parking slot. In its fifth phase, it waits the remaining UAVs reach their parking slots. Note that the first UAV ($n = 1$) skips the two first phases and the last UAV ($n = N$) skips the two last phases.

1) At $k = -1$, for $n > 1$, the n^{th} virtual UAV remains at $\mathsf{P}_{0,n}$, waiting the maximum amount of time for which the connectivity with the previous UAV is preserved. The traveled distance is:

$$L_{0,n} = (n-1)(\Delta\mathcal{S}_r - \Delta\mathcal{S}_c)$$

2) At $k = 0$, for $n > 1$, the n^{th} virtual UAV travels from $\mathsf{P}_{0,n}$ to $\mathsf{P}_1 = \mathsf{A}$, a distance that prevents mutual UAV collisions in the departure area, Δ_{A}. The traveled distance is:

$$L_{1,n} = L_{0,n} + (n-1)\Delta\mathcal{S}_c = (n-1)\Delta\mathcal{S}_r$$

3) At $k \in \{1 \dots K-1\}$, the n^{th} virtual UAV travels from $\mathsf{P}_1 = \mathsf{A}$ to $\mathsf{P}_K = \mathsf{B}$ along the path $\mathbf{P}$ covering a distance of $\mathcal{L}(\mathbf{P})$

$$L_{K,n} = L_{1,n} + \mathcal{L}(\mathbf{P})$$

This phase is decomposed into $K-1$ sub-phases, each being a travel from P_k to P_{k+1}

$$L_{k+1,n} = L_{k,n} + \mathsf{P}_k\mathsf{P}_{k+1}$$

4) At $k = K$, for $n < N$, the n^{th} virtual UAV enters in the arrival area Δ_{B} and travels up to $\mathsf{P}_{K+1,n}$ covering a distance preventing mutual UAV collisions.

$$L_{K+1,n} = L_K + (N-n)\Delta\mathcal{S}_c$$

5) At $k = K+2$, for $n < N$, the n^{th} virtual UAV waits at $\mathsf{P}_{K+1,n}$ for the other UAVs to reach their final destination points.

$$\begin{aligned} L_{K+2,n} &= L_{K+1,n} + (N-n)\left(\Delta\mathcal{S}_r - \Delta\mathcal{S}_c\right) \\ &= \mathcal{L}(\mathbf{P}) + (N-1)\Delta\mathcal{S}_r \end{aligned}$$

The overall travel duration, T is

$$T = \sum_{k=1}^{K+2} \Delta t_{k,n} = \Phi\left[\mathcal{L}(\mathbf{P}) + (N-1)\Delta\mathcal{S}_r\right] \tag{25}$$

In the case of the no speed-change scenario, note that the duration of phases 1,2,4 and 5, namely $\Delta t_{0,n}$, $\Delta t_{1,n}$, $\Delta t_{K+1,n}$, $\Delta t_{K+2,n}$ are specific to each UAV, not the third phase nor T. We define unit vectors $\overrightarrow{e_k}$, one for each segment line contained in the path $\mathbf{P}$ and two for Δ_A and Δ_B.

$$\overrightarrow{e_k} = \frac{\overrightarrow{\mathsf{P}_k\mathsf{P}_{k+1}}}{\mathsf{P}_k\mathsf{P}_{k+1}} \text{ for } k \in \{1 \dots K-1\}$$

We are now making use of a saturated ramp function defined as:

$$\mathcal{R}_a^b(x) = \begin{cases} 0 & \text{when} \quad x \le a \\ x-a & \text{when} \quad x \in (a,b) \\ b-a & \text{when} \quad x \ge b \end{cases} \tag{26}$$

The location of each virtual UAV can then be described in a shortened equation:

$$\begin{aligned} &\forall t \in [0,T],\ \forall n \in \{1 \dots N\}, \\ &\mathsf{V}_n(t) = \mathsf{A} - (n-1)\Delta\mathcal{S}_c\ \overrightarrow{e_0} \\ &\quad + \sum_{k=0}^{K+1} \mathcal{R}_{L_{k,n}}^{L_{k+1,n}}\left(\mathcal{S}(t)\right)\ \overrightarrow{e_k} \end{aligned} \tag{27}$$

The required number of UAVs is equal to:

$$N = \left\lfloor \frac{\mathcal{L}(\mathbf{P})}{\rho_R - 2\rho_c} \right\rfloor \tag{28}$$

Remark 1. *Equations (17), (23), (24), (27), (28), tell us that the information to be broadcast to all UAVs is composed of N, $\mathsf{P}_1 \dots \mathsf{P}_K$, ρ_c, ρ_R, θ_A, θ_B and $(v(\tau))_{\tau \le t}$.*

Theorem 2 validates equation (27). Its proof is available in [44].

Theorem 2. *Let equation (27) be the description of the moving virtual UAV $\mathsf{V}_n(t)$ using $v(t)$. Then this description is consistent with the path $\mathbf{P}$.*

$$\begin{cases} \mathsf{V}_n(t) \in \Delta_\mathsf{A} & \text{if} \quad t \in [0, \Phi[L_{1,n}]] \\ \mathsf{V}_n(t) \in \overline{\mathsf{P}_k\mathsf{P}_{k+1}} & \text{if} \quad t \in [\Phi[L_{k,n}], \Phi[L_{k+1,n}]] \\ & \quad \text{for } k \in \{1 \dots K-1\} \\ \mathsf{V}_n(t) \in \Delta_\mathsf{B} & \text{if} \quad t \in [\Phi[L_{K,n}], T] \end{cases} \tag{29}$$

This description is also consistent with objective 4.

$$\begin{cases} \left|\frac{d}{dt}\mathsf{V}_n(t)\right| = 0 & \text{if} \quad t \in [0, \Phi[L_{0,n}]] \\ \left|\frac{d}{dt}\mathsf{V}_n(t)\right| \overset{a.e.}{=} v(t) & \text{if} \quad t \in [\Phi[L_{0,n}], \Phi[L_{K+1,n}]] \\ \left|\frac{d}{dt}\mathsf{V}_n(t)\right| = 0 & \text{if} \quad t \in [\Phi[L_{K+1,n}], T] \end{cases} \tag{30}$$

Theorem 3 ensures the avoidance of any obstacle collision using assumption 1, it thereby yields a first prerequisite on the path. It makes use of, $\mathcal{D}$, the Euclidean distance between two sets of points. Proof of theorem 3 is available in [44].

Theorem 3. *When considering a complete path whose line segments are sufficiently far from any obstacles, the movement of the virtual points as described in equation (27) cannot lead to obstacle collision.*

$$\begin{aligned} &[\mathsf{P}_1 \dots \mathsf{P}_K] \in \mathbb{P}_c \text{ and } \forall i, \forall k \in \{1 \dots K-1\}, \\ &\mathcal{D}(\overline{\mathsf{P}_k\mathsf{P}_{k+1}}, \overline{\mathsf{C}_i\mathsf{D}_i}) \ge \rho_c,\ \mathcal{D}(\Delta_\mathsf{A}, \overline{\mathsf{C}_i\mathsf{D}_i}) \ge \rho_c, \\ &\underbrace{\mathcal{D}(\Delta_\mathsf{B}, \overline{\mathsf{C}_i\mathsf{D}_i}) \ge \rho_c}_{\Rightarrow \quad \forall t, n, i \quad \mathsf{R}_n(t) \notin \overline{\mathsf{C}_i\mathsf{D}_i}} \end{aligned} \tag{31}$$

Theorem 4 adds some other prerequisites and ensures with assumption 4, the avoidance of any mutual UAV collisions. Proof of theorem 4 is available in [44].

Theorem 4. *Let $\mathbf{P} = [\mathsf{P}_1\mathsf{P}_2 \dots \mathsf{P}_K]$ be a complete path, whose consecutive line segments have an absolute angle no greater than $\frac{\pi}{2}$ and for which any two non-consecutive line segments are always at a distance of at least $2\rho_c$ from each other. Then the movement of the virtual points as described in equation (27) cannot yield any mutual UAV collision.*

$$\begin{aligned} &\text{For } k \in \{2 \dots K-1\} \\ &\quad |\theta_A - \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{P}_1\mathsf{P}_2})| \le \tfrac{\pi}{2} \\ &\quad |\angle(\overrightarrow{\mathsf{P}_{k-1}\mathsf{P}_k}, \overrightarrow{\mathsf{P}_k\mathsf{P}_{k+1}})| \le \tfrac{\pi}{2} \\ &\quad |\theta_B - \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{P}_{K-1}\mathsf{P}_K})| \le \tfrac{\pi}{2} \\ &\quad \mathcal{D}(\Delta_\mathsf{A}, \Delta_\mathsf{B}) \ge 2\rho_c \\ &\quad \mathcal{D}(\Delta_\mathsf{A}, \overline{\mathsf{P}_k\mathsf{P}_{k+1}}) \ge 2\rho_c \\ &\quad \mathcal{D}(\Delta_\mathsf{B}, \overline{\mathsf{P}_{k-1}\mathsf{P}_k}) \ge 2\rho_c \end{aligned} \tag{32}$$

$$\begin{aligned} &\text{For } k, k' \in \{1 \dots K-1\} \\ &\quad |k-k'| > 1 \Rightarrow \mathcal{D}(\overline{\mathsf{P}_k\mathsf{P}_{k+1}}, \overline{\mathsf{P}_{k'}\mathsf{P}_{k'+1}}) \ge 2\rho_c, \\ &\underbrace{\qquad\qquad}_{\Rightarrow \quad \forall t, n_1 \ne n_2, \quad \mathsf{R}_{n_1}(t) \ne \mathsf{R}_{n_2}(t)} \end{aligned}$$

Theorem 5 and assumption 1 ensure that all UAVs remain connected with A or B. Proof of theorem 5 is available in [44].

Theorem 5. *Let $\mathbf{P}$ be a complete path and considering a number of UAVs no smaller than the value proposed in equation (28). Then the movement of the virtual points as described in equation (27) do not lose connectivity.*

$$\begin{aligned} &\mathbf{P} \in \mathbb{P}_c \text{ and } N > \frac{\mathcal{L}(\mathbf{P})}{\rho_R - 2\rho_c} - 1 \\ &\Rightarrow \forall t, n, \begin{pmatrix} \mathsf{R}_n(t) \boxtimes \mathsf{A} \\ \text{or} \\ \mathsf{R}_n(t) \boxtimes \mathsf{B} \end{pmatrix} \end{aligned} \tag{33}$$

The five prerequisites needed by theorem 3 and 4 and defined in equations (31) and (32) give rise to a new definition, that of a a *valid* path denoted $\mathbb{P}_v$.

Definition 4. *$[\mathsf{P}_1 \dots \mathsf{P}_K] \in \mathbb{P}_c$ is a valid path if and only if*

1) *it is far enough from any obstacle:*

$$\forall k \in \{1 \dots K-1\}, \forall i,\ \mathcal{D}(\overline{\mathsf{P}_k\mathsf{P}_{k+1}}, \overline{\mathsf{C}_i\mathsf{D}_i}) \geq \rho_c \quad (34)$$

2) *it does not cross or come close to the departure and arrival areas:*

$$\begin{aligned} &\forall k \in \{2 \dots K-1\},\ \mathcal{D}(\overline{\mathsf{P}_k\mathsf{P}_{k+1}}, \Delta_\mathsf{A}) \geq 2\rho_c \\ &\text{and}\ \ \mathcal{D}(\overline{\mathsf{P}_{k-1}\mathsf{P}_k}, \Delta_\mathsf{B}) \geq 2\rho_c \end{aligned} \quad (35)$$

3) *its first and last bends are not too sharp:*

$$\begin{aligned} &\left|\theta_A - \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{P}_1\mathsf{P}_2})\right| \leq \frac{\pi}{2}, \quad \text{and} \\ &\left|\theta_B - \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{P}_{K-1}\mathsf{P}_K})\right| \leq \frac{\pi}{2} \end{aligned} \quad (36)$$

4) *its others bends are also not too sharp:*

$$\forall k \in \{2 \dots K-1\},\ \left|\angle(\overrightarrow{\mathsf{P}_{k-1}\mathsf{P}_k}, \overrightarrow{\mathsf{P}_k\mathsf{P}_{k+1}})\right| \leq \frac{\pi}{2}, \quad (37)$$

5) *it does not cross or come close to its own path:*

$$\begin{aligned} &\forall k, k' \in \{1 \dots K-1\},\ |k-k'| > 1 \\ &\Rightarrow \mathcal{D}(\overline{\mathsf{P}_k\mathsf{P}_{k+1}}, \overline{\mathsf{P}_{k'}\mathsf{P}_{k'+1}}) \geq 2\rho_c \end{aligned} \quad (38)$$

B. Path planning

We propose here an algorithm for providing iteratively a valid path. It builds a weighted directed graph and then repeatedly increases some edge weights to help the Dijkstra algorithm to find a valid path. This need to help the Dijkstra algorithm stems from the non-local nature of conditions 4 and 5 of definition 4. When an edge is traversed by a valid path, other edges may, as a consequence, be excluded from being traversed by this path. The Dijkstra algorithm is repeatedly applied, each time increasing the weights of the problematic edges, until a valid path is found. Nodes are generated in section IV-B1, edges in section IV-B2, and weights in section IV-B3. Finally section IV-B4 discloses the proposed Dijkstra-based heuristic extracting a valid path from this directed graph. Note that ρ_R is not being used for path planning.

1) Generating the graph nodes: The nodes of the graph being built are points denoted $\mathsf{M} \in \mathbb{M}$, they are generated using the provided knowledge of the environment. Below is the list of these $\mathbb{M}$-points.

- A, B are two required points.
- Four points $\mathsf{C}_i^+, \mathsf{C}_i^-, \mathsf{D}_i^-, \mathsf{D}_i^+$ are defined for each line segments $\overline{\mathsf{C}_i\mathsf{D}_i}$, they are displayed so as to allow going around the obstacle. These four points are located at a distance of $\sqrt{2}\rho_c$ of each segment end, they have an angle of $\pm\frac{\pi}{4}$ with respect to $\overline{\mathsf{C}_i\mathsf{D}_i}$.

$$\begin{cases} \angle(\overrightarrow{\mathsf{D}_i\mathsf{C}_i}, \overrightarrow{\mathsf{C}_i\mathsf{C}_i^-}) = \frac{\pi}{4} & \mathsf{C}_i\mathsf{C}_i^- = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{D}_i\mathsf{C}_i}, \overrightarrow{\mathsf{C}_i\mathsf{C}_i^+}) = -\frac{\pi}{4} & \mathsf{C}_i\mathsf{C}_i^+ = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{C}_i\mathsf{D}_i}, \overrightarrow{\mathsf{C}_i\mathsf{D}_i^-}) = -\frac{\pi}{4} & \mathsf{C}_i\mathsf{D}_i^- = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{C}_i\mathsf{D}_i}, \overrightarrow{\mathsf{C}_i\mathsf{D}_i^+}) = \frac{\pi}{4} & \mathsf{C}_i\mathsf{D}_i^+ = \rho_c\sqrt{2} \end{cases} \quad (39)$$

- Four extra points, denoted $\mathsf{A}^-, \mathsf{A}^+, \mathsf{B}^-, \mathsf{B}^+$, are also considered to withhold the use of sharp turns at departure and arrival. They are located at a distance of $\rho_c\sqrt{2}$ of A and B and have an angle of $\theta_\mathsf{A} \pm \frac{\pi}{2}$ and $\theta_\mathsf{B} \pm \frac{\pi}{2}$ with $\overrightarrow{\mathsf{AB}}$.

$$\begin{cases} \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{AA}^+}) = \theta_\mathsf{A} - \frac{\pi}{2} & \mathsf{AA}^+ = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{AA}^-}) = \theta_\mathsf{A} + \frac{\pi}{2} & \mathsf{AA}^- = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{BB}^+}) = \theta_\mathsf{B} + \frac{\pi}{2} & \mathsf{BB}^+ = \rho_c\sqrt{2} \\ \angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{BB}^-}) = \theta_\mathsf{B} - \frac{\pi}{2} & \mathsf{BB}^- = \rho_c\sqrt{2} \end{cases} \quad (40)$$

2) Generating edges: $\mathbb{E}$, the set of edges, is derived from equations (34), (35) and (36) of definition 4.

Definition 5. *The set of edges, $\mathbb{E} \subset \mathbb{M}\times\mathbb{M}$, contains all pairs of points sufficiently distant from any obstacles, and which do not yield too sharp bend at departure and arrival.*

$$\begin{aligned} &(\mathsf{M}, \mathsf{M}') \in \mathbb{E} \Leftrightarrow \\ &\overbrace{\forall i\ \mathcal{D}(\overline{\mathsf{M}\mathsf{M}'}, \overline{\mathsf{C}_i\mathsf{D}_i}) \geq \rho_c,}^{} \\ &\mathcal{D}(\overline{\mathsf{M}\mathsf{M}'}, \Delta_\mathsf{A}) \geq 2\rho_c, \\ &\mathcal{D}(\overline{\mathsf{M}\mathsf{M}'}, \Delta_\mathsf{B}) \geq 2\rho_c \\ &\textit{if}\ (\mathsf{M} = \mathsf{A})\ \textit{and}\ \left|\angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{AM}'}) - \theta_A\right| \leq \tfrac{\pi}{2} \\ &\textit{if}\ (\mathsf{M}' = \mathsf{B})\ \textit{and}\ \left|\angle(\overrightarrow{\mathsf{AB}}, \overrightarrow{\mathsf{MB}}) - \theta_B\right| \leq \tfrac{\pi}{2} \end{aligned} \quad (41)$$

3) Generating weights: Let us denote l the iteration number and $\mathcal{W}_l$ the edge-weight map at the l^{th} iteration. At the first iteration, each edge weight is assigned to its length.

$$\mathcal{W}_1((\mathsf{M}, \mathsf{M}')) = \mathsf{M}\mathsf{M}' \quad (42)$$

The average weight is also computed.

$$W = \frac{1}{|\mathbb{E}|}\sum_{e\in\mathbb{E}} \mathcal{W}_1(e) \quad (43)$$

where $|\mathbb{E}|$ is the number of edges in $\mathbb{E}$.

By applying the Dijkstra algorithm on the graph $(\mathbb{M}, \mathbb{E}, \mathcal{W}_l)$, we get $\mathbf{P} \in \mathbb{P}_c$ complying with equations (34), (35) and (36)

of definition 4. Denoting an edge by e, we define $\mathcal{I}(\mathbf{P}, e)$ an indicator of how e is problematic, in that it accounts for how many times e is not compliant with equations (37) and (38).

To ease the technical description of how weights are evolving, we define some path-related attributes. For a given path $\mathbf{P}$ and an edge $\mathbf{e}$, $\mathcal{K}(\mathbf{P}, \mathbf{e})$ is an index, $\overline{\mathcal{K}(\mathbf{P}, \mathbf{e})}$ a line segment and $\overrightarrow{\mathcal{K}(\mathbf{P}, \mathbf{e})}$ a vector. The $\mathcal{K}$-indexes are defined as:

$$\begin{aligned} &\text{Given } \mathbf{P} = [\mathsf{P}_1 \dots \mathsf{P}_K] \text{ and } \mathbf{e} = (\mathsf{M}, \mathsf{M}') \\ &\text{if } \exists k \leq K-1, \mathsf{P}_k = \mathsf{M} \text{ and } \mathsf{P}_{k+1} = \mathsf{M}' \\ &\quad \mathcal{K}(\mathbf{P}, \mathbf{e}) = k \text{ and } \overline{\mathcal{K}(\mathbf{P}, \mathbf{e})} = \overline{\mathsf{P}_k \mathsf{P}_{k+1}} \text{ and} \\ &\quad \overrightarrow{\mathcal{K}(\mathbf{P}, \mathbf{e})} = \overrightarrow{\mathsf{P}_k \mathsf{P}_{k+1}} \\ &\text{else } \mathcal{K}(\mathbf{P}, \mathbf{e}) = 0 \end{aligned} \tag{44}$$

Note that definition 1 ensures the unicity of k when it exists. The following definition makes use of a function denoted $\mathbf{1}$ and mapping propositions into $\{0, 1\}$:

$$\mathbf{1}(\mathcal{P}) = \begin{cases} 1 & \text{if } \mathcal{P} \text{ is true} \\ 0 & \text{if } \mathcal{P} \text{ is false} \end{cases}$$

Definition 6. *$\mathcal{I}(\mathbf{P}, \mathbf{e})$ denotes an integer valued function.*

$$\begin{array}{ccccc} \mathcal{I}: & \mathbb{P}\times\mathbb{E} & \to & \mathbb{N} \\ & ([\mathsf{P}_1 \dots \mathsf{P}_K], \mathbf{e}) & \mapsto & \mathcal{I}(\mathbf{P}, \mathbf{e}) \end{array} \tag{45}$$

with $\mathcal{I}(\mathbf{P}, \mathbf{e}) = 0$ when $\mathcal{K}(\mathbf{P}, \mathbf{e}) = 0$ and otherwise

$$\begin{aligned} &\mathcal{I}(\mathbf{P}, \mathbf{e}) = \\ &\overbrace{\sum_{k \neq \mathcal{K}(\mathbf{P}, \mathbf{e})} \mathbf{1}\left(\mathcal{D}(\overline{\mathcal{K}(\mathbf{P}, \mathbf{e})}, \overline{\mathsf{P}_k \mathsf{P}_{k+1}}) < 2\rho_c\right) + \sum_{|k - \mathcal{K}(\mathbf{P}, \mathbf{e})| \leq 1} \mathbf{1}(|\angle(\overrightarrow{\mathcal{K}(\mathbf{P}, \mathbf{e})}, \overrightarrow{\mathsf{P}_k \mathsf{P}_{k+1}})| \leq \tfrac{\pi}{2})}^{} \end{aligned} \tag{46}$$

Problematic edges have their weights increased as this induces the Dijkstra algorithm to yield paths avoiding such edges. At the l^{th} iteration, the modified edge-weight map is assigned to:

$$\mathcal{W}_l(\mathbf{e}) := \mathcal{W}_{l-1}(\mathbf{e}) + \mathcal{I}(\mathbf{P}_{l-1}, \mathbf{e}) W \tag{47}$$

where $\mathbf{P}_{l-1}$ is the path found at the previous iteration.

4) Generating a valid path: The proposed algorithm generating a valid path (Algorithm 1) is in three parts. The first part builds a graph with its nodes, edges and weights. The second part yields a complete path. The third part tests if this path is valid. And if not, weights of problematic edges are increased, making it less likely that these problematic edges are again traversed. The second and third parts are repeated until a valid path is found.

Algorithm 1 Generating a valid path

INPUT: A initial point, B terminal point, $(\mathsf{C}_i \mathsf{D}_i)$ set of obstacles, ρ_c collision safe-distance
OUTPUT: $\mathbf{P}$ trajectory
INITIALIZATION:
$\mathbb{M}$ is set as in section IV-B1
$\mathbb{E}$ is set by definition 5
$l := 1$
$\mathcal{W}_1$ is set by equation (42)
W is set by equation (43)
loop
  $\forall \mathsf{M} \in \mathbb{M},\ \mathcal{V}(\mathsf{M}) := +\infty$
  for $(\mathsf{M}, \mathsf{M}') \in \mathbb{E}$ **do**
    $\mathcal{V}(\mathsf{M}') := \min\left[\mathcal{V}(\mathsf{M}'), \mathcal{V}(\mathsf{M}) + \mathcal{W}_l((\mathsf{M}, \mathsf{M}'))\right]$
  end for
  $\mathbf{P} := [\mathsf{A}]$
  while $\mathcal{F}(\mathbf{P}) \neq \mathsf{B}$ **do**
    $\widehat{\mathsf{M}}' := \underset{(\mathcal{F}(\mathbf{P}), \mathsf{M}') \in \mathbb{E}}{\arg\min}\ \mathcal{V}(\mathsf{M}')$
    $\mathbf{P} := [\mathbf{P} \widehat{\mathsf{M}}']$
  end while
  if $\mathbf{P} \in \mathbb{P}_v$ **then**
    exit loop and return $\mathbf{P}$
  end if
  $l := l + 1$
  for $\mathbf{e} \in \mathbb{E}$ **do**
    $\mathcal{W}_l(\mathbf{e}) := \mathcal{W}_{l-1}(\mathbf{e}) + \mathcal{I}(\mathbf{P}, \mathbf{e}) W$
  end for
end loop

To simplify the description of the Dijkstra algorithm, we denote by $\mathcal{F}(\mathbf{P})$ the final destination of a path.

$$\mathcal{F}([\mathsf{P}_1 \dots \mathsf{P}_K]) = \mathsf{P}_K \tag{48}$$

Concerning the **loop** on l, it is of great importance for the Dijkstra algorithm to examine edges in a specific order, that of the breadth-first search algorithm.

We do not know the theoretical complexity of algorithm 1. However in all simulations, the path generated at the first use of the Dijkstra algorithm is already valid. Hence, except possibly for some unlikely geometrical configuration of obstacles, the complexity is that of the Dijkstra algorithm. And [37] indicates that it is $O(|V| \ln(|V|) + |E|)$, where $|V|$ and $|E|$ are the numbers of edges and vertices.

Remark 2. *Based on this algorithm and on remark 1, the transmitted information includes* A, B, $\mathsf{C}_i, \mathsf{D}_i$, ρ_c, ρ_R.

V. Simulation Testbed and Experimental Procedures

The proposed path planning scheme is implemented and evaluated in MATLAB and Network Simulator-2 (NS-2), respectively. Applying the proposed technique, MATLAB provides a collision-free flyable trajectory according to the given environmental conditions. MATLAB feeds trajectory coordinates to NS-2 to evaluate the tracking of the designed path and the end-to-end connectivity performance. We consider off-line path planning, but any UAV can lead the entire fleet

dynamically at any desired speed or request hovering. Any UAV within the fleet or node on the ground can be a source of data. We consider rectangular-shaped obstacles of random sizes. The path designing process takes a safe distance margin of 5[1] meters. The experiment considers the simulation area of 2000m×2000m in NS-2 environment with start and destination coordinates mentioned in Table II . Each UAV is equipped with an Omni-directional antenna with communication architecture based on 802.11. Each UAV can transmit within a range of 220m. MATLAB provides coordinate information to be sampled by NS-2 with a sampling frequency of 10Hz and a virtual UAV speed of 10m/s. Figures 6 and 7 present all the possible routes between starting and destination points and the final selected one by the proposed scheme, respectively. Table II enlists the complete simulation parameters.

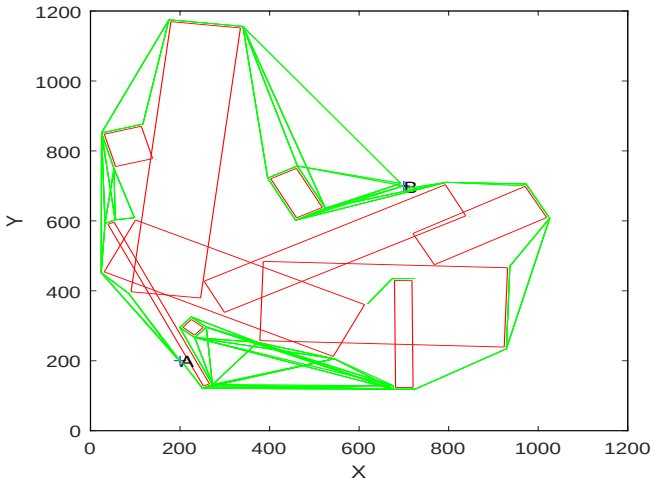


Fig. 6. All possible routes (green connected line segments) between start (lower left corner) and destination point (upper right part of the plot). Obstacles are figured as red rectangles. Both scales are in meters.

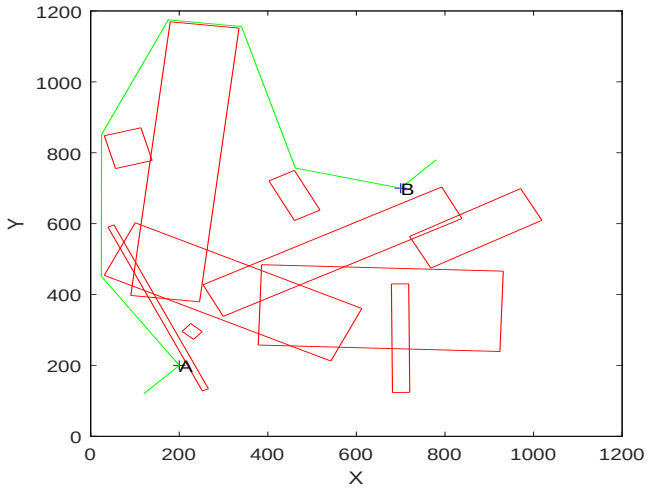


Fig. 7. The selected path (green connected line segments) by the proposed scheme joining A and B. Both scales are in meters.

[1]This value is here taken according to the GPS accuracy as provided by https://www.gps.gov/systems/gps/performance/accuracy/, but it should be increased to include all kind of hazards.

TABLE II
SIMULATION PARAMETERS

Parameter	**Value**
Safe margin distance ρ_C	5 m
Number of obstacles	10
Transmission distance ρ_R	220 m
MAC protocol	802.11
Hello interval	1 s
Data rate	0.8 kbps
Number of UAVs	9
Starting coordinate (m)	(200,200)
Destination coordinate (m)	(700,700)
Max. UAV speed	10 m/s [35]
Ground users	6
Sampling frequency for motion purposes	10 Hz
θ_{A} and θ_{B}	0

A. Performance metrics

We use the following performance parameters to evaluate backhaul connectivity, fleet connectivity, collision avoidance, and scalability features for the proposed approach.

- BS_{con}: To validate the real-time aspect and backhaul connectivity, we plot the number of packets received at BS with respect to time. A zero-slope line part indicates the non-reception of data packets at BS and no connectivity with BS. The non-zero slope line, on the other hand, implies fleet connectivity with BS.
- $T_{\Delta v}$: To demonstrate the fleet connectivity and integrity in response to change in speed requests by the leaders, we plot followers' response time for all such speed change requests. Smaller reaction time for followers indicates quick response to adapt their speed according to the speed instructions given by leader and thus maintaining the desired distance from each other.
- T_{BS}: The average time to reach BS for all data packets.
- D_{min}: To highlight the collision avoidance and evaluate trajectory tracking, we plot, among all UAVs, the minimum UAV-UAV and UAV-obstacles distances for the entire flight time. D_{min} with zero value indicates UAV-UAV or UAV-obstacle collision.
- S_{r-d}: To show the scalability feature in terms of routing, we plot the number of packets received at the BS and packets dropped for the ground users for different leader positions and fleet speed. Zero-slope line area for drop packets curve indicates full coverage to ground users.

VI. Results and Discussion

This section presents experimental results to validate the proposition of the proposed scheme. The first subsection enlists results in which data is generated only by the UAVs within the fleet and for fixed and varying speed of fleet. The second subsection provides simulation results for scenarios wherein only ground users transmit data under the constant and changing speed of the UAV fleet.

A. UAVs as a Source of Data

1) UAV fleet moving with varying speed: In this scenario, the UAV fleet tracks a pre-given trajectory, and each UAV

follows the leader for speed variations. To evaluate D_{min}, Fig. 8 presents, among all UAVs within the fleet, the minimum distance between any two UAVs during the entire simulation time. Before time stamp 200s and after time stamp 500s, the minimal distance is about 15m, that is $2\rho_C\sqrt{2}$, the distance between two consecutive parking slots at departure or at arrival. In between these two time stamps, this minimal distance is increased, yet it remains below 200m which is less than ρ_R (i.e. the threshold above which communication would no longer be guaranteed). Flat line in this figure shows that at least two UAVs are at ground either at the starting or at the landing positions. Similarly, Fig. 9 presents, for all the UAVs and obstacles, the minimum distance between any UAV and obstacle during the entire simulation time. It can be inferred from these two figures that none of the UAV collide with other UAVs or obstacles and this shows the ability to follow some given speed instructions while avoiding all collisions.

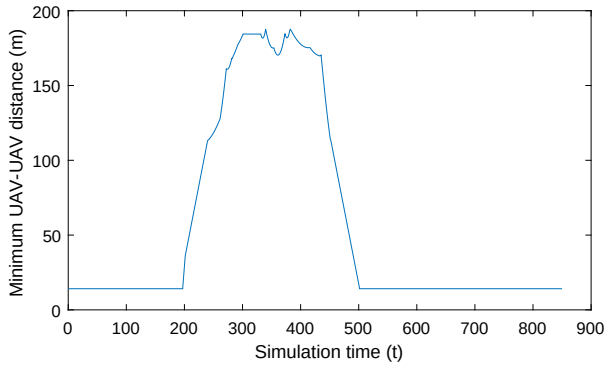


Fig. 8. Minimum UAV-UAV distance with simulation time

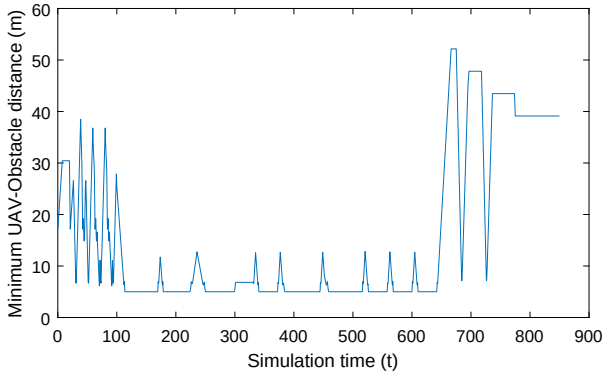


Fig. 9. Minimum UAV-Obstacles distance with simulation time

$T_{\Delta v}$ is evaluated with Fig. 10 showing the reaction time measured for each non-leader UAV to receive the speed change request. This experiment is done considering 9 UAVs and assuming that UAV-1 is the leader sending the speed change request. Using our proposed numbering scheme, UAV-2 is following UAV-1, UAV-3 is following UAV-2 and so on. Requests are delivered using the multi-hop scheme where each non-last UAV transmits the received requests to its follower. This multi-hop scheme explains that throughout all the simulation time, reaction time of UAV-$(n+1)$ is higher than reaction time of UAV-n. As a result, UAV-9 is the latest to receive requests, and its maximal measured delay is of 0.58 s. Such a delay would impact safety considering a speed of 10 m/s and a safe margin of 5 m, if this was not taken into account in the manner of applying speed change requests (each UAVs are assumed to have time synchronization and the requests indicate a common time at which speed is to changed). Simulations with 5 UAVs show also the reaction time between any two neighboring UAVs below 0.5s. It appears that reaction time exhibits at most slow increase with respect to the number of UAVs.

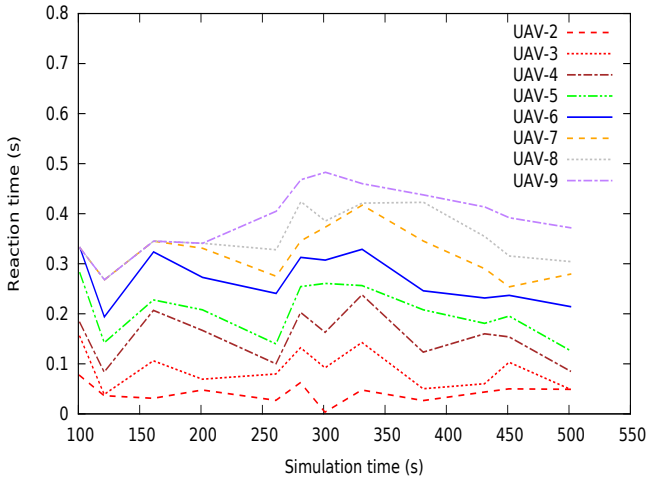


Fig. 10. UAV's reaction time to speed change requests (8 transmitting UAVs)

To evaluate backhaul connectivity with BS (i.e., BS_{con}), Fig. 12 plots the number of packets received at BS with time. In this figure, the non-zero slope line indicates data reception at the BS, while any zero slope line implies no data reception at BS. Consequently, Fig. 12 demonstrates continuous connectivity with BS for the entire fleet, except around 365 s simulation time due to connection handover from one BS to another. Fig. 11 presents the delay incurred by the data packets (i.e., T_{BS}) traveling from the UAVs to BS. As it should be obvious, Fig. 11 shows that increasing the number of transmitting UAVs increases the delay incurred by the data packets. During the start and near the end of the mission, data packets encounter a lower hop count to reach BS due to fewer aerial UAVs resulting in a lower delay. During the middle part of mission time, the number of hops to BS increases due to the increased aerial UAVs leading to higher delay values.

2) UAV fleet moving with fixed speed: This subsection demonstrates the efficacy of the proposed scheme under a fixed UAV tracking speed, in that all UAVs in the fleet maintain a safe distance from each other and with all obstacles.

To evaluate D_{min}, we have computed the UAVs' minimimum distances from any other UAV for a fixed fleet speed of 2, 5, and 10 m/s, and found that it is very similar to Fig. 8. Considering all UAVs and obstacles, we have also computed the minimum UAV-obstacle distance with fixed fleet speed of 2, 5, and 10 m/s, and found it very similar to Fig. 9.

Fig. 13 plots the number of packets received at BS with time to validate end-to-end connectivity (i.e., BS_{con}) for a fleet with a speed of 2, 5, and 10 m/s. The UAV fleet with

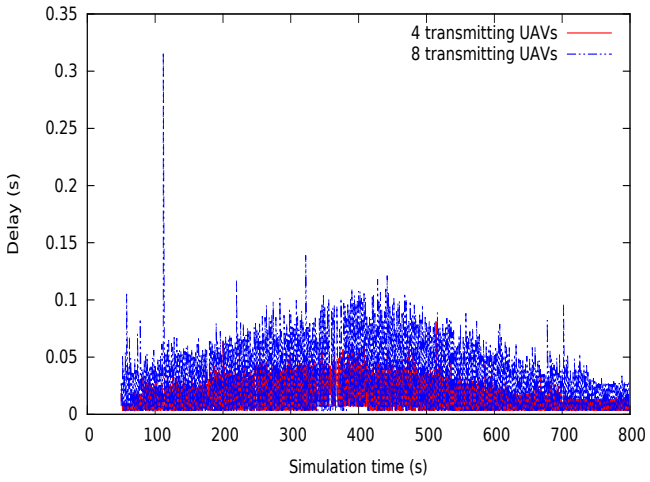


Fig. 11. Data packets delay to reach BS

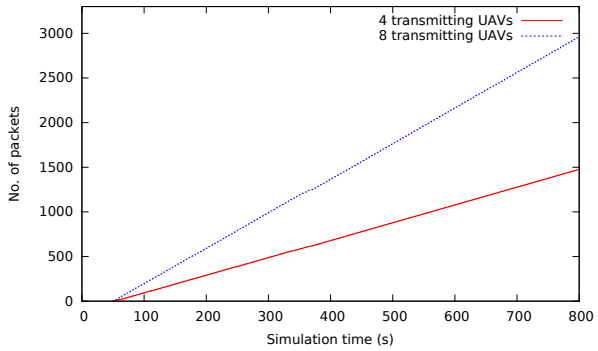


Fig. 12. Number of packets received at BS (end-to-end connectivity check)

2m/s speed takes 1800 seconds to complete its mission and remains connected all the time with BS. Even though, the fleet retains end-to-end connectivity with higher speed values, it lags in comparison to lower speeds due to frequent fleet link changes with BSs.

We collected the delay faced by data packets (i.e., T_{BS}) for UAV fleet speed of 2, 5, and 10m/s. These delays remain similar regardless of the speed. They have the same hat-shape time-dependency than Fig. 11 (messages need much less hops at the beginning and at the end of the experiment when most UAVs are close to A or B).

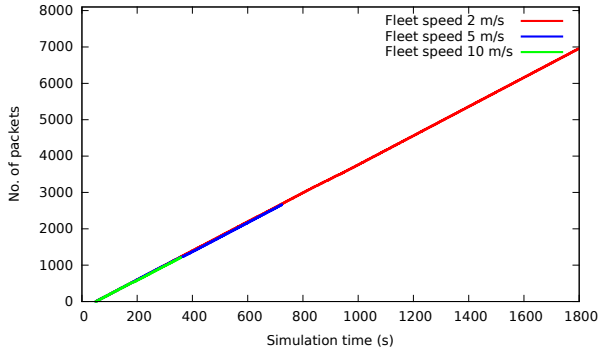


Fig. 13. Number of packets received at BS (end-to-end connectivity check)

B. Ground Users as a Source of Data

In this section, we evaluate the fleet response to connectivity requests arising from uniformly deployed ground users. We also assess the effect of a leader's position within a fleet on the system performance. The first section presents results for varying fleet speed, whereas the second part assesses the system performance for fixed speed values.

1) UAV fleet moving with varying speed: Fig. 14 plots reaction time (i.e., $T_{\Delta v}$) for follower UAVs in response to speed change requests made by the leaders with ground users as a source of data. Fig. 14 shows that the reaction time difference between neighboring UAVs remains modest and eliminates any possible collisions among them. This figure also concludes that the average reaction time for UAVs drops down when a leader lies in the middle of a fleet.

To evaluate scalability feature (i.e., S_{r-d}), Fig. 15 presents the number of packets sent by ground users and received at the BS or dropped. During the start and near the end of the mission, most of the UAVs reside, respectively, at the departure and arrival points, and the moving UAVs cover only part of the path resulting in drop rates. Fig. 15 shows that the number of packets drops to zero as the entire fleet becomes airborne and ensures connectivity to ground users. It is also observable in Fig. 15 that the leader position has a very marginal effect on the packets received and dropped. We collected the data packet delays (i.e., T_{BS}) for the ground users to reach BS and for different leaders' locations within the fleet. And when plotted against time, these delays appear very similar to Fig. 11. Leaders' position plays only a slight role in determining the delay incurred by the data packets.

2) UAV fleet moving with fixed speed: To analyze scalability feature that offers routing services to ground users (i.e., S_{r-d}), Fig. 16 plots the cumulated number of packets received at BS or dropped, with the UAV fleet having speed values of 2, 5, and 10 m/s. UAV fleet completes its mission in the 1800 s with a speed of 2 m/s. BS receives 3211 packets compared to 5250 transmitted packets with a success ratio of 61.16. Furthermore, 1666 packets get dropped. In Fig. 16, for a fleet speed of 5 m/s, BS, with a success ratio of 62.1, receives 1276 out of 2028 send packets having a drop count of 602. Finally, BS successfully receives packets with a 62.7 percentage and ground users dropping 243 out of 930 packets. Like Fig. 15, the shape of the curves indicate that most often, packets are dropped when part of the fleet is at departure or at arrival. We collected the delay incurred (i.e., T_{BS}) by the data packets for ground users to reach BS under 2, 5, and 10 m/s fleet speed. When plotted against time, we get a figure similar to Fig. 11. Delays appear as unaffected by the variation in the fleet speed. Fig. 16 and this last simulation show that a higher speed fleet allows full coverage to deployed ground users during a shorter period of time compared to lower speed fleet wherein the fleet takes more time to reach all deployed ground nodes.

C. Discussion

The proposed connectivity-aware UAVs' path planning is a flexible approach that adapts well according to the application

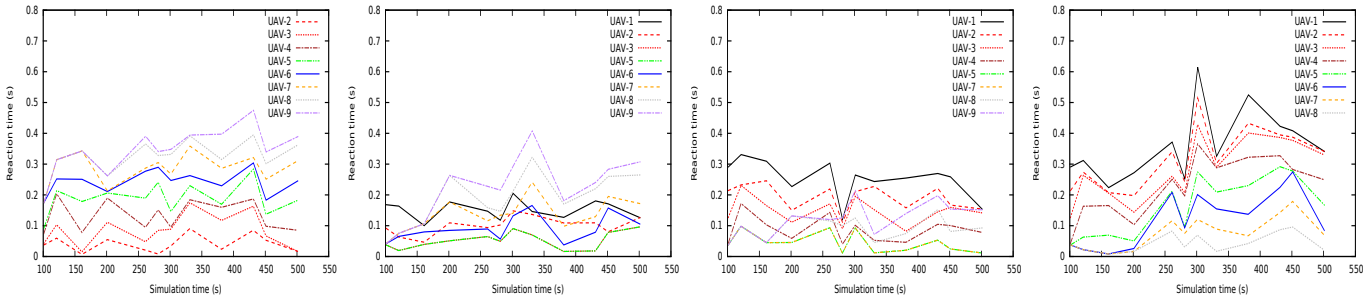


Fig. 14. UAV's reaction time to speed change requests (a) UAV-1 as leader (b) UAV-4 as leader (c) UAV-6 as leader (d) UAV-9 as leader

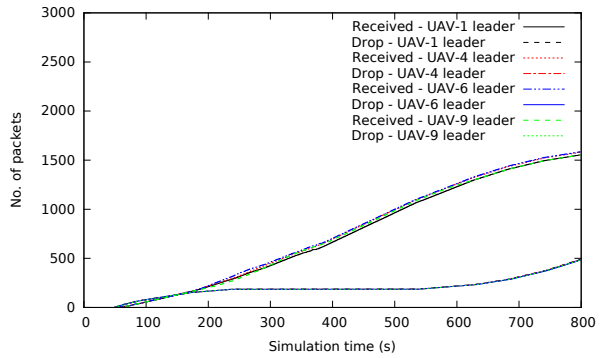


Fig. 15. Number of packets received at BS (end-to-end connectivity check for ground users)

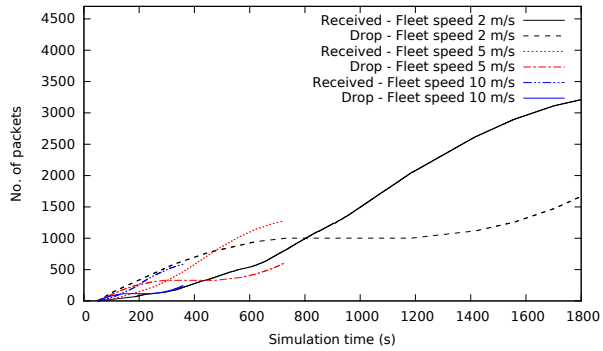


Fig. 16. Number of packets received at BS (end-to-end connectivity check for ground users)

requirements. The proposed scheme can be applied to real-time applications as the entire fleet remains connected while ensuring connectivity with at least one BS. In this regard, Fig. 12 and Fig. 13 validate all-time fleet backhaul connectivity with varying and constant fleet speed. It can be inferred from Fig. 10, and 14 that any UAV can lead the entire fleet at any specific pace with good response time to speed variations commands. This scheme offers a scalability feature in the sense that any new UAV or ground user can become a part of the network without adding any extra overhead. It is observable in Fig. 15 that UAV's fleet provides full network coverage to ground users once all UAVs become air-born and hover over the deployed region. Results in Fig. 8, and 9, depict the minimum distance maintained by the UAVs to each other and the obstacles and hence validate the collision-free path planning aspect of the proposed scheme.

We are well aware of the energy limitations of UAVs and aim to address this issue in our future work. One possible solution could be to have intermediate BSs according to UAVs' energy resources and provide battery charging options at intermediate BSs. We also intend to include UAVs' dynamics into our path planning model to make the system overall more robust against environmental disturbances. In future work, we plan to adapt our proposal to obstacles with varying heights, first by considering the 3D obstacles as an array of 2D maps, each map addressing a specific height at which UAVs can remain.

VII. Conclusion

In this article, we have proposed collision-free UAV fleet path planning constrained by backhaul connectivity. The proposed approach considers offline graph-based path planning with obstacles modeled as line segments. The path planning model includes a safe margin distance parameter to tolerate any uncertainty arising due to environmental disturbances. All the UAVs receive trajectory coordinates before departure, and any UAV can lead the entire fleet dynamically owing to the continuous fleet and backhaul connectivity. Line formation for the fleet acts as a backbone of the network and adds a scalability feature to our scheme in the sense that any additional UAV or ground user can become a part of the fleet. The proposed approach is mathematically proved, implemented in Matlab, and evaluated in the network simulator. The simulation results demonstrate that all UAVs track their designated paths within permissible limits even with the variation in the leader's position, speed variations, and the fleet provides all-time link connectivity to all UAVs within the fleet or outside users.

References

[1] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, A. Tuncer. 2017. "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, pp. 22–28.

[2] Zhe. Z, Yifeng. N, Lincheng. S. 2020. "Adaptive level of autonomy for human-UAVs collaborative surveillance using situated fuzzy cognitive maps," *Chinese Journal of Aeronautics*, vol. 33, pp. 2835–2850.

[3] Kim. H, Ben-Othman. J. 2018. "A collision-free surveillance system using smart UAVs in multi domain IoT," *IEEE communications letters*, vol. 22, pp. 2587–2590.

[4] Ejaz. W, Ahmed. A, Mushtaq. A, Ibnkahla. M. 2020. "Energy-efficient task scheduling and physiological assessment in disaster management using UAV-assisted networks," *Computer Communications*, vol. 155, pp. 150–157.

[5] Munawar. H. S, Ullah. F, Qayyum. S, Khan. S. I, Mojtahedi. M. 2021. "UAVs in Disaster Management: Application of Integrated Aerial Imagery and Convolutional Neural Network for Flood Detection," *Sustainability*, vol. 13, p. 7547.

[6] Erdelj. M, Król. M, Natalizio. E. 2017. "Wireless sensor networks and multi-UAV systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86.

[7] J. Chen, K. Xiao, K. You, X. Qing, F. Ye, Q. Sun. 2021. "Hierarchical Task Assignment Strategy for Heterogeneous Multi-UAV System in Large-Scale Search and Rescue Scenarios," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 7353697.

[8] F. L. Pereira, , 2021. "Optimal Control Problems in Drone Operations for Disaster Search and Rescue," *Procedia Computer Science*, vol. 186, pp. 78–86.

[9] X. Liu, B. Lai, B. Lin, V. C. Leung. 2022. "Joint Communication and Trajectory Optimization for Multi-UAV Enabled Mobile Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13.

[10] Fawaz. W. 2018. "Effect of non-cooperative vehicles on path connectivity in vehicular networks: A theoretical analysis and UAV-based remedy," *Vehicular Communications*, vol. 11, pp. 12–19.

[11] Zhao. H, Wang. H, Wu. W, Wei. J. 2018. "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 2015–2031.

[12] Peng. G, Xia. Y, Zhang. X, Bai. L. 2018. "UAV-aided networks for emergency communications in areas with unevenly distributed users," *Journal of Communications and Information Networks*, vol. 3, pp. 23–32.

[13] F. Syed, S. K. Gupta, S. Hamood Alsamhi, M. Rashid, X. Liu. 2021. "A survey on recent optimal techniques for securing unmanned aerial vehicles applications," *Transactions on Emerging Telecommunications Technologies*, vol. 32, p. e4133.

[14] S. Hayat, E. Yanmaz, C. Bettstetter, T. X. Brown. 2020. "Multi-objective drone path planning for search and rescue with quality-of-service requirements," *Autonomous Robots*, vol. 44, pp. 1183–1198.

[15] 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.

[16] R. Carli, G. Cavone, N. Epicoco, M. Di Ferdinando, P. Scarabaggio, M. Dotoli. 2020, October. "Consensus-Based Algorithms for Controlling Swarms of Unmanned Aerial Vehicles," In *Proceedings of International Conference on Ad-Hoc Networks and Wireless*, Bari, Italy, pp. 84–99.

[17] S. R. Bassolillo, E. D?Amato, I. Notaro, L. Blasi, M. Mattei. 2020. "Decentralized mesh-based model predictive control for swarms of UAVs," *Sensors*, vol. 20, p. 4324.

[18] M. Khachumov, V. Khachumov. 2018. "The model of UAV formation based on the uniform allocation of points on the sphere," In *Proceedings of MATEC Web of Conferences*, Russia, Article ID 03022.

[19] G. Flores-Caballero, A. Rodríguez-Molina, M. Aldape-Pérez, M. G. Villarreal-Cervantes. 2020. "Optimized path-planning in continuous spaces for unmanned aerial vehicles using meta-heuristics," *IEEE Access*, vol. 8, pp. 176774–176788.

[20] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, T. Vicsek. 2018. "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, Article ID eaat3536.

[21] H. Nawaz, H. M. Ali, A. A. Laghari. 2021. "UAV communication networks issues: a review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1349–1369.

[22] M. A. Dehghani, M. B. Menhaj. 2016. "Communication free leader–follower formation control of unmanned aircraft systems," *Robotics and Autonomous Systems*, vol. 80, pp. 69–75.

[23] J. N. Yasin, S. A. S. Mohamed, M. H. Haghbayan, J. Heikkonen, H. Tenhunen, M. M. Yasin, J. Plosila. 2020. "Energy-efficient formation morphing for collision avoidance in a swarm of drones," *IEEE Access*, vol. 8, pp. 170681–170695.

[24] Y. Chen, J. Yu, X. Su, G. Luo. 2015. "Path planning for multi-UAV formation," *Journal of Intelligent & Robotic Systems*, vol. 77, pp. 229–246.

[25] 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.

[26] G. Lee, D. Chwa. 2018. "Decentralized behavior-based formation control of multiple robots considering obstacle avoidance," *Intelligent Service Robotics*, vol. 11, pp. 127–138.

[27] T. M. Cabreira, L. B. Brisolara, P. R. Ferreira Jr. 2019. "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, p. 4.

[28] F. Fabra, W. Zamora, P. Reyes, J. A. Sanguesa, C. T. Calafate, J. C. Cano, P. Manzoni. 2020. "MUSCOP: Mission-based UAV swarm coordination protocol," *IEEE Access*, vol. 8, pp. 72498–72511.

[29] 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.

[30] 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.

[31] 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.

[32] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, D. Wu. 2017. "Offline and online search: UAV multiobjective path planning under dynamic urban environment," *IEEE Internet of Things Journal*, vol. 5, pp. 546–558.

[33] J. Yoon, Y. Jin, N. Batsoyol, H. Lee. 2017, March. "Adaptive path planning of UAVs for delivering delay-sensitive information to ad-hoc nodes," In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, USA, pp. 247–264.

[34] N. Toorchi, F. Hu, E. S. Bentley, S. Kumar. 2020. "Skeleton-Based Swarm Routing (SSR): Intelligent Smooth Routing for Dynamic UAV Networks," *IEEE Access*, vol. 9, pp. 1286–1303.

[35] 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.

[36] Z. Jiang, X. Wang, and J. Yang. Distributed line formation control in swarm robots. In *2018 IEEE International Conference on Information and Automation (ICIA)*, pages 636–641, 2018.

[37] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[38] Yiguo Yang, Liefa Liao, Hong Yang, and Shuai Li. An optimal control strategy for multi-uavs target tracking and cooperative competition. *IEEE/CAA Journal of Automatica Sinica*, 8(12):1931–1947, 2021.

[39] Zongyu Zuo, Cunjia Liu, Qing-Long Han, and Jiawei Song. Unmanned aerial vehicles: Control methods and future challenges. *IEEE/CAA Journal of Automatica Sinica*, 9(4):601–614, 2022.

[40] Daqian Liu, Weidong Bao, Xiaomin Zhu, Bowen Fei, Tong Men, and Zhenliang Xiao. Cooperative path optimization for multiple uavs surveillance in uncertain environment. *IEEE Internet of Things Journal*, 9(13):10676–10692, 2022.

[41] Z. Wu, Z. Yang, Ch. Yang, J. Lin, Y. Liu, and X. Chen. Joint deployment and trajectory optimization in uav-assisted vehicular edge computing networks. *Journal of Communications and Networks*, 24(1):47–58, 2022.

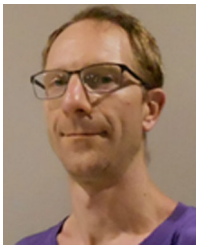
[42] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov. Path following for small unmanned aerial vehicles using l1 adaptive augmentation of commercial autopilots. *Journal of Guidance, Control, and Dynamics*, 33(2):550–564, 2010.

[43] E. Frew, D. Lawrence, and S. Morris. Coordinated standoff tracking of moving targets using lyapunov guidance vector fields. *Journal of Guidance, Control, and Dynamics*, 31(2):290–306, 2008.

[44] Nouman Bashir, Saadi Boudjit, and Gabriel Dauphin. Supplementary materials, 2023. https://hal.science/hal-04081656.

Nouman Bashir is currently a Wireless Firmware Engineer at NXP Semiconductors (Caen, France). He obtained his Ph.D. from the L2TI laboratory of Université Sorbonne Paris Nord in 2022. He received his M.Sc. degree in Electronic Engineering from GIK Institute of Engineering Sciences and Technology, Pakistan, in 2017. 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.