A 3D Mobility Model for Autonomous Swarms of Collaborative UAVs

Ema Falomir, Serge Chaumette, Gilles Guerrini

A 3D Mobility Model for Autonomous Swarms of Collaborative UAVs

Ema Falomir[1,2], Serge Chaumette[1] and Gilles Guerrini[2]

***Abstract*— Collaboration between several Unmanned Aerial Vehicles (UAVs) can produce high-quality results in numerous missions, including surveillance, search and rescue, tracking or identification. Such a combination of collaborative UAVs is referred to as a swarm. These several platforms enhance the global system capabilities by supporting some form of resilience and by increasing the number and/or the variety of the embedded sensors. Furthermore, several UAVs organized in a swarm can (should the ground control station support this) be considered as a single entity from an operator point-of-view. We aim at using such swarms in complex and unknown environments, and in the long term, allow compact flights.**

Dynamic path planning computation for each UAV is a major task to perform their mission. To define this path planning, we have implemented a three-dimensional (3D) mobility model for swarms of UAVs using both the Artificial Potential Fields (APF) principle and a global path planning method. In our model, the collaboration between the platforms is made by sharing information about the detected obstacles. To provide a significant validation of our mobility model, we have simulated real-world environments and real-world sensors characteristics, using the OMNeT++ network simulator.

I. Introduction

Unmanned Aerial Vehicles (UAVs) are major tools today to perform numerous tasks in a wide range of operations. Indeed, the variety of tasks one would like to achieve increases quicker than the capacity of a single UAV. Therefore swarming became, quite recently, an important field of interest and research [16] [7]. Among the many advantages offered by swarms are: persistent flight, multi-sensor capabilities, reorganization of sensors, resilience and possible replication of information.

The UAVs of a swarm have to communicate with each other to collaborate. Then, in addition to the sensors they embed, the platforms are equipped with wireless network features that allow them to form a Flying Ad hoc Network (FANET) [4].

To perform a surveillance mission, the UAVs have to be able to move from a geographical area to another. This is the problematic that we address here.

A large number of studies that deal with swarms of UAVs consider area coverage missions (keeping connectivity constraints in mind [19], [15], [23]) but most of them consider obstacle-free environments. In our work, we consider the case of urban-like areas. In such configurations, the UAVs cannot always fly above buildings because in this case, some areas would be hidden to the sensors due to the buildings. Consequently, obstacle avoidance is a task of prime importance.

In this paper we present the 3-dimensional mobility model for autonomous swarms of UAVs that we have created. It is based on the Artificial Potential Fields (APF) principle and additionally includes a global path planning process. This model called 3DGPeach (3-Dimensional Global Peach) extends one of our previous model which we have called Peach and which is described in [10].

This paper is organized as follows. Section II presents related work on multi-UAVs mobility strategies and path planning using APF. In section III, we describe the Peach mobility model and the new extension 3DGPeach. Section IV focuses on simulations and results, and we compare our results to those of other mobility models for swarms of UAVs. Finally, conclusion and future work are addressed in section V.

II. Related Work

Path planning methods in the robotic field have been extensively studied and described. The following sections focus on methods designed for FANETs and on mobility strategies based on the the APF principle (introduced in [11]).

A. *Mobility Strategies for FANETs*

Mobile Ad hoc NETworks (MANETs) are continuously self-configuring and infrastructure-less networks composed of mobile devices connected wirelessly [4]. A FANET is a type of MANET composed of UAVs. When several UAVs have to collectively fulfill a task, close collaboration between the platforms can enhance the performance of the system. Due to the substantial number of existing path planning methods, we have selected here the most studied methods and those which are the closest to our work. Nevertheless, none of them was adapted to our study because the constraints differ. Indeed, we consider path calculation in unknown environments, with limited computation power and require fully autonomous UAVs.

Using multiple UAVs can be very efficient for surveillance missions thanks to collaboration between the platforms because this makes it possible to support a large number of capabilities (multi-sensor, multi-platform, multi-modal capabilities). However, since the on board communication systems have a limited range, a compromise has to be found between two adversary criteria: maximizing area coverage and preserving network connectivity [19]. A method achieving these objectives, based on chaotic dynamics and ant

[1] Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France {ema.falomir-bagdassarian, serge.chaumette}@labri.fr

[2] Thales DMS France, F-33700 Mérignac, France {ema.falomir, gilles.guerrini}@fr.thalesgroup.com

colony optimization (ACO), has been presented by Rosalie *et al.* [19] and Bouvry *et al.* [6].

Boskovic and Moshtagh [5] proposed a global system including mission planning, using evolutionary algorithms, dedicated to military operations. Their solution is distributed and robust to communication latency and loss. Furthermore, it uses a real-time learning algorithm and provides dynamic mission re-planning.

Peng *et al.* [17] enhanced the Rapidly-exploring Random Tree (RRT) based path planning principle allowing cooperative UAVs to fulfill a search mission.

Li *et al.* recently proposed a Particle Swarm Optimization (PSO) mobility model designed for FANETs [13]. This PSO method takes into account the characteristics of the UAVs, including kinematic and dynamic constraints. Their method generates velocities and waypoints for each UAV, which are adjusted to avoid collision with neighbors.

Li also proposed a global method [12] in order to search a single static target, including a collaborative mobility model. In their work, there is no obstacle in the Area of Interest (AoI). Information on the presence of the target in each cell of the discretized environment are shared. At each iteration, each UAV analyzes its environment, updates its own map, shares it with the other UAVs, updates its own map by merging it with the information received from the other UAVs and chooses its next flying direction. For this last step, each UAV has two choices: to move into one of the four adjacent cells or not to move. The choice criteria is to go towards the cell with the highest probability of target presence.

B. *APF-based methods*

The original APF method was introduced by Khatib in 1986 [11]. The APFs are composed of two kinds of fields: an attractive field directed towards a target point and a repulsive field for each obstacle. The moving direction of a robot (UAV) is then computed as the negative gradient of the resulting APF.

Numerous studies have been carried out on APF methods. First, because they are easy to implement and have low computational cost [24]. Second, because Khatib's method presents some weaknesses, and can then be improved, as local minima (which induce the robot to be trapped in), Goal Non Reachable with Obstacle Nearby (GNRON) problem or impossibility to reach the goal when it is aligned with an obstacle.

A solution to avoid local minima has been proposed by Liu and Zhao [14]. It is composed of two steps. First, if there is a superposition in the influence area of several obstacles, they are considered as a single larger obstacle. This step reduces the number of local minima. Second, if a UAV reaches such a minima anyway, it creates a temporary waypoint allowing it to escape.

In 1990, Connolly and Burns [8] proposed an approach combining APFs and harmonic functions (solutions of Laplace's Equation), as a global path planning method allowing obstacle avoidance. This method does not suffer from local minima but has a high computational cost and numerical precision issues. More recently, this principle was improved by Wray *et al.* [22] who proposed a log-space algorithm which he tested with a humanoid robot.

Very close to our approach, Sun *et al.* [20] propose a collision avoidance model for cooperative UAVs based on improved APFs. One can quote their local solution to avoid the jitter problem. Furthermore, their definition of potential fields removes local minima, the GNRON problem and the impossibility to move between close obstacles. Their model performs well in 3D to allow several UAVs to reach a common target point. Nevertheless, the description of the hypothesis they make is not precise enough to appreciate the relevance of their results.

III. OUR CONTRIBUTION: THE 3DGPEACH MOBILITY MODEL

In this section we summarize our previous mobility model, which will be called Peach. It is described in details in [10]. Then we will describe 3DGPeach, an extension of Peach which includes 3-dimensional environments and trajectories computation additionally to global path planning. Both mobility models were created for autonomous UAVs combined as a swarm, which collaborate by sharing their knowledge of the environment. Furthermore, we consider the UAVs as holonomic vehicles allowed to hover, and kinematics constraints are not taken into account.

A. *Common Features of Peach and 3DGPeach Models*

In the framework of our simulation, the UAVs that we consider evolve in a bounded rectangular area containing obstacles they are not aware of. The environment is meshed by square cells of equal size (this size is defined by the user), which are at least as large as a UAV so as to avoid collisions. The flying time is also discretized in steps of one second, empirically, but this value is representative according to the maximum speed of the UAVs which is arbitrary set at $10m.s^{-1}$. The distance traveled by a UAV in one iteration at maximal speed always has to be smaller than its obstacle sensor range. At each iteration, each UAV analyzes the environment thanks to its sensor(s) and eventually detects obstacles around it. Then, it calculates its next move in any possible direction without any kinematic constraint.

According to the APF principle, the goal point of a UAV is associated to the lowest potential, and the obstacles (fix or mobile -other UAVs-) have high potentials. A UAV calculates its path by choosing points with potentials as low as possible, which allow obstacle avoidance.

Each UAVs in turn share its vision of the environment with its neighbours (see details in section IV-A). As soon as a UAV receives these information, it updates its own vision, and considers all received information in the same way to calculate its movements.

B. *Notations*

The UAVs evolve in a meshed environment. All cells are of equal size. If the environment is 2D, the cells are squares.

In 3D, cells are cuboids with square horizontal faces and their height can be set independently of the other dimensions.

Hence we consider the environment as a table in 2 or 3 dimensions, represented by a table noted: $(c_{i,j,k})_{1\leq i\leq \mathrm{lin}, 1\leq j\leq \mathrm{col}, 0\leq k\leq \mathrm{lev}}$ where $\mathrm{lin}, \mathrm{col} \in \mathbb{N}\backslash\{0\}$ and $\mathrm{lev} \in \mathbb{N}$, respectively represent the number of lines, columns and number of altitude levels in the discretized environment. The cell containing the target point is noted g, and its components are noted g_i, g_j, g_k. When the environment contains only 2 dimensions, the third one is omitted.

C. Peach Mobility Model

The Peach mobility model follows the original APF method by considering several potential fields with high potentials for the obstacles and low potentials for the target point. The field related to the goal point is defined at the beginning of the mission and its impact in the cell (i, j) it is defined as follows:

$$\mathrm{goal}(i, j) = \sqrt{(i - g_i)^2 + (j - g_j)^2} \tag{1}$$

The potential related to an obstacle discovered in a cell (o_i, o_j) is defined as follows:

$$\forall c_{i,j} | (i, j) \in [\![o_i - 1, o_i + 1]\!] \times [\![o_j - 1, o_j + 1]\!]$$
$$p^o_{ij} = \begin{cases} \alpha & \text{if } (i, j) = (o_i, o_j) \\ \frac{\alpha}{2} & \text{else} \end{cases} \tag{2}$$

where α is a fixed high value, arbitrary set in the order of the target potential at distance 10 of the target. If a neighbour (UAV) is detected, the potential is equal to α in all the adjacent cells.

As defined by the model, the UAVs always try to go towards lower potentials. They evolve in a discrete environment, and can move at each iteration to one of the eight cells adjacent to its current location, or remain in the same cell. Then, at each iteration, nine cells are studied. The principle is presented in algorithm 1, while the calculations are given in our previous paper.

This mobility model uses three matrices. The first one, noted pot stores the APF described above. The second one, noted avoid, is re initialized at each iteration and stores temporary APFs related to neighbors to and to the anticipation of obstacle avoidance (see definition in [10], section IV-F). To choose its next move, a UAV compares the sum of the potentials stored in these two matrices for each adjacent cell and go in the one with the smallest potential (or do not move if it is in the cell with the smallest potential). Finally, the third matrix, obs, contains the cell status :

- -1 there is no information about the cell;
- 0 it does not contain an obstacle and is safe;
- 1 it is closed to an obstacle or has already been revisited;
- 2 it contains an obstacle;

To optimize communication between UAVs this matrix is the only one shared between them. As soon as a UAV receives information from its neighbours, it updates its own knowledge of the environment, *i.e.* its obs matrix. One can note that unlike the first version of the Peach model (presented in our previous paper), the status "path already used by a UAV to reach the target" does not appear anymore. This is because now the UAVs can have different goals; this information is thus not relevant anymore.

In this model, we introduced an original method that allows to anticipate obstacle avoidance and to prevent, in most cases, the local minima problem. Its principle is described in algorithm 1.

Data: UAV cell (i, j), goal cell, pot and obs matrices
$\mathrm{avoid} \leftarrow 0_{\mathrm{lin,columnsNb}}$
if *new fix obstacles are detected* **then**
  Update obs and pot matrices
foreach *obstacle stored in* obs **do**
  Update avoid matrix
if *neighbours are detected* **then**
  Update avoid matrix
Calculate the potential of the 9 considered cells
Store the cell with minimum potential in c_{minPot}
if *the UAV is on the cell with smallest potential* **then**
  Increase $\mathrm{pot}_{i,j}$;   `// In order to delete the local minimum`
Next cell $\leftarrow c_{\mathrm{minPot}}$

Algorithm 1: Sequence of events during an iteration for a UAV, with the Peach model. Calculations are detailed in [10].

D. 3DGPeach Mobility Model

In this extension of Peach, we kept the same definitions for potential and a UAV will still go towards lower potentials. The two main improvements are the following: 1) the UAVs can evolve in 3D and 2) they calculate a global path towards the target with as few waypoints as possible.

1) 3D: In order to move in three dimensions, the environment is meshed as cuboids and the matrices supporting the potentials thus also have 3 dimensions. Definitions of potentials are an extension in 3D of those defined in Peach. The user of the simulation can choose independently the horizontal and vertical cells sizes (but the bases of the cuboids are square).

2) Global Path: When moves are limited to the surrounding cells, only eight directions can be taken in 2D and 26 in 3D. But, in order to shorten the paths, it is necessary to authorize more directions. This is done as follows. In this model, the UAVs compute one or several waypoints (WPs) at each iteration. The calculation method depends on the location of the UAV and on the characteristics of the obstacles (width, height, shape...). The resulting WPs are not necessarily in adjacent cells and the UAVs are thus allowed to move in any direction (see Fig. 1). The list of WPs calculated by a UAV constitutes its path. This list of WPs is updated when obstacles are detected on the path, *i.e.* between WPs.

The principle of our model is described in algorithms 2 and 3, and details are given in the following paragraphs.

Data: UAV cell, goal cell, pot and obs matrices
if *new fix obstacles are detected* **then**
  Update obs and pot matrices
if *obstacles (fix or neighbour) were detected on the path* **then**
  Calculate new WPs *(see algorithm 3)*
if *the first WP is reachable in one move* **then**
  Reach the first WP
  Delete the first WP of the path
else
  Go toward the next WP at maximal speed

Algorithm 2: Sequence of events during an iteration for a UAV, with the 3DGPeach model.

Data: UAV cell c_{uav}, goal cell c_{goal}, pot and obs matrices
avoid $\leftarrow 0_{\text{lin,col,lev}}$
foreach *known obstacle **and** neighbour* **do**
  update avoid matrix
$\text{WP}_0 = c_{\text{uav}}$
$\alpha \leftarrow 0$
while $\text{WP}_\alpha \neq g$ **do**
  $\alpha \leftarrow \alpha + 1$
  Calculate the potential of the 27 possible cells to $\text{WP}_{\alpha-1}$
  Store the cell with minimum potential in c_{minPot}
  if *the UAV is on the cell with smallest potential* **then**
    Increase $\text{pot}_{\text{WP}_\alpha}$; `// In order to delete the local minimum`
  $\text{WP}_\alpha \leftarrow c_{\text{minPot}}$
  if *the path is safe between* $\text{WP}_{\alpha-2}$ *and* WP_α ***and*** $\alpha > 1$ **then**
    Delete $\text{WP}_{\alpha-1}$

Algorithm 3: Path calculation.

When a UAV is close to an obstacle, it is not always possible to find safe waypoints far away from this UAV. For this reason, we have to consider 3 modes depending on the environment of the UAV, in order to optimize the path:

1) No obstacle detected in the direction of the target
2) Obstacle detected in the direction of the target
3) UAV in a U-shape obstacle (dead-end)

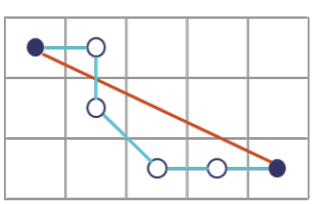


Fig. 1: Interest of spaced out waypoints. The waypoints, centered in square cells, are represented by circles.

When a UAV changes of case, the path is reinitialized and recalculated. Fig.2 illustrates the path calculation depending on the UAV environment presented in 2D for the sake of readability, but the process is identical in 3D.

In the first mode, the UAV goes at maximal speed towards the target (see Fig. 2(a), 2(f), 2(g) and 2(h)). While a UAV is in this situation, it uses the shortest path to reach the target: straight on toward it. The path is then composed of one single waypoint: the cell closest to the target in the range of the UAV sensor.

In the second mode, the UAV cannot go directly towards the target because of obstacles or neighbors. Then the Peach algorithm is ran iteratively using the UAV current knowledge of the environment to add waypoints to the path until the target is reached (see Fig. 2(b), 2(c) and 2(d)).

Finally, as a UAV sensor range is limited in range, a UAV can first consider that two obstacles are around it, and later discover that it is actually surrounded by a single building (or several buildings too close to offer a passage between them). In this mode, the UAV is in a dead-end: it will retrace its steps until the exit of the dead-end and significantly increase the potential (in the pot matrix) of the cells inside this obstacle in order to avoid them next time. In this mode, the path is composed of the cells already visited by the UAV inside the dead-end, from the most recent to the oldest.

As noticed before, only the first waypoint is used to make the move decision at each iteration, independently of its distance from the UAV. Then, in order to have flights that are as short as possible, as few waypoints as possible should be considered, and they should be as far as possible from the UAV (as shown on Fig.). We thus created a function to delete useless waypoints.

Such deletion of waypoints has been performed on Fig. 2(b), 2(c) and 2(d) and made it possible to reduce the necessary number of iterations to reach the target.

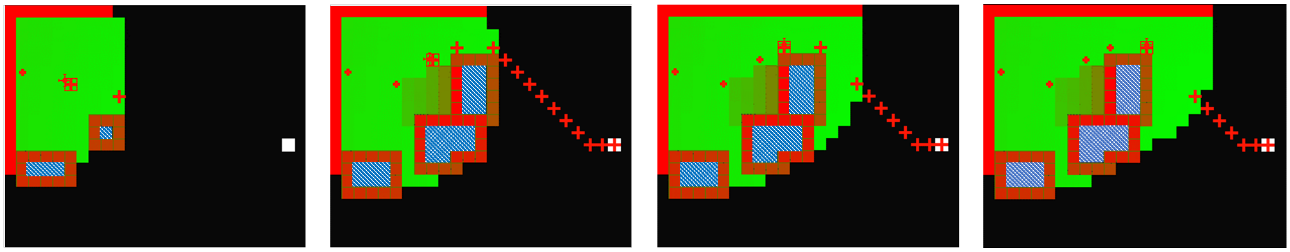
IV. Simulations and Results

So as to be able to run experiments with real UAVs using our mobility model in our future work, we studied the best pieces of equipment that make sense in our context, and simulated them.

A. Embedded Equipments

Each UAV embeds the required equipment to ensure communication with its neighbors and detection of obstacles.

1) Obstacles Detection: The UAV "DJI Mavic Air" seems to have the most advanced collision avoidance system among the small UAVs that are available on the market today [2], [18], [9]. It is equipped with several sensors [1], [9] including a stereo vision system, with a detection range of up to 24 meters and with limited field of view (horizontal: 50°, vertical ±19°). While a Mavic Air uses its detect and avoid system to follow a given direction, our UAVs have to discover their environment to calculate a path. Then, they have to be aware of obstacles in any direction. This is why we chose to simulate an obstacle detection system of 24-meter-range, corresponding to two stereo cameras (one on

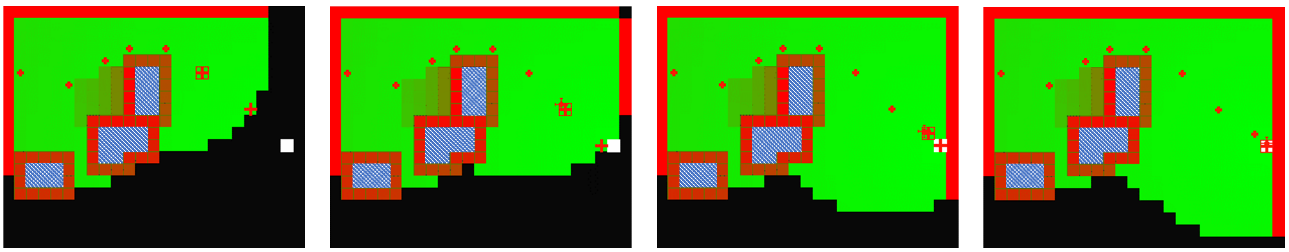


(a) Mode 1: straight on towards the target.

(b) Obstacle detected on the way, passing to mode 2, calculation of a new path and suppression of useless waypoints.

(c) No new obstacle and not possible to go straight on towards the target, following the path and deletion of waypoints.

(d) No new obstacle and not possible to go straight on towards the target, following the path and deletion of waypoints.



(e) Passing to mode 1, path cleaned and creation of a single waypoint.

(f) Mode 1.

(g) Mode 1.

(h) Mode 1.

Fig. 2: Moves of a UAV in an unknown environment iteration by iteration. Small crosses represent the cells where the UAV has been, and the large ones represent the waypoints. The target cell is the white one, cells in black are unknown, the colored ones have been discovered. The color of each known cell corresponds to its potential (green: low potential, red: high, dark blue hatched: highest corresponding to obstacles).

the top and one on the bottom of the UAV) each mounted on a gimbal supporting rotations. The UAVs then acquire their environment in a sphere centered on themselves, as shown on Fig. 3.

2) Communication Module: In our model, the swarm is collaborative because its UAVs share information about the environment. Along with the Peach model, the UAVs carry XBee communication modules, which are particularly adapted for small UAVs application because they are small, have low energy consumption, have light weight and are easily customizable [21].

B. Simulation Setup

In order to simulate realistic communication characteristics, we chose the highly customizable network simulator OMNeT++ [3]. The main parameters used are the same

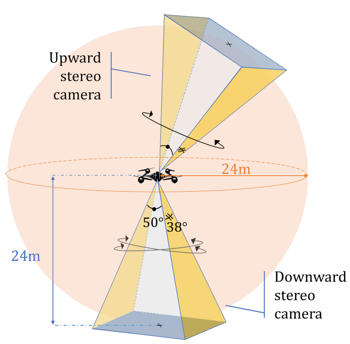


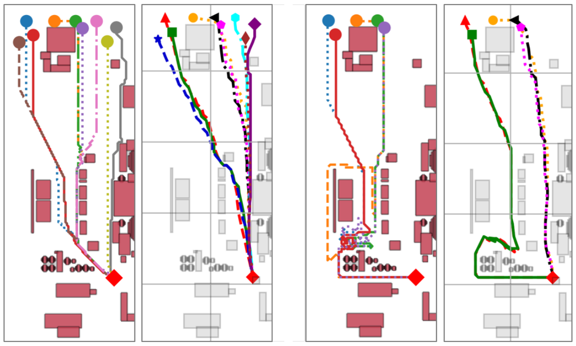
Fig. 3: Embedded sensors allowing obstacle detection.

as in our previous work [10] and are given in appendix. Concerning the sensor simulation, we suppose that the UAVs have information on the presence of obstacles within their sensors ranges. For all the simulations, we make the following assumptions:

- At the beginning of the mission, all the UAVs are within the AoI (Area of Interest), they know how many they are in the swarm, and they know the limits of the AoI.
- The only moving obstacles within the AoI are the UAVs composing the swarm.
- Calculation power and batteries are sufficient to perform the whole mission.
- Each UAV knows with sufficient precision its own location and its target point location (inside the AoI).
- Each UAV is equipped with a XBee communication module and with sensors allowing obstacle detection in a range of 24 meters.
- The maximum speed of the UAVs is $10 m.s^{-1}$ and they are not submitted to kinematics constraints.

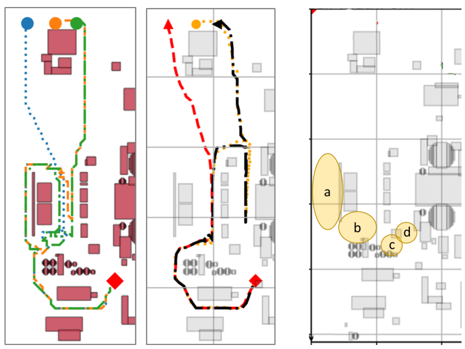
Xbee data rate is up to 250 $kbits.s^{-1}$, half-duplex. Then, if we want the UAVs to communicate every two seconds, the maximal possible volume of the exchanged matrix is 500kbit. One single matrix is exchanged between the platforms, containing as many elements as cells in the discretized environment, and each element contains an integer coded on 2 bits (because each element of the obs matrix can take one of four values).

In our use case, the whole AoI is 500m large, 500m long, and the UAVs can fly between 4m (to avoid people and the



(a) 9-UAV swarm, horizontal discretization of 2m.

(b) 3-UAV swarm, horizontal discretization of 4m.



(c) 3-UAV swarm, horizontal discretization of 6m.

(d) Areas legend.

Fig. 4: Trajectories followed by autonomous swarms of UAVs. For each subfigure, on the the left, calculations made with Peach, on the right, calculations made with 3DGPeach

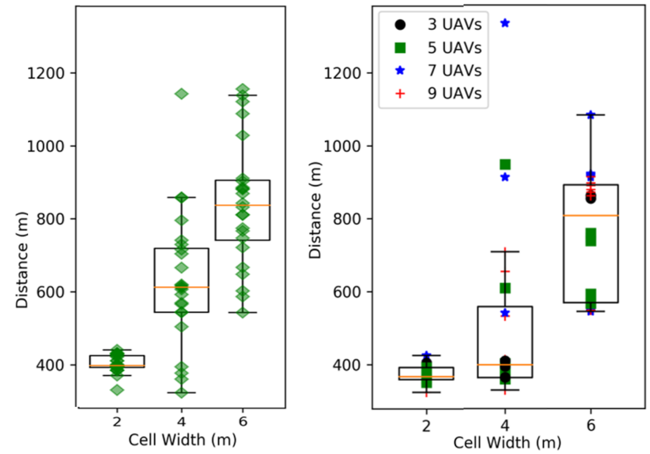
vehicles) and 24m (camera range) of altitude. So a meshing composed of cubes with edge lengths of 4m is adapted. Each UAV has a unique ID. Every 2 seconds, each UAV sends its vision of the environment; this is achieved by increasing ID. Depending on the dimension of the AoI, the size of the cells can be set accordingly by the user.

C. Comparison with Peach

The first step to evaluate our model was to compare the trajectories computed with 3DGPeach to those computed with Peach, in a use case representative of our study: we chose to simulate the buildings of a power station[1]. Three different cell widths are tested: 2, 4 and 6m. Simulations were performed with swarms of 3, 5, 7 and 9 UAVs. Fig. 4(a) to 4(c) show some examples of the trajectories followed by the UAVs, in two dimensions, with both mobility models. Fig. 4(d) will be used to explain the differences in the trajectories in the following paragraphs.

More precisely, the UAVs paths computed with a cell width of 2 meters are clearly shorter than those with larger

[1]From the French land register: `https://www.geoportail.gouv.fr/carte?c=-0.6601155939472929,45.12124680059938&z=17&l0={ORTHOIMAGERY}.{ORTHOPHOTOS}::{GEOPORTAIL}:{OGC}:{WMTS}(1)&l1={CADASTRALPARCELS}.{PARCELS}::{GEOPORTAIL}:{OGC}:{WMTS}(1)&permalink=yes`



(a) Results for Peach mobility model. The 72 UAVs reached the target.

(b) Results for 3DGPeach mobility model. 3 UAVs out of 72 did not reach the target.

Fig. 5: Superposition for a given cell width (2, 4 and 6m) of the distances traveled until the goal by each individual UAV in swarms of 3, 5, 7 and 9 vehicles (so a total of 72 UAVs) in the 2-dimensional environment represented in Fig.4.

cells (see Fig. 5). Indeed, this cell width allow to cross areas c and d (defined on Fig. 4(d)).

With a cell width of 4 meters, it is not possible to cross area c anymore. In this case, the median traveled distance is significantly shorter with 3DGPeach than with Peach. The substantial difference with Peach is due to two factors. First, when several UAVs are in area b, they needed a large number of moves to escape this complex area while avoiding collisions with the neighbors. This is because the move decision during one iteration is taken independently from the previous and next ones. Second, when many UAVs were close to area c, some escaped this passage and preferred area d to avoid collisions. These two solutions increase the traveled distance. 3DGPeach solves both problems and most of the UAVs easily escape from area c.

Finally, a cell width of 6 meters forbids the passage to area d in addition to area c. As a result, the traveled distance with both models are substantially longer. In these conditions, the shortest path is a bit smaller than 600m (distance traveled by the red UAV in Fig. 4(c)). With 3DGPeach more than 25% of the UAVs paths were shorter than 600m. The UAVs whose path measured approximately 800 meters had a trajectory close to the orange and black UAVs shown on Fig. 4(c). For all these platforms, the path cannot be shorter because of their limited vision of the environment. Finally, one can notice that the longest traveled distance with cell width of 6 meters are followed by UAVs in the largest swarms. This is due to the collision avoidance system between the platforms which reduces the optimization of the path, by decreasing the deletion of waypoints.

These simulations show that the smaller the cells, the shorter the paths. Nevertheless, UAVs wingspan is approx-

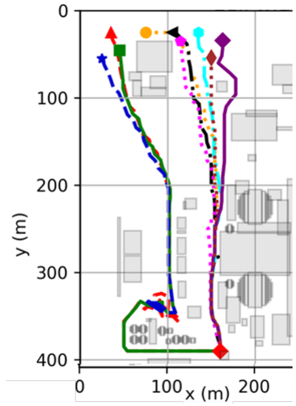


Fig. 6: Trajectories followed by 9 autonomous UAVs evolving in swarm in 2 dimensions, with a discretization of 4m.

imately 50cm so it would not make sense to use thinner discretization. Furthermore, the mobility model is based on three matrices representing the environment (see section III) which contain as many elements as cells. Then, the smaller the cells the larger the matrices. Because one of these three matrices is exchanged between the platforms and because the XBee data rate is limited, a very thin discretization would not allow collaboration between the platforms.

Nevertheless, on these simulations, 3DGPeach has a weakness: in 2 simulations out of 12, some UAVs remained in local minima. Indeed, they alternated with the mode 1 (no obstacle detected in the direction of the target, presented in section III-D.2) and a path calculation. Fig. 6 shows such a simulation, were the dark blue UAV never reaches the target point.

To conclude, our mobility model performs well in a complex environment in 2 dimensions, especially with a thin discretization.

D. Comparison with Sun et al.'s Model

As explained in section II-B, the work by Sun *et al.* [20] is close to ours. Then, created a simulation very close to their testing environment to compare the trajectories calculated by their model to our model (see Fig. 6 of their paper [20]). From their figure, we suppose that their testing environment is 110m long and 70m high. In our simulations we arbitrarily chose a width of 15m and tested various vertical discretizations. Fig. 7 shows a 3D view of a 6-UAV swarm trajectory in this environment and Fig. 8 shows sectional views with different discretizations.

As for the results of Sun *et al.*, the UAVs use different paths to reach the target point. One can note that the trajectories are smooth except for few UAVs which shift from their shortest trajectories to avoid collisions with others, especially when close to obstacles. Furthermore, the distance traveled in one iteration is much larger in obstacle-free environment than in thin passages, because only few waypoints can be deleted there.

To conclude, our model performs well in this 3-dimensional complex environment, even in environments

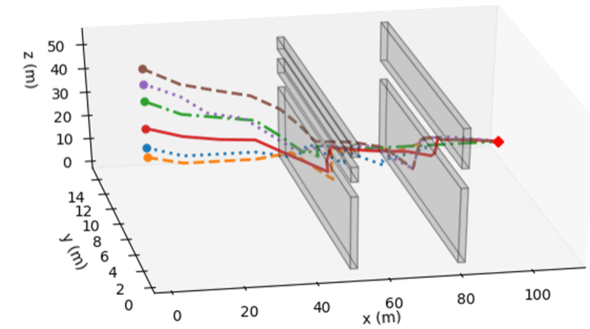


Fig. 7: 3-dimensional perspective of the simulation with 6 UAVs and horizontal and vertical discretization of 1m.

of the literature which have not been used to develop it. Furthermore, it can take advantage of obstacle-free areas to increase the speed of the UAVs.

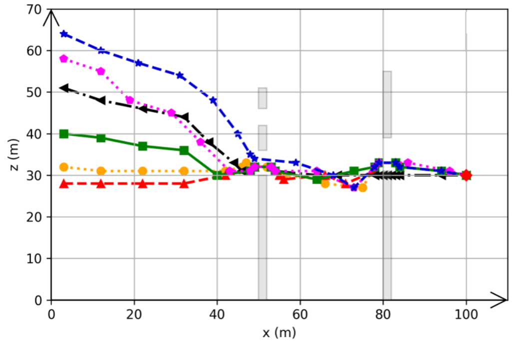
E. Swarm Simulation in Complex 3-Dimensional Environment

Finally, we tested our mobility model in 3D in a representation of a power station. The UAVs successfully perform horizontal or vertical obstacle avoidance, depending on the building height and on the location of their neighbors.

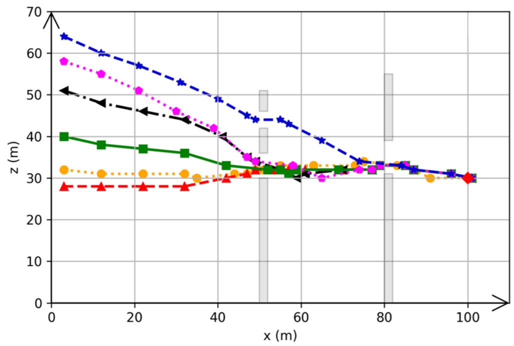
Fig. 9 shows the trajectories of a 4-UAV swarm in this complex environment. One can note that in this test, the UAVs have different goal points. Indeed, as the collaborative process is based on the obstacles locations sharing, the mobility model can be used as it is with individual departures position and/or goals.

V. Conclusion and Future Work

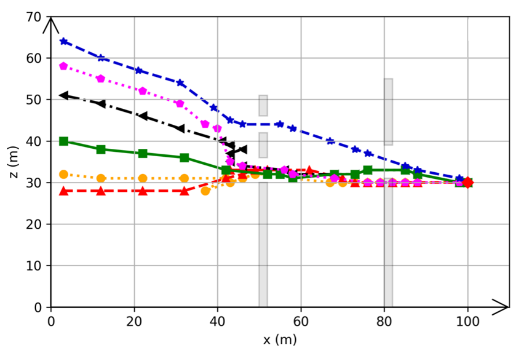
This paper presents a 3D mobility model for swarms of collaborative UAVs based on APF method. We introduced the 3DGPeach mobility model, an extension of the Peach mobility model, to which we added a global path planning method and the third dimension. The method is validated using OMNeT++ with swarms of 3 to 9 UAVs in several environments containing 3D obstacles. Simulations are used for comparisons with the mobility model of Sun *et al.* [20] and for comparison with our previous work [10] in a reconstruction of a real environment. Communication between the platforms precisely simulates a XBee module, and simulations of embedded sensors allowing obstacles detection are also performed. In future work, the simulation will be improved to be more realistic in terms of kinetics by using multi-rotors behaviour in our model. A more operational approach will also be developed by integrating surveillance sensors such as airborne visible/IR camera. The will be to determine the impact of the sensors capability in our model.



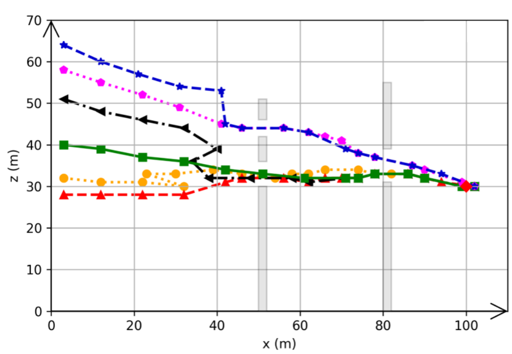
(a) Cell width = 1m.



(b) Cell width = 2m.

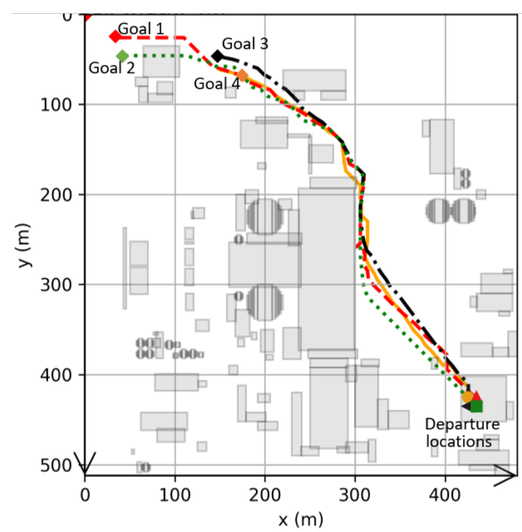


(c) Cell width = 3m.

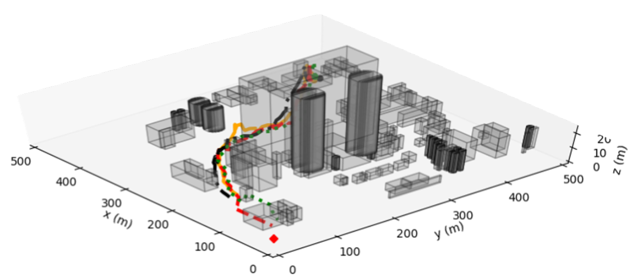


(d) Cell width = 4m.

Fig. 8: Projections in a vertical plan of the trajectories of a 6-UAV swarm, with a vertical discretization of 1m and various horizontal discretizations, towards a common target point represented by a red square.



(a) Upper view of the trajectories.



(b) 3D view of the trajectories.

Fig. 9: Trajectories of a 4-UAV swarm in 3D with an horizontal discretization of 4m and a vertical discretization of 4m between 4 and 24m of altitude.

APPENDIX

Main Communication Parameters used is OMNeT++:

- Wireless Local Area Network (WLAN)
  - type name: "IdealWirelessNic"
  - communication range: 90m
  - full duplex: false
- Routing
  - destination address: 10.0.255.255
  - forwarding: false
  - optimize routes: false
- UDP application
  - type name: own model based on "UDPBasicApp"
  - send interval: 2s
  - bitrate: 250kbits/s
- Radio
  - radio type: "APSKScalarRadio"
  - radio medium type: "APSKScalarRadioMedium"
  - carrier frequency: 2.4Ghz
  - bandwith: 2MHz
  - background noise power: -90dBm
  - transmitter power: 63mW
  - receiver sensitivity: -102dBm
  - receiver snir threshold: 4dB
  - receiver ignore interference: false
  - transmitter header bit length: 192b
- Environment
  - ground type: "FlatGround"
  - ground elevation: 0m
  - path loss type: "TwoRayGroundReflection"

REFERENCES

[1] DJI Mavic Air – Specs, Tutorials & Guides. `https://www.dji.com/mavic-air/info`.

[2] Top Collision Avoidance Drones - COPTRZ. `https://www.coptrz.com/top-collision-avoidance-drones/`.

[3] Varga András and Hornig Rudolf. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, Marseille, France, March 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[4] İlker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying Ad-Hoc Networks (FANETs): A survey. *Ad Hoc Networks*, 11(3):1254–1270, May 2013.

[5] Jovan Boskovic, Nathan Knoebel, Nima Moshtagh, Jayesh Amin, and Gregory Larson. Collaborative Mission Planning & Autonomous Control Technology (CoMPACT) System Employing Swarms of UAVs. In *AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, August 2009. American Institute of Aeronautics and Astronautics.

[6] Pascal Bouvry, Serge Chaumette, Grégoire Danoy, Gilles Guerrini, Gilles Jurquet, Achim Kuwertz, Wilmuth Müller, Martin Rosalie, and Jennifer Sander. Using Heterogeneous Multilevel Swarms of UAVs and High-Level Data Fusion to Support Situation Management in Surveillance Scenarios. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, 2016.

[7] Pascal Bouvry, Serge Chaumette, Grégoire Danoy, Gilles Guerrini, Gilles Jurquet, Achim Kuwertz, Wilmuth Müller, Martin Rosalie, Jennifer Sander, and Florian Segor. ASIMUT project: Aid to SItuation Management based on MUltimodal, MUltiUAVs, MUltilevel acquisition Techniques. In *3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet)*, DroNet '17 Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, pages 17 – 20, Niagara Falls, United States, June 2017.

[8] C.I. Connolly, J.B. Burns, and R. Weiss. Path planning using Laplace's equation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2102–2106, Cincinnati, OH, USA, 1990. IEEE Comput. Soc. Press.

[9] Fintan Corrigan. Top Collision Avoidance Drones And Obstacle Detection Explained. `https://www.dronezon.com/learn-about-drones-quadcopters/top-drones-with-obstacle-detection-collision-avoidance-sensors-explained/`, June 2018.

[10] Ema Falomir, Serge Chaumette, and Gilles Guerrini. A Mobility Model Based on Improved Artificial Potential Fields for Swarms of UAVs. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, October 2018. IEEE.

[11] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[12] X. Li and J. Chen. An Efficient Framework for Target Search with Cooperative UAVs in a FANET. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 306–313, December 2017.

[13] Xianfeng Li, Tao Zhang, and Jianfeng Li. A Particle Swarm Mobility Model for Flying Ad Hoc Networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, December 2017.

[14] Yuecheng Liu and Yongjia Zhao. A virtual-waypoint based artificial potential field method for UAV path planning. In *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese*, pages 949–953. IEEE, 2016.

[15] M. Messous, S. Senouci, and H. Sedjelmaci. Network connectivity and area coverage for UAV fleet mobility model with energy constraint. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6, April 2016.

[16] Kamesh Namuduri, Serge Chaumette, Jae H. Kim, and James P. G. Sterbenz. *UAV Networks and Communications*. Cambridge University Press, November 2017.

[17] Hui Peng, Fei Su, Yanong Bu, Guozhong Zhang, and Lincheng Shen. Cooperative area search for multiple UAVs based on RRT and decentralized receding horizon optimization. pages 298–303, 2009.

[18] Drew Prindle. Pocket-sized and practically perfect, the Mavic Air is DJI's best drone yet. `https://www.digitaltrends.com/drone-reviews/dji-mavic-air-review/`, July 2018.

[19] Martin Rosalie, Matthias R. Brust, Gregoire Danoy, Serge Chaumette, and Pascal Bouvry. Coverage Optimization with Connectivity Preservation for UAV Swarms Applying Chaotic Dynamics. pages 113–118. IEEE, July 2017.

[20] Jiayi Sun, Jun Tang, and Songyang Lao. Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm. *IEEE Access*, 5:18382–18390, 2017.

[21] G. A. Venkatesh, P. Sumanth, and K. R. Jansi. Fully Autonomous UAV. In *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 41–44, April 2017.

[22] Kyle Hollins Wray, Dirk Ruiken, Roderic A. Grupen, and Shlomo Zilberstein. Log-space Harmonic Function Path Planning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1511–1516, Daejeon, South Korea, October 2016. IEEE.

[23] E. Yanmaz. Connectivity versus area coverage in unmanned aerial vehicle networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 719–723, June 2012.

[24] Min Zhang, Yi Shen, Qiang Wang, and Yibo Wang. Dynamic artificial potential field based multi-robot formation control. In *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings*, pages 1530–1534, Austin, TX, USA, 2010. IEEE.