DOI. No. 10.1109/MAES.2020.2978959

Feature Article:

UAVs Formation Approach Using Fast Marching Square Methods

C. A. Monje, S. Garrido, L. Moreno, C. Balaguer,
Carlos III University of Madrid

INTRODUCTION

Multiagent systems improve the performance, flexibility, and robustness of the mission [1], including the common applications of exploration [2], search and rescue [3], and surveillance [4], [5], among others. The formation problem requires us to address important research topics, such as modeling and control of agents [6], collision avoidance [7], mapping and state estimation [8], and formation control and planning [9].

Regarding formation control and planning, the main problem is to provide a group of coordinated agents to perform specific tasks while keeping certain geometric configurations. The coordination of the agents is key research topic. When the operation is performed in limited spaces or for collaborative tasks, the movements of the agents have to be planned and coordinated efficiently. As well, a computationally fast solution is also required so that the travel speed can be maintained.

There exist several strategies that describe how to control the evolution of a formation. For instance, the multiagent coordination problem is studied in Ogren *et al.* [10] under the framework of Lyapunov control. Other approaches are based on potential fields, which are combined in order to get the desired behavior of the formation [11]. In other behavior-based approaches [12], each agent has basic primitive actions that generate the desired behaviors in response to sensory inputs. A common solution in the leader–followers approach, is the model predictive controller [13], which was recently introduced for holonomic robots [14].

Another interesting approach is that by Olfati-Saber on flocking for multiagent dynamic systems [15], subsequently adapted by Iovino et al. [16] for UAV swarming with obstacle avoidance capability. This method is based on collective potentials between alpha agents that are flock members, beta agents that are used to represent obstacles, and gamma agents that represent partial objectives.

The main drawbacks of the methods cited before are, among others, the mathematical complexity needed to obtain satisfactory results and the existence of local minima during the execution of the algorithms. As demonstrated in Gomez *et al.* [17], the fast marching square (FM^2) approach shows a robust performance when it comes to these two issues. This is why we have taken a step toward its application to UAVs formations.

In this article, an approach is presented for the calculation of the trajectories that the UAVs of a formation must follow when moving toward an objective, based on a leader–followers scheme. At the same time, the followers are positioned with respect to the leader according to a geometric shape that can change, within a given range, in order to face the environment's characteristics [18], [20], [21].

Different from the approach in [17] and [18], referring to indoor applications for mobile robots, the main contributions of this article are as follows 1) the FM^2 technique is extended to be applied in 3-D outdoor environments for UAVs formation applications with more restrictive kinematic constraints; 2) the FM^2 method is modified to introduce two adjustment parameters p_1 and p_2 that allow both changing the smoothness of the paths and setting the flight level in a very intuitive way and without adding computational complexity to the approach; 3) the generated paths are optimal in terms of distance cost, safety, and smoothness; 4) the approach can be equally applied when the number of followers is drastically increased (two followers have been selected in this article for the sake of simplicity), and even for swarm configurations (no leader); 5) the planning method do not rely on either probabilistic techniques or optimization methods (not proper when it comes to certification issues), which makes it more suitable for its use in real aviation applications.

The Problem Statement section presents the environment and the mission characteristics. The section on UAVS Formation Approach presents an approach to

Authors' current addresses: C. A. Monje, S. Garrido, L. Moreno, C. Balaguer, RoboticsLab, Department of Systems and Automation, Carlos III University of Madrid, Madrid 28903, Spain (e-mail: sgarrido@ing.uc3m.es).
Manuscript received July 26, 2019, revised November 20, 2019; accepted February 24, 2020, and ready for publication March 4, 2020.
Review handled by Giancarmine Fasano.
0885-8985/20/$26.00 © 2020 IEEE

Credit: Image licensed by Ingram Publishing

create a formation composed of three UAVs and also explains how this formation is able to adapt to the environment with respect to the ground. In the next section, we address the path planning with flight-level constraint is addressed. We then present and discuss results from two cases of application: formation performance with and without flight-level constraint. Finally, the main conclusions of the work are outlined and guidance for future work provided.

PROBLEM STATEMENT

PROBLEM STATEMENT

In this section, the problem statement is divided in two different issues: First of all, the environment where the path planning is carried out is described; later, the mission for the UAVs formation is described.

ENVIRONMENT

The 3-D environment where the path planning for UAVs formation is carried out is represented in Figure 1. The left part of the figure represents an open field with mountainous terrain, where the surface is rather uneven. The 3-D grid map has a dimension of 120 × 90 × 40 cells, where each cell of the map is equivalent to 15 × 15 × 15 m. It is not necessary to consider the sizes of the UAVs, since it is assumed to be smaller than the size of a cell. On the other hand, the right part of the figure represents the frontal view of the environment, where two lateral mountains are appreciated. These two mountains form a fissure, which will be crossed by the UAVs formation.

MISSION FOR THE UAVS FORMATION

The mission presented requires a UAVs formation moving throughout an open field, avoiding any obstacle in the terrain. There are several types of formations; however, this article focuses on a leader–followers formation. That is, the trajectory is calculated for a single UAV (leader), which flies from a start position to a goal position, being the head of the formation, and rest of the UAVs (followers) follow the leader respecting several geometrical relations. The formation in this article is formed by three UAVs that compose a triangular shape among them. The formation will avoid any obstacle in the environment, deforming and adapting the path to its characteristics, and also taking into account the rest of the UAVs of the formation.

The approach implemented to find the trajectory for the leader and its followers is based on our FM^2 approach. The method to achieve a restriction in flight level is implemented here, keeping the leader with a fixed flight level with respect to the ground. This entire process is explained in the following sections.

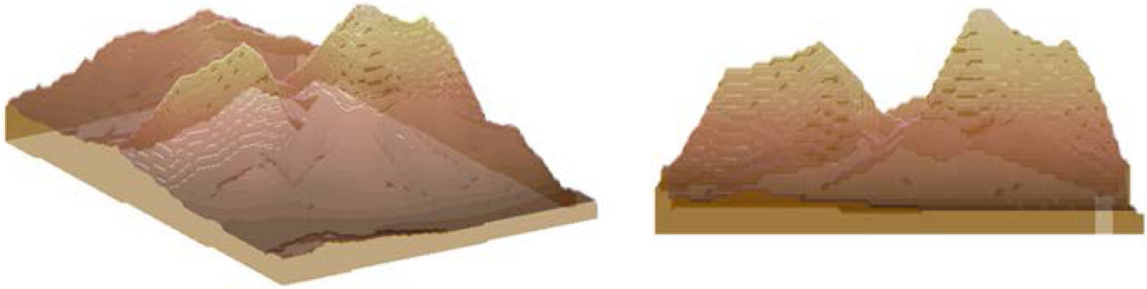


Figure 1.
The left side of the image shows the 3-D simulated representation of the open field environment. The right side shows the front view of the environment.

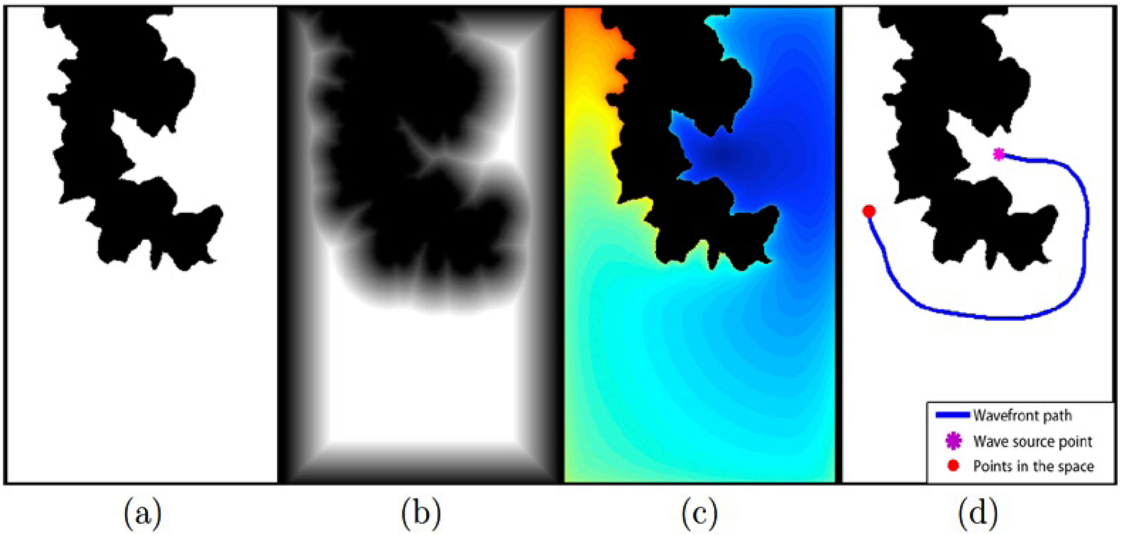


Figure 2.
The fast marching square (FM^2) method. (a) Binary map W_0. (b) Velocities map W. (c) Time of arrival map D. (d) Resulting path.

FAST MARCHING AND FAST MARCHING SQUARE METHODS

The fast marching method (FMM), introduced by Sethian [22], generates optimal trajectories in terms of distance. However, this is not the only thing to take into account when carrying out a path planning for a robot. The trajectory must be smooth without sharp curves, always respecting a certain turning radius. Also, the trajectory must have safety margins with respect to the obstacles to prevent accidents with the environment. These two deficiencies are solved by applying the FM^2 method.

The FM^2 method was introduced by Garrido *et al.* [19] and, as mentioned above, is based on applying the FMM twice. The first time that FMM is applied, a potential map W is generated, then, the FMM is applied again to generate the path between two points. The procedure to obtain a path between two points is as follows.

FAST MARCHING SQUARE METHOD

- *Environment (W_0)*: The input of the method is a 3-D grid map, which is read as a binary map [see Figure 2 (a)]. The obstacles are identified with value 0 (black) and the free space is identified with value 1 (white).

- *Velocities map (W):* Each cell of the grid map labeled as obstacle is used as source point of the FMM. In this way, a potential map is generated as shown in Figure 2 (b). This map in grayscale is rescaled to fix the maximum and minimum values as 1 and 0, respectively. The value of each cell is proportional to the distance from the obstacles; in other words, now the free space keeps a certain distance from the obstacles. This map is also called *velocities map* because the value of each cell can be interpreted as the speed of the vehicle, that is the speed is faster when the vehicle is far from the obstacles (clear areas) and the speed is slower when the vehicle is approaching obstacles (obscured areas). But in this article, the speed has not been considered when carrying out the path.

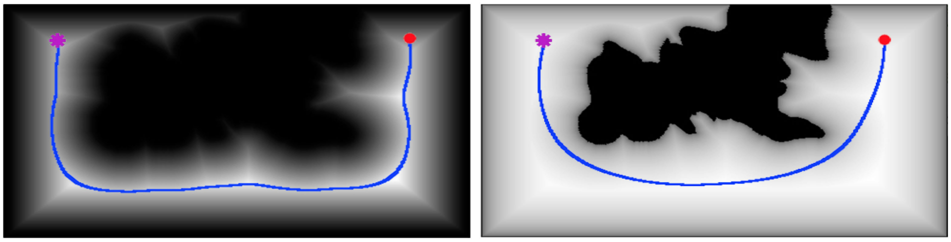
- *Time of arrival map (D):* The FMM is applied again over the map W, where the wave is expanded from the goal point until the start point. The result of this process is the *time of arrival map D* shown in Figure 2(c).

- *Resulting path:* The resulting path [see Figure 2(d)] is obtained applying the gradient descent over D from the start point to the goal point. The resulting path is the most optimal in terms of smoothness and safety.

However, many times the resulting paths are not optimal in terms of safety margins or smoothness, since the path does not befit the requirements of the mission. Thus, W can be modified according to certain specifications, such as security margins and kinematics of the vehicle.

Each cell of the map W can be raised to a value specified by the user. This value is called adjustment parameter. This procedure causes a lightening or darkening of W, as shown in Figure 3(a). If the cells are raised to a value greater than 1, the map is darkened, producing paths with sharp curves and further away from the obstacles. By contrast, if the cells are raised to a value smaller than 1, the paths are smoother and closer to the obstacles [see Figure 3(b)].

It is noteworthy that, thanks to the adjustment parameters introduced in the path planning algorithm, feasible trajectories for the UAVs can be achieved. In [23], Gonzalez *et al.* have proven to generate paths with adaptive smoothness and compatible with UAV kinematic restrictions, where the paths resulting from FM^2 are compared with those resulting from considering the Dubins model.



(a) (b)

Figure 3.
W raised to different values. (a) W raised to $\frac{3}{2}$. (b) W raised to $\frac{1}{4}$.

Besides, since the method guarantees feasible paths without the need to implicitly include the kinematic models into the algorithm, the computational cost is reduced considerably, allowing this algorithm to be executed even in real time for dynamic environments. This characteristic will be explored in future research.

UAVS FORMATION APPROACH

UAVS FORMATION APPROACH

This section presents the approach to create a formation composed of three UAVs with triangular shape, which operates in a 3-D open field environment. It also explains how this formation is able to adapt to the environment, deforming according to a fixed flight level with respect to the ground. The algorithm used in this approach is an adaptation of the FM^2 path planner method described in Gomez *et al.* The improvements included here are: 1) the adaptation of the formation to a 3-D environment, and 2) the imposition of a fixed flight level with respect to the ground for the leader. As mentioned above, the formation considered follows the leader–followers configuration. The positions of each follower are determined with geometric equations according to the leader's pose; that is, each position of the followers is found taking the position of the leader as reference. These geometric relations are shown in Figure 4, where v is the direction of the leader, u is the perpendicular to the direction, and the partial goals are the next positions for each follower. In the following section, a detailed description of our approach is presented.

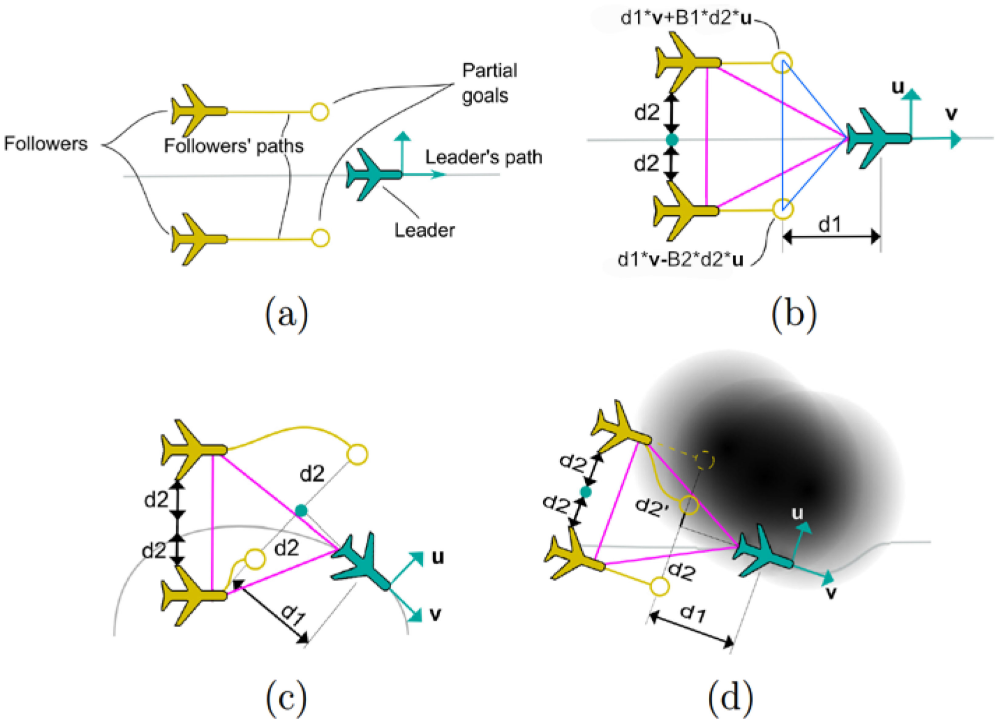


Figure 4.
Behavior of the UAV formation algorithm. (a) Main components. (b) Triangular-shaped UAV formation. (c) Partial goals according to the leader position. (d) Partial goals according to the obstacles of the environment.

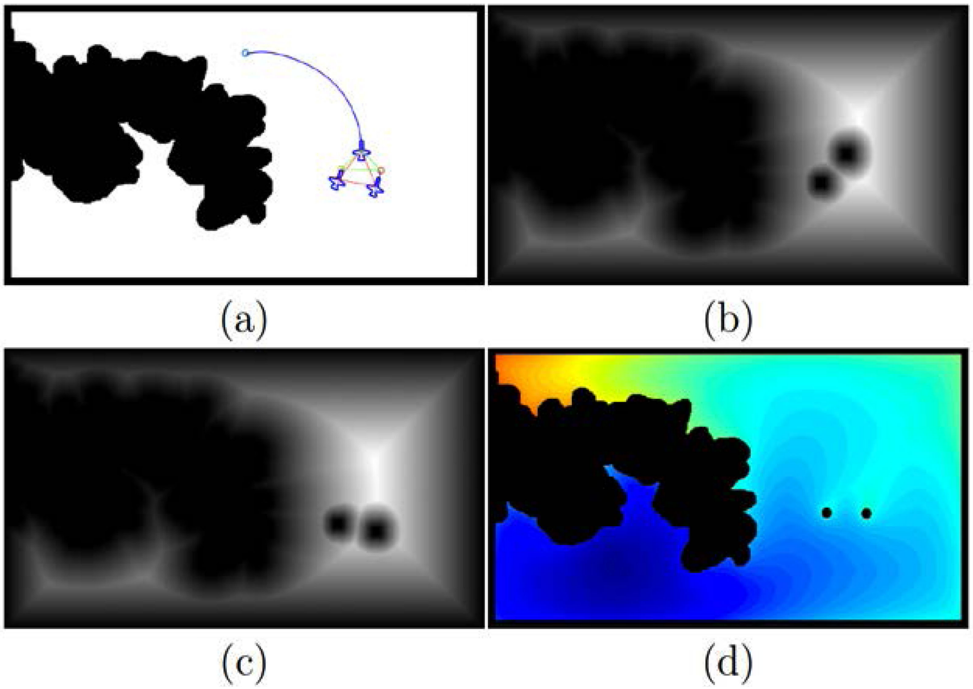


Figure 5.
UAVs formation approach. (a) UAVs formation following a path. (b) First potential map taking the leader and one follower as obstacles. (c) First potential map taking the followers as obstacles. (d) Second potential map taking the followers as obstacles.

UAVS FORMATION ALGORITHM

As described previously, the FM^2 technique uses the FMM twice to create two different potential maps. The first time, the FMM creates a potential map identified as W, and the second time it generates a wave front growing into W and giving the map T as a result. To achieve the deformation of the triangular shape formation, not only the characteristics of the environment are taking into account, but also the complete UAVs set. For this reason, each of these vehicles is treated as an obstacle in the environment. Each of the UAVs, like the rest of obstacles, must have additional repulsive forces, preventing them from colliding with each other. Therefore, the integration of the potential given by the FMM into each UAV of the formation is necessary. The steps to follow for the formation planning are detailed next:

1) The environment map is read as a binary map (W_0), where the obstacles are identified with value 0 (black) and the free space with value 1 (white).
2) The FMM is applied to W_0 generating the first potential map W.
3) The FMM is applied again into W giving rise to the second potential map T.
4) The gradient descent is applied over T according to the FM^2 method. The generated path is the route to be followed by the leader.
5) Once the path for the leader has been generated, a loop starts generating each path that the followers must follow.

The movement of each UAV$_i$ is represented by a cycle t, where the path for each of them toward their next position is calculated. This next position is the partial goal, which the followers have to reach in each cycle. The cycles t are generated by a loop, which is described as follows:

- Each UAV$_i$ of the formation is included in its binary map $W_{0,i}^t$ together with the rest of the UAVs, leaders and followers, labeled as obstacles. Figure 5 (a) represents the original binary map $W_{o,i}^t$ with the obstacles in black (value 0) and the admissible part in white (value 1).

- For each UAV$_i$, a new first potential W_i^t is generated from $W_{o,i}^t$ in each cycle t. Figure 5(b) shows the W_i^t gray map obtained from the initial binary map $W_{o,i}^t$ with two black points representing the leader and the other follower. Then, the FMM is applied using as initial points all the black ones. This step can also be done using the distance transform (in MATLAB the "bwdist" command), but the performance is not that good because of its discrete nature. Figure 5(c) represents the W_i^t gray map corresponding to the leader, obtained from the initial binary map $W_{o,i}^t$ with two black points representing the two followers and then applying the FMM using as initial points all the black ones.

- The partial goals $Pd(x_{g,j}, y_{g,j}, z_{g,j})$ for each follower are calculated from the position of the leader. These partial goals indicate the desired next position for each follower; when that position is calculated, the obstacles of the environment (including the rest of UAVs) are taken into account. For this purpose, the gray level of each partial goal's position is calculated, thus modifying the distances (edges) of the triangle shape. This method works as a repulsive force with the obstacles of the environment. Figure 4 shows how the geometry of the formation is affected and how the next partial goals are calculated, where the triangular shape distances for follower 1 and follower 2 are given by following:

$$\begin{aligned} Pd_{f1,i} &= P_{l,i} - 2 \cdot d_1 \cdot \mathbf{v} + B_{f1,i+1} \cdot d_2 \cdot \mathbf{u} \\ Pd_{f2,i} &= P_{l,i} - 2 \cdot d_1 \cdot \mathbf{v} - B_{f2,i+1} \cdot d_2 \cdot \mathbf{u}. \end{aligned} \tag{1}$$

  In these equations, $P_{l,i}$ is the position of the leader, $Pd_{f1,i}$ and $Pd_{f2,i}$ are the desired positions of the followers, d_1 and d_2 are the safety distances in the direction of the leader and in perpendicular direction between the UAVs, shown in Figure 4 and $B_{f1,i+1}$ and $B_{f2,i+1}$ are $2 - W_i^{t+1}$, where W_i^{t+1} is the gray level of the follower's current position (level of proximity to obstacles)

$$\begin{aligned} B_{f1,t+1} &= 2 - W_1^{t+1} \\ B_{f2,t+1} &= 2 - W_1^{t+1}. \end{aligned} \tag{2}$$

  The generalization for regular polygons of n sides, with the leader as fist vertex and radius r (distance from the center to the vertices) is

$$Pd_{fk,i} = P_{l,i} + r * \left[B_{fk,t+1} * \cos\left(\frac{2k\pi}{n}\right), \sin\left(\frac{2k\pi}{n}\right), 0 \right]$$

  where $P_{l,i}$ is the position of the leader and $Pd_{fk,i}$ is the position of the follower k. The term $B_{fk,t+1}$ represents the safety margin in perpendicular direction.

  As has been noted, the distances are only modified in the plane $x - y$; this means that the flight level for the followers is the same as that of the leader. For the case when a flight level is fixed, the flight level of each follower is stipulated by the terrain.

- The second potential map T_i^t is obtained applying the FMM into W_i^t. Figure 5(d) shows how the FMM is applied taking into account the followers as obstacles.
- The gradient descent is applied into T_i^t obtaining the path for each UAV_i from its current position until its partial goal.
- Each follower moves forward following the generated path until a new iteration is completed. The low computational cost of the FM^2 method allows an adequate refresh rate. All this process is summarized in Figure 6.

Regarding the execution of follower cycles, each cycle is executed with the minimum time span possible between cycles, and this time spam depends on the planning algorithm computational cost and the number of followers. Once the partial paths are calculated for each UAV in a cycle, the next cycle starts immediately. Table 1 has been computed in order to show the computation time required for each cycle depending on the number of followers. The number of followers has been increased from two to four, the computation time being around 2.5 s, and showing a linear growth with the number of UAVs.

PATH PLANNING WITH FLIGHT-LEVEL CONSTRAINT

The characteristic that makes the FM^2 method so interesting is that by modifying the gray levels of matrix W, it is possible to generate the desired trajectories.

Algorithm 1. Introduction of the flight level in the method.

Require: The velocities map W of a gridmap G of size m x n x l.
Require: Flight level L_w with respect to the ground.
Require: Adjustment parameters p_1 and p_2.
Ensure: The velocities map W with the clarified flight level cells.

```
for $k$ to $l$ do
  for $j$ to $n$ do
    for $i$ to $m$ do
      $SurfaceValue \leftarrow Surface(i, j)$
      if $k = (L_w + SurfaceValue)$ then
        $w_{i,j,k} \leftarrow (w_{i,j,k})^{p_1}$
      else
        $w_{i,j,k} \leftarrow (w_{i,j,k})^{p_2}$
      end if
    end for
  end for
end for
```

In many cases there are restrictions on an absolute flight level, or on the surface of the terrain. The restrictions refer to the trajectory with the exception of the start and end points.

As the fast marching wave front propagates more rapidly through the lighter areas, it is necessary to impose

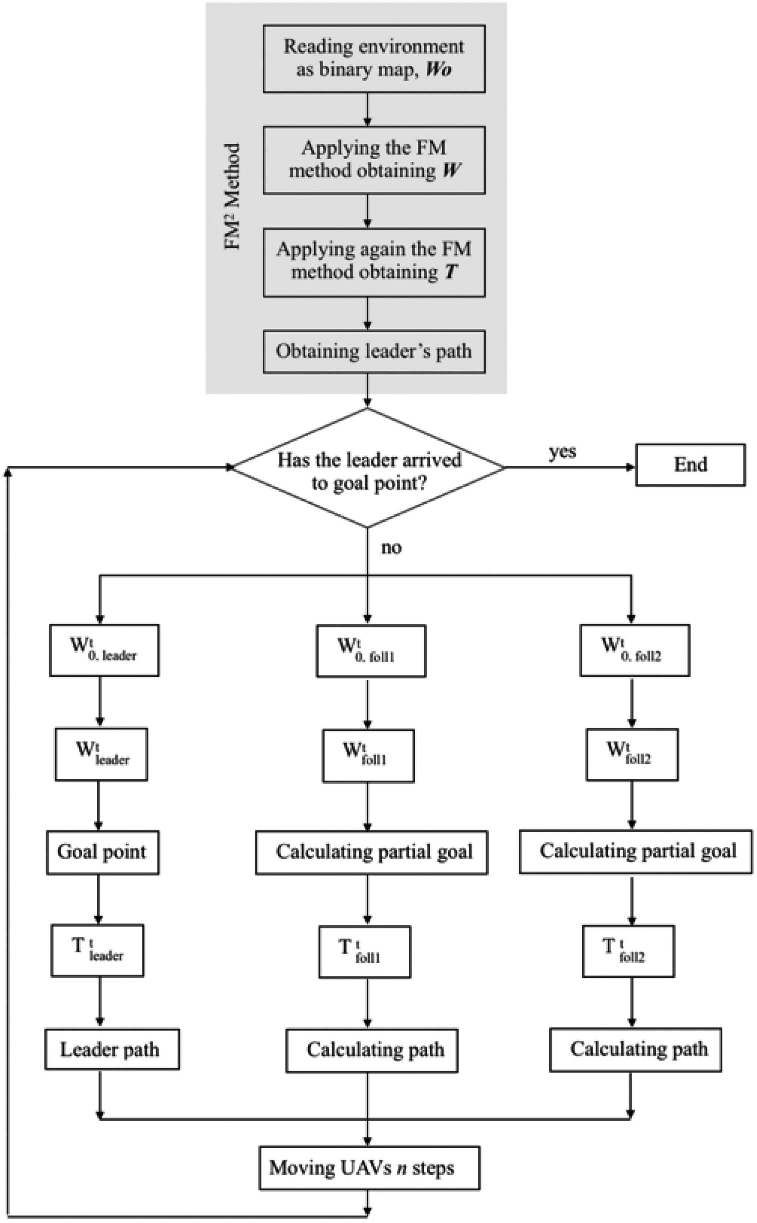


Figure 6.
UAVs formation computation flow.

that the pixels corresponding to the desired flight level are clearer in the W matrix.

It should be noted that the method works with a 3-D grid. Therefore, it is enough to clarify the desired W layers and darken the unwanted ones, as shown in Figure 7.

The method to apply these ideas into FM^2 and to set a level of flight relative to the terrain is shown in Algorithm 1. As you can see, two adjustment parameters p_1 and p_2 are used to clarify or obscure the cells of matrix W in order to obtain the desired trajectory.

The process to modify the gray level of the cells of matrix W is the following:

- First, the elevation of the terrain is calculated (see Algorithm 1, line 4), based on the first layer of the matrix that is considered to be sea level.
- Second, the desired flight level is added to the value of the elevation of the terrain (Algorithm 1, line 5). To clarify these cells, their value is raised to p_1 (Algorithm 1, line 6). This forces the trajectories to go through these layers.
- To obscure the rest of the layers, the value of the corresponding cells is raised to p_2 (Algorithm 1, line 8). This makes it more difficult for the trajectories to pass through them.

Table 1.

Computation Time in Seconds Required for Each Cycle Depending on the Number of Followers

Grid map dimension (cells)	120×90×40
Leader+2 followers	2,10
Leader+3 followers	2,35
Leader+4 followers	2,58

RESULTS AND DISCUSSION

The path for the leader is planned from a start point to a goal point with the approach based on the FM^2 method, taking into account the simulated environment and maintaining a fixed flight level with respect to the ground. The path for the followers is estimated by geometric equations, where the goal point of each follower is placed according to the leader's pose. Any obstacle in the terrain is avoided by all the UAVs of the formation.

The simulation results carried out show how the formation of three UAVs is maintained during the whole planning and is only deformed when the UAVs avoid the obstacles of the environment. Here, two cases are presented: The first case presents a planning without altitude constraint for the UAVs formation; the second case presents a planning with altitude constraint for the same formation. In this latter case, the fixed flight level is maintained by the leader of the formation, using parameters p_1 and p_2 to modify the map W accordingly. The followers have a fixed flight level imposed by the user, which, in this case, is the same as the flight-level restriction for the leader. In this way, it is appreciated how the formation changes according to the different flight levels of its agents.

The start and goal points are the same for both cases, being p_s (40, 30, 25) and p_g (40, 112, 30), respectively. Next, the two simulation cases are discussed in detail.

CASE 1: FORMATION WITHOUT FLIGHT-LEVEL CONSTRAINT

The aim of this experiment is to plan the optimal trajectory for a UAVs formation without imposing a fixed flight level with respect to the ground. We will test how the formation is deformed when the UAVs find obstacles in their paths and how they avoid other agents of the formation. Figure 8 shows the results after running the algorithm for this particular case, presenting the resulting sequence of movements. The distance between UAVs is eight cells and each frame of the sequence is chosen in time intervals of 20 s, approximately. Figure 8(a) shows the movements of each element of the formation. The green triangle represents the partial objective for each UAV, that is to say, the desired position for each UAV, while the red triangle represents the real position for each UAV. Figure 8(b) shows the same movements as in the previous sequence, but referring to the map W. The altitude at which each UAV is flying is 30 cells, approximately. As can be seen, each UAV is represented as an obstacle in the environment, which avoids the agents from colliding with each other. Furthermore, Figure 9 shows the simulated path with respect to the ideal path and the terrain. It can be seen how the UAVs do not maintain a fixed flight level with respect to the ground. The computational time of the total planning, including the loop for the planning of the followers, is about 130.6 s.

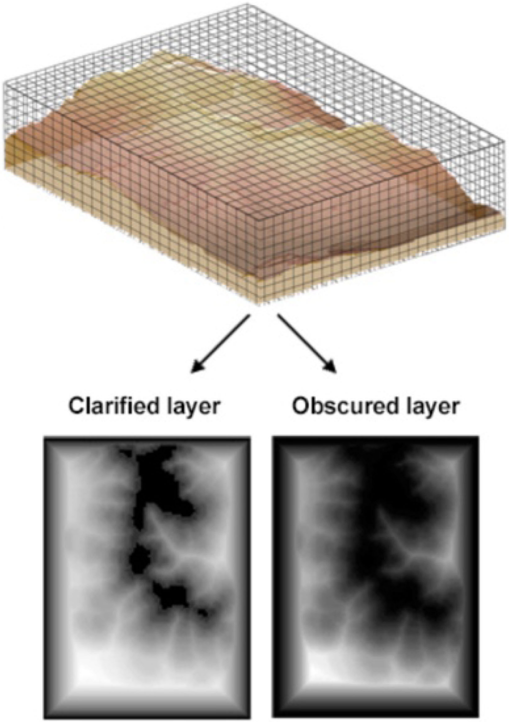


Figure 7.
The top image shows the 3-D grid map of the environment. The lower image shows clarified and obscured layers of the map W.

CASE 2: FORMATION WITH FLIGHT-LEVEL CONSTRAINT

This second case analyzes the optimal trajectory for a UAVs formation with a fixed flight level with respect to the ground. Here, the computed trajectory for the leader is the result from the FM^2 algorithm. However, the flight level of the followers has been fixed by the user at the same value specified for the leader. In this way, it is appreciated how the formation adapts to the environment taking into account the different flight levels of each of its elements. To maintain the corresponding flight level for the leader, it is necessary to fix the values of p_1 and p_2 as explained in the section "Path Planning With Flight-level Constraint." In this case, the values of p_1 and p_2 has been

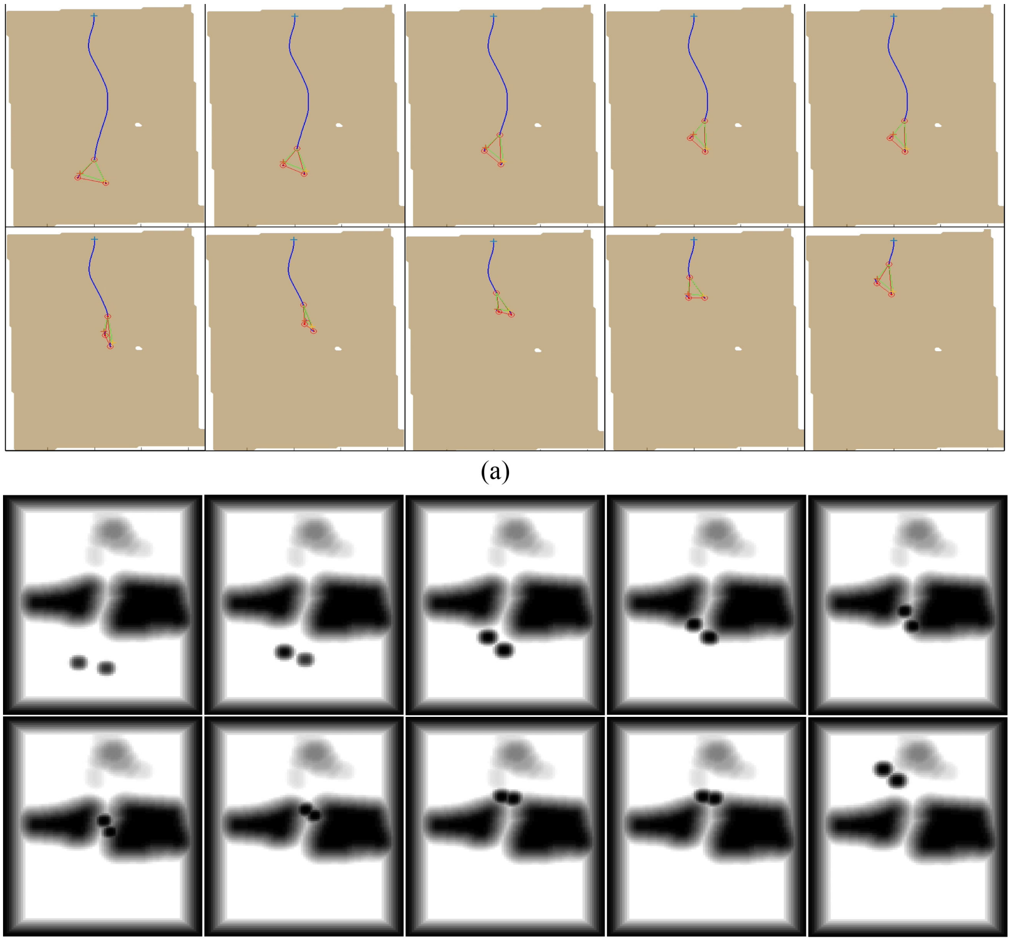


Figure 8.
Sequence without flight-level restriction. (a) Normal map. (b) Map W from leader perspective.

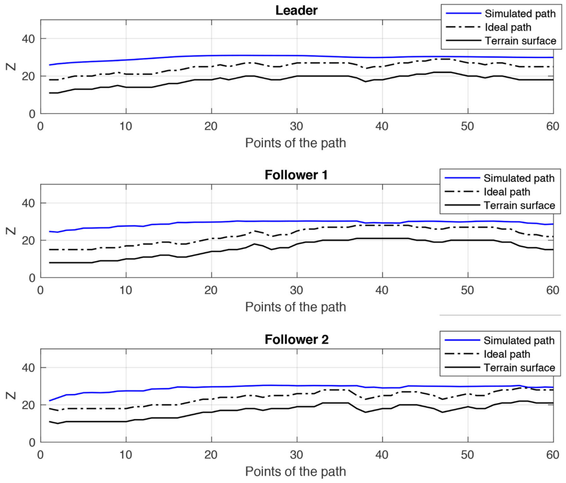
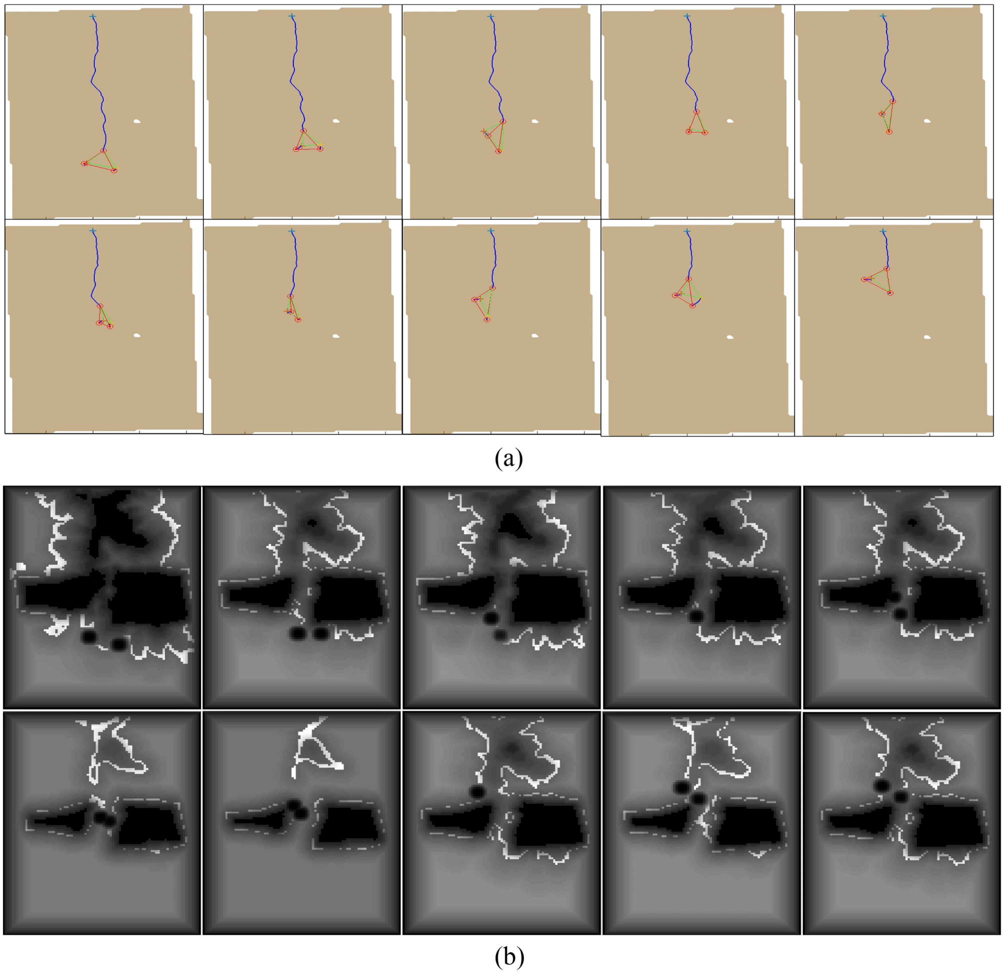


Figure 9.
Comparison of the resulting path without flight restriction against the ideal path and the terrain profile: (a) leader; (b) follower 1; (c) follower 2.

fixed to 1 and 0.5, respectively, and the altitude with respect to the ground is seven cells. As has been discussed previously, the values chosen for p_1 and p_2 allow to maintain the desired altitude and give a sufficient smoothness to the path. Figure 10(a) shows the sequence of movements of each UAV as component of the formation. Each frame of this sequence is taken in time intervals of 50 s.

This time increment is due to a higher computation cost imposed by the introduction of the flight level into the algorithm. Figure 10(b) shows the same sequence, but referring to the map W from the perspective of the follower 1. As can be seen, in some movements a single UAV appears as obstacle.

The paths of each agent of the formation can be seen in Figure 11. Unlike the previous case, the leader flight respects the flight level with respect to the ground at its particular positions, and the followers flight level changes in all points, depending on the value fixed by the user. In this case, the computational time of the total planning is about 135.6 s.



(a)

(b)

Figure 10.
Sequence with flight level 7 restriction. (a) Normal map. (b) Map W from follower 1 perspective.

A previous work by Gomez *et al.* [24] presents a deeper discussion on how the computation time changes with the number of cells of the environment. The article shows that the FMM and its variants are very competitive to this respect.

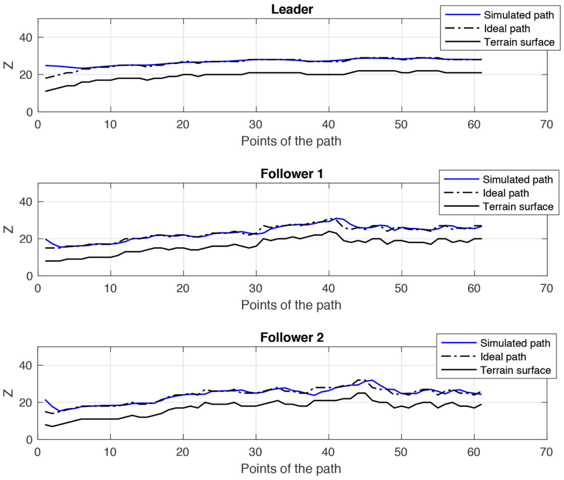


Figure 11.
Comparison of the resulting path with flight restriction against the ideal path and the terrain profile: (a) leader; (b) follower 1; (c) follower 2.

CONCLUSIONS AND FUTURE WORKS

This article has introduced the path planning problem for 3-D UAV formations based on our FM^2 algorithm. The approach has been based on a leader–followers scheme and the flight-level constraint has been considered. The simulation results have shown that the formation is able to adapt its shape so that the obstacles are avoided at the same time that it fulfils the flight-level restriction.

Our approach has been proven with successful results. The approach works in different irregular surfaces and for different vision fields of the UAV, obtaining always the feasible path with minimum cost. This results demonstrate that our approach generates paths to save energy or fuel, and in addition, generates trajectories compatibles with the kinematics of the UAV.

In further research, the explicit relationship between the adjustment parameters p_1 and p_2 and the kinematic and dynamic constraints of the UAV will be studied.

ACKNOWLEDGMENTS

This research was supported by RoboCity2030-DIH-CM Madrid Robotics Digital Innovation Hub (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos, Fase IV; S2018/NMT-4331), funded by Programas de Actividades I +D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

REFERENCES

[1] M. Martin, P. Klupar, S. Kilberg, and J. Winter, "Techsat 21 and revolutionizing space missions using micro-satellites," in *Proc. IAIAA/USU Conf. Small Satellites*, 2001, pp. 1–10.

[2] A. Dewan, A. Mahendran, N. Soni, and K. Krishna, "Heterogeneous UGV-MAV exploration using integer programming," in *Proc. AIAA/USU Conf. Small Satellites Intell. Robots Syst.*, 2013, pp. 5742–5749.

[3] S. Hauert, J. Zufferey, and D. Floreano, "Reverse-engineering of artificially evolved controllers for swarms of robots," in *Proc. IEEE Congr. Evol. Comput.*, 2009, pp. 55–61.

[4] J. Acevedo, B. Arrue, I. Maza, and A. Ollero, "Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions," *J. Intell. Robot. Syst.*, vol. 70, no. 1–4, pp. 329–345, 2013.

[5] M. Likhachev *et al.*, "Planning for opportunistic surveillance with multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5750–5757.

[6] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," PhD dissertation, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2007.

[7] S. Krabar, "Reactive obstacle avoidance for rotorcraft UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 4967–4974.

[8] S. Shen, N. Michael, and V. Kumar, "3D estimation and control for autonomous flight with constrained computation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 4967–4974.

[9] T. Hino, "Simple formation control scheme tolerant to communication failures for small unmanned air vehicles," in *Proc. Int. Congr. Aeronaut. Sci.*, 2010, pp. 1–9.

[10] P. Ogren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multiagent coordination," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 847–851, Oct. 2002.

[11] M. Zhang, Y. Shen, Q. Wang, and Y. Wang, "Dynamic artificial potential field based multi-robot formation control," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, 2010, pp. 1530–1534.

[12] Z. Cao, L. Xie, B. Zhang, S. Wang, and M. Tan, "Formation constrained multi-robot system in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 735–740.

[13] A. Ahmad, T. Nascimento, A. Conceicao, A. Molina, and P. Lima, "Perception-driven multi-robot formation control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1851–1856.

[14] K. Kanjanawanishkul and A. Zell, "A model-predictive approach to formation control of omnidirectional mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2771–2776.

[15] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[16] S. Iovino, A. R. Vetrella, G. Fasano, D. Accardo, and A. Savvaris, "Implementation of a distributed flocking algorithm with obstacle avoidance capability for UAV swarming," in *AIAA Information Systems-AIAA Infotech@Aerospace*, Jan. 2017. [Online]. Available: https://doi.org/10.2514/6.2017-0878

[17] J. Gomez, A. Lumbier, S. Garrido, and L. Moreno, "Planning robot formations with fast marching square including uncertainty conditions," *J. Robot. Auton. Syst.*, vol. 61, no. 2, pp. 137–152, 2013.

[18] D. Alvarez, J. Gomez, S. Garrido, and L. Moreno, "3D robot formations planning with Fast Marching Square," *J. Intell. Robot. Syst.*, vol. 80, no. 3, pp. 507–523, 2014.

[19] S. Garrido, L. Moreno, M. Abderrahim, and D. Blanco, "FM^2: A realtime sensor-based feedback controller for mobile robots," *Int. J. Robot. Autom.*, vol. 24, no. 1, pp. 3169–3192, 2009.

[20] S. Garrido, L. Moreno, and P. Lima, "Robot formation motion planning using fast marching," *J. Robot. Auton. Syst.*, vol. 59, no. 9, pp. 675–683, 2011.

[21] W. Yu, G. Chen, and M. Cao, "Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities," *Syst. Control Lett.*, vol. 59, no. 9, pp. 543–552, 2010.

[22] J. A. Sethian, "Theory, algorithms and applications of level set methods for propagating interfaces," *Acta Numerica*, vol. 5, pp. 309–395, 1996.

[23] V. González, C. Monje, L. Moreno, and C. Balaguer, "Fast marching square method for UAVs mission planning with consideration of Dubins model constraints," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 164–169, 2016.

[24] J. V. Gomez, D. Alvarez, S. Garrido, and L. Moreno, "Fast methods for eikonal equations: An experimental survey," *IEEE Access*, vol. 7, pp. 39005–39029, 2019.