uc3m | Universidad Carlos III de Madrid

DOCTORAL THESIS

Unmanned Aerial Vehicle Mission Planning Based on Fast Marching Square Planner and Differential Evolution

Author:

Verónica González Pérez

Directors:

Concepción Alicia Monje Micharet

Carlos Balaguer Bernaldo de Quirós

Tutor:

Concepción Alicia Monje Micharet

DOCTORAL PROGRAM IN ELECTRICAL ENGINEERING, ELECTRONICS AND AUTOMATION

Leganés, September 2018

Ph.D. Thesis

Unmanned Aerial Vehicle (UAV) Mission Planning Based on Fast Marching Square (FM^2) Planner and Differential Evolution (DE)

Candidate:
Verónica González Pérez

Advisers:
Prof. Concepción Alicia Monje Micharet
(Universidad Carlos III de Madrid)
Prof. Carlos Balaguer Bernaldo de Quirós
(Universidad Carlos III de Madrid)

Review Committee

Signature

Chair: _____

Member: _____

Secretary: _____

Degree: Doctorado en Ingeniería Eléctrica, Electrónica y Automática
Grade: _____

Leganés, September 2018

Acknowledgements

Doing a Ph.D thesis is not easy. It is a long way to go which requires a great dedication and enormous effort. Therefore, I want to thank those people without whom it would not have been possible to complete this adventure.

Starting with Concha Monje, my advisor, who has been by my side all the time. Without her advice, tenacity and support, it would not have been possible to finish this thesis. To Carlos Balaguer, my co-advisor, for trusting me and fighting for me to be accepted at Robotics Lab. And, of course, to Luis Moreno, who has been a great force and guide for me throughout the thesis.

I cannot forget my comrades in the battle: Juanmi, Pablo, Raúl and Félix. Invaluable companions and excellent people with whom I have shared great moments.

Tamara and Irene, my buddies and confidants, thanks for always being there and supporting me in the most difficult moments.

And finally, I want to dedicate this thesis to my family. I would not have come this far without them and their unconditional support.

To all, thank you for accompanying me in this important stage of my life.

Verónica González

Abstract

Nowadays, mission planning for Unmanned Aerial Vehicles (UAVs) is a very attractive research field. UAVs have been a research focus for many purposes. In military and civil fields, the UAVs are very used for different missions. Many of these studies require a path planning to perform autonomous flights. Several problems related to the physical limitations of the UAV arise when the planning is carried out, as well as the maintenance of a fixed flight level with respect to the ground to capture videos or overlying images.

This work presents an approach to plan missions for UAVs keeping a fixed flight level constraint. An approach is proposed to solve these problems and to generate effective paths in terms of smoothness and safety distance in two different types of environments: 1) 3D urban environments and 2) open field with non-uniform terrain environments.

Many proposed activities to be carried out by UAVs in whatever the environment require a control over the altitude for different purposes: energy saving and minimization of costs are some of these objectives. In general terms, the planning is required to avoid all obstacles encountered in the environment and to maintain a fixed flight level during the path execution. For this reason, a mission planning requires robust planning methods.

The method used in this work as planner is the Fast Marching Square (FM^2) method, which generates a path free of obstacles. As a novelty, the method proposed includes two adjustment parameters. Depending on the values of these parameters, the restriction of flight level can be modified, as well as the smoothness and safety margins from the obstacles of the generated paths. The Dubins airplane model is used to check if the path resulting from the FM^2 is feasible according to the constraints of the UAV: its turning rate, climb rate and cruise speed.

Besides, this research also presents a novel approach for missions of Coverage Path Planning (CPP) carried out by UAVs in 3D environments. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization

process based on the Differential Evolution (DE) algorithm in combination with the FM^2 planner.

Finally, the UAVs formation problem is introduced and addressed in a first stage using the planner proposed in this thesis.

A wide variety of simulated experiments have been carried out to illustrate the efficiency and robustness of the approaches presented, obtaining successful results in different urban and open field 3D environments.

Resumen

Hoy en día la planificación de misiones para vehículos aéreos no tripulados (UAV) es un campo de investigación muy atractivo. Los UAV son foco de investigación en numerosas aplicaciones, tanto en el campo civil como militar. Muchas de estas aplicaciones requieren de un sistema de planificación de ruta que permita realizar vuelos autónomos y afrontar problemas relacionados con las limitaciones físicas del UAV y con requerimientos como el nivel de vuelo sobre el suelo para, entre otras funciones, poder capturar videos o imágenes.

Este trabajo presenta una propuesta de planificador para vehículos aéreos no tripulados que permite resolver los problemas citados previamente, incluyendo en la planificación las consideraciones cinemáticas del UAV y las restricciones de nivel de vuelo, generando rutas suaves, realizables y suficientemente seguras para dos tipos diferentes de entornos 3D: 1) entornos urbanos y 2) campos abiertos con terrenos no uniformes.

El método utilizado en esta tesis como base para la planificación es el método Fast Marching Square (FM^2), que genera un camino libre de obstáculos. Como novedad, el método propuesto incluye dos parámetros de ajuste. Dependiendo de los valores de estos parámetros, se puede modificar la restricción de nivel de vuelo, así como la suavidad y los márgenes de seguridad respecto a los obstáculos de las rutas generadas. El modelo cinemático de Dubins se utiliza para verificar si la ruta resultante de nuestro planificador es realizable de acuerdo con las restricciones del UAV: su velocidad de giro, velocidad de ascenso y velocidad de crucero.

Además, esta tesis también presenta una propuesta novedosa para la planificación de misiones de Coverage Path Planning (CPP) en entornos 3D. Estas misiones se centran en la planificación de rutas para cubrir un área determinada de un entorno con el fin de llevar a cabo tareas de rastreo, búsqueda o rescate. La metodología seguida utiliza un proceso de optimización basado en el algoritmo Differential Evolution (DE) en combinación con nuestro planificador FM^2.

Como parte final de la tesis, el problema de formación de UAVs se introduce y aborda en una primera etapa utilizando el planificador FM^2 propuesto.

Este trabajo investigador recoge una gran variedad de experimentos simulados para ilustrar la eficiencia y robustez de las propuestas presentadas, obteniendo resultados exitosos en diferentes entornos urbanos y de campo abierto.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

This chapter deals with the initial introduction, motivation and presentation of this thesis, which addresses a path planner approach for UAVs mission planning in different 3D environments.

The method used in this work as a planner is the Fast Marching Square (FM^2) method, which generates a path free of obstacles. As a novelty, the method proposed includes two adjustment parameters. Depending on the values of these parameters, the restriction of flight level can be modified, as well as the smoothness and safety margins from the obstacles of the generated paths. The Dubins airplane model is used to check if the path resulting from the FM^2 is feasible according to the constraints of the UAV: its turning rate, climb rate and cruise speed.

Besides, this research also presents a novel approach for missions of Coverage Path Planning (CPP) carried out by UAVs in 3D environments. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the Differential Evolution (DE) algorithm in combination with the FM^2 planner.

Finally, the UAVs formation problem is introduced and addressed in a first stage using the planner proposed in this thesis.

A wide variety of simulated experiments have been carried out to illustrate the efficiency and robustness of the approaches presented here, obtaining successful results in different urban and open field 3D environments.

1.1 Motivation and Objetives

Path planning with Unmanned Aerial Vehicles (UAVs) has taken a great importance in recent years. There is a wide variety of applications where the UAVs have become an indispensable tool. Many researchers have focused on different subjects related to human security and environmental protection. Due to this fact, the Federation Aviation Administrator (FAA) has facilitated the introduction of Unmanned Aerial Systems (UAS) in the medium to carry out these different operations.

All types of missions to be performed by the UAV require a proper path planning to reach an adequate autonomous flight (Barrientos et al. (2009)). The planning needs to be adapted to any environment respecting its geometry. The minimisation of the collision risk and safety distance with the obstacles are two key elements to consider.

Different techniques are utilized to plan trajectories for UAVs, for example, Rapidly exploring Random Tree (RRT) (Lin and Saripalli (2014)), A* Search algorithms (Bo-Bo and Xiaoguang (2010)) and Probabilistic Roadmap Method (PRM) (Yan et al. (2014)). Besides, in previous works (Arismendi et al. (2015), Garrido et al. (2009)) the Fast Marching Square (FM^2) algorithm has been used to plan optimal trajectories in bidimensional scenarios for mobile robots. FM^2 has been also used as a planner to create formations of vehicles in 2D (Gómez et al. (2013)) and 3D (Alvarez et al. (2014)) environments. All these works have obtained successful results.

However, when the problem includes kinematic constraints, a simple path planning may not be sufficient to describe a solution path. Any method used for the path planning has to find an adequate solution respecting the kinematic model of the vehicle (Arismendi et al. (2015)), in this case a UAV.

This context motivates the work in this thesis, whose main objective is to develop a path planner for UAVs mission planning in 3D environments, considering the restrictions given by the environment itself, the kinematic constraints of the UAV and the safety of the operation.

The novelty of this approach lies in the fact that it is the first time that the FM^2 algorithm is implemented for path planning with UAVs in a 3D environment, where the kinematic constraints are more demanding than in mobile robots. For the sake of feasibility, we will check that the generated paths fulfill kinematic restrictions such as turning rate, climb rate and cruise speed using the Dubins model as a reference (Hanson et al. (2011), Beard and McLain (2013), Chitsaz and LaValle (2007)), since this model is the most widespread.

The Dubins model is a technique used in mobile robots and UAVs by which the minimum distance path between two configurations (initial and final position) is obtained, considering certain kinematic constraints. Thanks to this, a bounded control is reached on the vehicle model. The Dubins model has been implemented in several works for moving a vehicle in 2D environments (Hanson et al. (2011),

Bhatia et al. (2008), McGee et al. (2005)). In other works (Beard and McLain (2013), Chitsaz and LaValle (2007), Lin and Saripalli (2014), Bidabad and Sedaqat (2010)) the constraint for the altitude has been implemented for UAV applications, obtaining a more complex planning in 3D environments.

Regarding fields of application, in this thesis two different types of scenarios will be considered: on the one hand, urban environments with buildings of different geometries; on the other hand, open fields with mountainous terrain, whose obstacles are the mounds in the terrain.

According to these scenarios, the main constraints to be achieved by the planner will be:

- Imposition of a flight level;
- It has to accomplish with the kinematic limitations of the UAV;
- It has to guarantee a safe operation;
- Other cost functions could be considered, such as energy consumption.

Besides, even though the environments considered in this thesis are static, the planner proposed could also be used for dynamic planning, where the conditions of the environment are changing. This aspect will be discussed throughout the document.

As a novelty, the method proposed includes two adjustment parameters. Depending on the values of these parameters, the restriction of flight level can be modified, as well as the smoothness and safety margins from the obstacles of the generated paths.

As another contribution, this research also presents a novel approach for missions of Coverage Path Planning (CPP) carried out by UAVs in 3D environments. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the Differential Evolution (DE) algorithm in combination with the FM^2 planner.

Finally, the UAVs formation problem is introduced and addressed in a first stage using the planner proposed in this thesis.

In general terms, the advantages of our approach lie in the following aspects:

- Easy concept. The method is based on the natural movement of a wave, so conceptually, it is very easy to understand. Thanks to this, the imposition of the flight level constraint is simple to implement.
- Smoothed and safety trajectories. The method provides very smooth paths, since it is based on the propagation of a wave. Thus, the paths do not need to be refined in order to respect the kinematics of the UAV.

- Fast response. Though the expansion time of the wave depends on the complexity of the environment, the method has a fast and efficient computational speed.

All these aspects will be deeply discussed throughout the different chapters of this document.

1.2 Document Organization

The document is organized as follows:

Chapter 2 introduces the FM and FM^2 methods and their mathematical formulations, together with a discussion on the advantages of using FM^2 versus FM. Part of the content of this chapter has been previously published as a contribution of this thesis in González et al. (2016).

Chapter 3 discusses the planner approach proposed in this thesis based on FM^2. An adjustment parameter is introduced in order to modify the smoothness and safety distance of the resulting paths. The Dubins airplane model is used to check if the path resulting from the method is feasible according to the constraints of the UAV: its turning rate, climb rate and cruise speed. Part of the content of this chapter has been previously published as a contribution of this thesis in González et al. (2016).

Chapter 4 introduces the flight level constraint in the planning problem and discusses how the planner proposed fulfills this restriction with the introduction of two parameters in the algorithm. Depending on the values of these parameters, the restriction of flight level can be modified, as well as the smoothness and safety margins from the obstacles of the generated paths. Part of the content of this chapter has been previously published as a contribution of this thesis in González et al. (2017a) and González et al. (2017b).

Chapter 5 presents the CPP approach, where the missions focus on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the DE algorithm in combination with the FM^2 planner.

Chapter 6 introduces the UAVs formation problem and addresses it in a first stage using the planner proposed in this thesis.

Chapter 7 contains the concluding remarks, key contributions and recommendations for future improvements of this work.

1.3 List of Publications

The list of papers published on the novel contributions of this thesis is presented next.

Journal Papers

1. (González et al., 2017a) **González, V.**, Monje, C., Moreno, L., and Balaguer, C. (2017). UAVs Mission Planning with Flight Level Constraint using Fast Marching Square Method. *Robotics and Autonomous Systems*, 94:162–171.

2. (González et al., 2017b) **González, V.**, Monje, C., Moreno, L., and Balaguer, C. (2017). UAVs Mission Planning with Imposition of Flight Level through Fast Marching Square. *Cybernetics and Systems*, 48(2):1–12.

Conference/Symposiums Papers

1. (González et al., 2016) **González, V.**, Monje, C., Moreno, L., and Balaguer, C. Fast Marching Square Method for UAVs Mission Planning with Consideration of Dubins Model Constraints. In *20th IFAC Symposium on Automatic Control in Aerospace (ACA 2016)*, 49(17):164–169, 2016.

2. **González, V.**, Monje, C., Moreno, L., and Balaguer, C. Planificación de Trayectorias para UAVs con Fast Marching Square Adaptadas a Requerimientos de Vuelo. In *XXXVII Jornadas de Automática*, pp. 856–862, 2016, Madrid, Spain.

3. **González, V.**, Monje, C., and Balaguer, C. Planificación de Trayectorias con Vehículos Aéreos No Tripulados en un Entorno Aeroportuario. In *XXXVI Jornadas de Automática*, pp. 459–465, 2015, Bilbao, Spain.

4. **González, V.**, Monje, C., and Balaguer, C. Planificación de Misiones de Vehículos Aéreos No Tripulados con Fast Marching en un Entorno 3D. In *XXXV Jornadas de Automática*, pp. 19–25, 2014, Valencia, Spain.

As detailed in the previous section, our papers González et al. (2016) and González et al. (2017a) represent the main contributions of this thesis, and therefore the thesis document partially coincides with the content of these two papers.

Chapter 2

Fast Marching and Fast Marching Square Methods

Nowadays, any autonomous vehicle application requires an optimal path planning. The generated trajectories must be safe and smooth in order to be followed faithfully by the vehicle. The method chosen here as planner is the FM^2 method, which is based on applying the FM method twice. The FM method explains how a wave is expanded in a media, as if it was a thick liquid. A summary of these two methods is presented next.

2.1 Fast Marching Method

The FM method is a numerical algorithm that solves the arrival time of an expanding wave in every point of the space. FM was introduced by Sethian in 1996 (Sethian (1996)) and is a particular case of level set methods (Osher and Sethian (1988)) where the wave is always expanded forward, that is to say, with non-negative velocity.

The fundamental base of the FM method is the same as the Fermat principle in optics, which established in short, that the light travelling between two points always chooses the optimal path in terms of time.

The wave propagation is described by the Eikonal equation, where $\boldsymbol{\rho}$ is a point of the space, $T(\boldsymbol{\rho})$ is the arrival time of the wave front and $F(\boldsymbol{\rho})$ is the propagation speed of the wave:

$$1 = F(\rho)|\nabla T(\rho)|. \tag{2.1}$$

The point where the wave is born is ρ_0, its time being $T_0 = 0$. When the wave starts expanding, the FM method computes in each iteration the time $T_{i,j}$ for each point of the space $x_{i,j}$, according to the discretization of the Eikonal equation:

$$max\left(\frac{T - T_1}{\triangle x}, 0\right)^2 + max\left(\frac{T - T_2}{\triangle y}, 0\right)^2 = \frac{1}{F_{i,j}^2}, \tag{2.2}$$

where $F_{i,j}$ is the expansion speed of the wave for each point of the grid, $\triangle x$ and $\triangle y$ are the grid spacing for the directions of x and y, respectively, and

$$\begin{aligned} T &= T_{i,j}, \\ T_1 &= min(T_{i-1,j}, T_{i+1,j}), \\ T_2 &= min(T_{i,j-1}, T_{i,j+1}). \end{aligned} \tag{2.3}$$

This process is meticulously explained in Gómez (2015). The result is the distance map shown in Fig. 2.1(a), also denoted as time of arrival map. In order to obtain the trajectory shown in Fig. 2.1(b), the gradient descent is applied in any point of the time of arrival map up to the source point of the wave.

The FM procedure is detailed in Algorithm 1 (Arismendi et al. (2015)). The solution for the Eikonal equation can be solved iteratively over a grid map, whose cells are labeled with different types of labels: Unknown, cells whose value of T is still not known, since the wave front has not reached them yet; Narrow band, candidate cells to be part of the wave front in the next iteration (the value of these cells can be changed in future iterations); Frozen, cells whose T value is fixed since the wave has already passed.

The algorithm consists of three stages: initialization, loop and finalization. In the initialization stage, the T value of the cell where the wave starts is set to 0 and

Algorithm 1 FM algorithm.

Input: A gridmap G of size m x n, source point x_0.
Output: The gridmap G with the T value set for all cells.

```
1:  Initialization;
2:  for all g_ij ∈ x_0 do
3:    g_ij.T ← 0;
4:    g_ij.state ← FROZEN;
5:    for all g_kl ∈ g_ij.neighbours do
6:      if g_kl = FROZEN then
7:        skip;
8:      else
9:        g_kl.T ← solveEikonal(g_kl);
10:       if g_kl.state = NARROW BAND then
11:         narrow_band.update_position(g_kl);
12:       end if
13:       if g_kl.state = UNKNOWN then
14:         g_kl.state ← NARROW BAND
15:         narrow_band.insert_in_position(g_kl);
16:       end if
17:     end if
18:   end for
19:   Iterations;
20:   while narrow_band NOT EMPTY do
21:     g_ij.T ← narrow_band.pop_first();
22:     for all g_kl ← g_ij.neighbours do
23:       if g_kl = FROZEN then
24:         skip;
25:       else
26:         g_kl.T ← solveEikonal(g_kl);
27:       end if
28:       if g_kl.state = NARROW BAND then
29:         narrow_band.update_position(g_kl);
30:       end if
31:       if g_kl.state = UNKNOWN then
32:         g_kl.state ∈ NARROW BAND
33:         narrow_band.insert_in_position(g_kl);
34:       end if
35:     end for
36:   end while
37: end for
```

labeled as Frozen. Later, its Manhattan neighbours are labeled as Narrow band and T is computed for each of them. In the loop stage, each iteration of the loop solves the Eikonal equation for each Manhattan neighbour (which has not been Frozen yet) of the Narrow band cell with a lesser T value. Then this cell is labeled as Frozen. It is noted that the Narrow band maintains an order from the lowest value to the highest value. The algorithm finalizes when all the cells are labeled as Frozen. This process is shown in Fig. 2.2(a) (Valero-Gómez et al. (2013)).

The black points represent the cells labeled as Frozen. The grey points are the Narrow band and the white cells are Unknown. It is noted that the expansion over the cells is like a natural movement of a wave. Figure 2.2(b) (Valero-Gómez et al. (2013)) shows the FM method application for a 3D environment, where one only minimum is appreciated at the source.

The result is a distance map, also called times-of-arrival map. A path is obtained by the calculation of the geodesic, that is to say, the shorter path between two points, where the source point is T = 0 (minimum global), being the rest of the points T > 0 (Arismendi et al. (2015)).

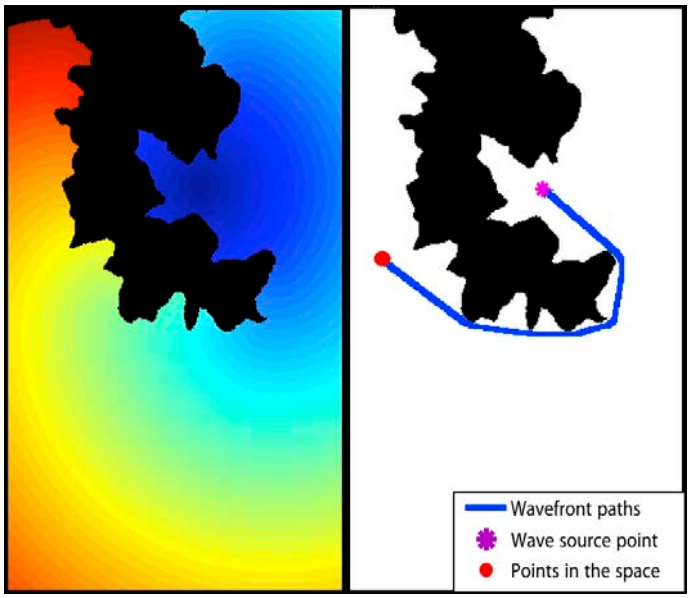


Figure 2.1: *Application of the FM method: (a) the time of arrival map; (b) the resulting path in the initial map.*

It should be noted that the FM algorithm generates optimal trajectories in terms of time, but its curves suffer abrupt changes and are very close to the obstacles (see Fig. 2.1(b)). This is due to the fact that the geodesic loses their smoothness when abrupt changes are produced in the velocity wave (Gómez et al. (2015a)). The FM^2

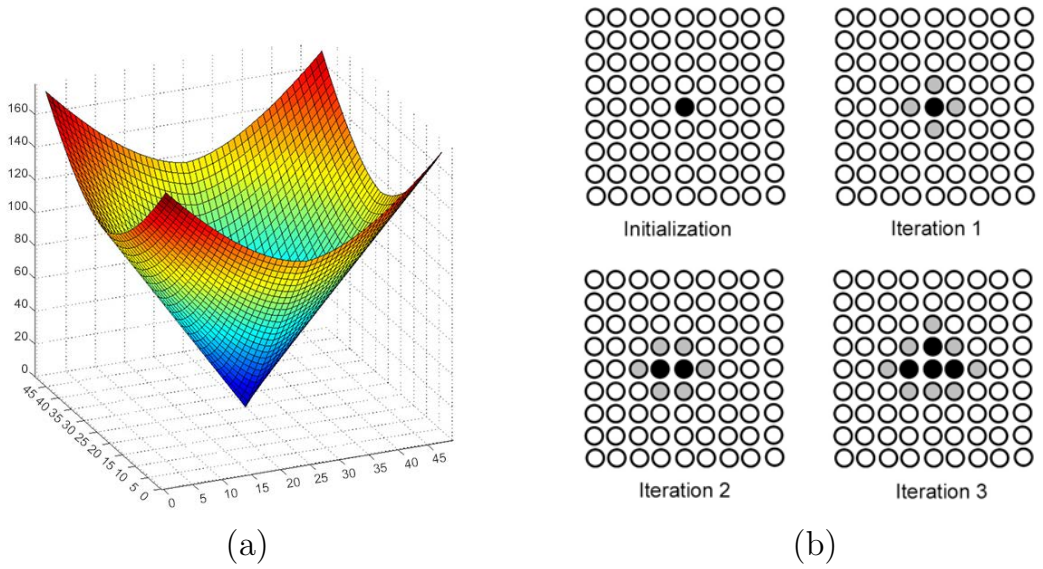


Figure 2.2: *Process of wave expansion of the Fast Marching method: (a) iterative wave expansion; (b) Fast Marching method in a 3D map.*

method solves these problems, as explained next.

2.2 Fast Marching Square Method

The FM^2 was introduced by Garrido et al. in 2009 (Garrido et al. (2009)) and consists on applying the FM method twice. This method solves the problem mentioned in the previous section, providing paths with an adequate smoothness and sufficient safety distances from the obstacles. The following procedure describes how the FM^2 method works:

1. The 3D environment used as an input, W_0, is read as a binary grid map (see Fig.2.3(a)). The cells belonging to obstacles are labeled in black (value 0) and the cells of free space are labeled in white (value 1).

2. The FM method is applied over the binary map (Fig. 2.3(a)) using each cell belonging to the obstacles as wave source, expanding several waves at the same time. In this way, each cell of the map acquires a certain value, which is the time that the wave takes to reach each point of the space. The resulting map is rescaled to fix the maximum cell value as 1. Then, each cell has a value between 0 and 1. The resulting map is a potential field of the original map denominated velocities map (W), as shown in Fig. 2.3(b). This map is denoted so because the value of each cell is proportional to the distance from

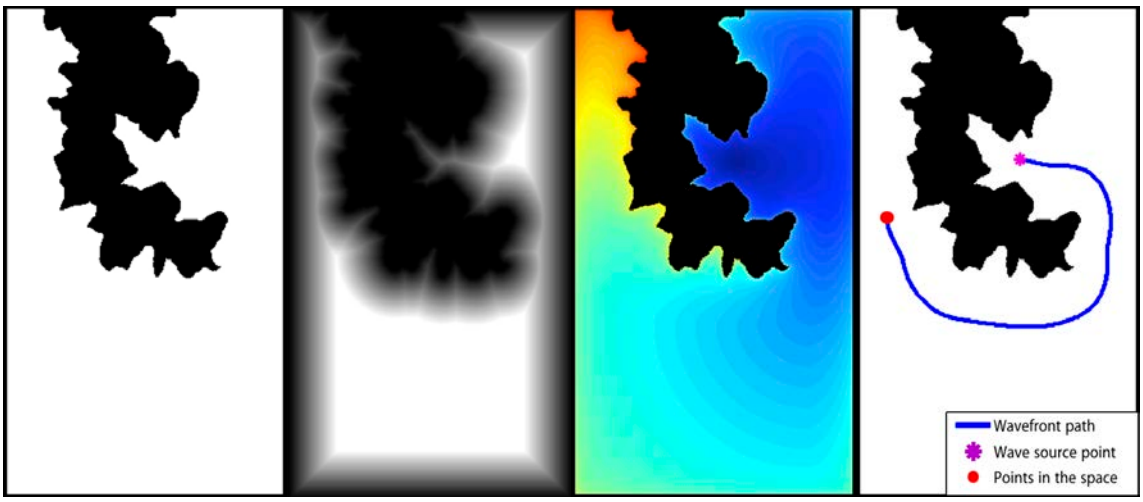


Figure 2.3: *Application of the FM^2 method: (a) map read as binary gridmap; (b) the velocities map W; (c) the time of arrival map T; (d) the resulting path in the initial map.*

obstacles and it can be interpreted as the velocity of the vehicle at the same time. So, W provides the maximum admissible speed at each point of the environment. That is to say, if the vehicle is near the obstacles, the admissible speed will be less than if the vehicle is away from the obstacles.

3. The FM algorithm is applied again from the goal point, using this point as wave source. The wave is expanded over the map W until the start point is reached. Figure 2.3(c) shows the time of arrival map T as the result of this process.

4. The gradient descent is applied over T from the start point to the goal point obtaining the optimal path in terms of smoothness and safety, as shown in Fig. 2.3(d).

The FM^2 method has a low computational cost and can be used: 1) as a local planning method, for instance in the cases of obstacles avoidance in a dynamic environment (Gómez et al. (2013)), and 2) as a global planning method, with path planning purposes. In both types of applications, the algorithm will find the optimal solution in terms of safety and smoothness.

Though the local planning method is not the case of study presented in this thesis, for dynamic environments the algorithm would need to be executed several times with a specific time interval between executions, depending on the rate of change of the environment.

2.3 Modified Fast Marching Square Method: Adjustment Parameter

Although the paths generated by the FM^2 are good in terms of safety and smoothness, those paths are often not sufficiently optimal for the proposed mission. For this reason, we propose to introduce an adjustment parameter that modifies the planning according to the requirements of the mission. The procedure is explained next.

Each cell of the map W is raised to this adjustment parameter, increasing or decreasing its value. This causes a lightening or darkening of each cell in accordance with the value of that parameter. The greater the adjustment parameter, the greater the darkening of the map W, causing paths further away from the obstacles. On the contrary, if the value is lower, the map W is clarified and this will generate smoother paths closer to the obstacles. This process is illustrated in Fig. 2.4.

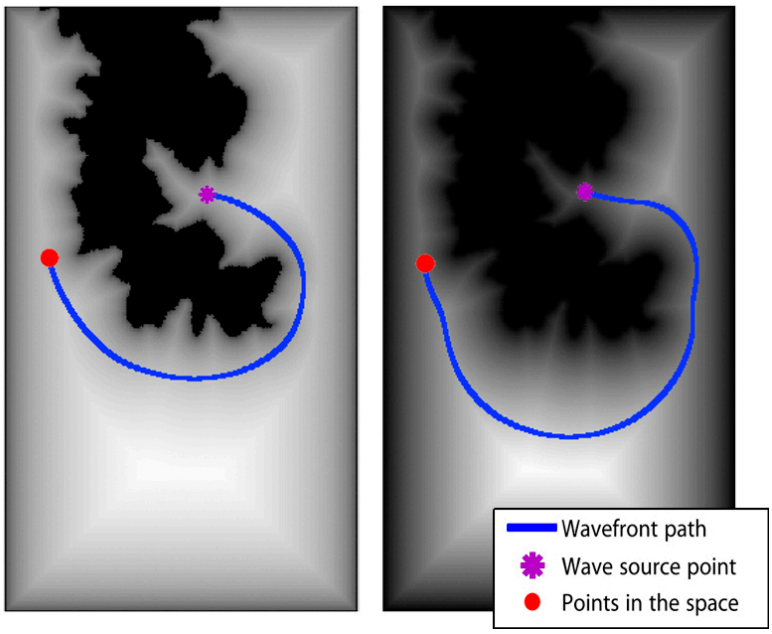


Figure 2.4: *The map W raised to different values and their respective paths.*

The map W is saturated according to the different constraints that should be taken into account when the planning is carried out. In our case, W is saturated to obtain certain security margins and smoothness in the path. So, the speed is also modified due to this saturation. If the map is obscured, the permissible speed of the vehicle will be lower than if the map is clarified. According to this, the speed of the UAV is a consequence of the treatment of the map.

The map W can be modified to attend to other optimization criteria, as demonstrated in Garrido et al. (2013) and (Garrido et al. (2015). One of the cost functions to be implemented could be related to fuel optimization, for instance.

2.4 Conclusions

This chapter has presented a summary of the FM and FM^2 methods used as the base for the planner approach developed in this thesis. The introduction of an adjustment parameter has been discussed, whose mission is to modify the velocity map according to different flight constraints. Next chapter illustrates a case study of the application of the modified FM^2. Later, in Chapter 4, as another novel approach of this thesis, two adjustment parameters p_1 and p_2 will be introduced to govern kinematic and flight level restrictions during the flight.

Chapter 3

Modified Fast Marching Square Approach: A Case Study

In this chapter we present a case study of the application of the modified FM^2 method for the mission planning to be carried out by a UAV in a 3D urban environment. The Dubins airplane model is used to check if the path resulting from the FM^2 is feasible, considering constraints in flight such as turning rate, climb rate and velocity. All these constraints are considered for a fixed-wing aircraft. Thanks to this approach, the trajectory generated by the FM^2 will be perfectly feasible for a UAV, always considering its constraints and the static obstacles of the environment. Two examples of application will be shown to demonstrate the good performance of the approach, and the comparison with the Rapidly exploring Random Tree (RTT) algorithm will be discussed.

3.1 Dubins Airplane Kinematic Model

The kinematic model used here to validate the feasibility of the path resulting from the FM^2 algorithm was proposed by Dubins (Dubins (1957)). This model defines the minimal paths with bounded curvature between two configurations. It was proposed in a two-dimensional Euclidean space, where the mobile vehicle advances with a given minimum turning radius at constant speed. This model was denominated Dubins car model and its movements equations are as follows:

$$\dot{x} = V\cos\psi, \quad \dot{y} = V\sin\psi, \quad \dot{\psi} = u, \tag{3.1}$$

where ψ is the heading angle, V is the constant speed of the vehicle and u is the control input of the vehicle. So, the turning rate $\dot{\psi}$ is limited $\left|\dot{\psi}\right| < \dot{\psi}_{max}$ (McGee et al. (2005)).

When the problem is related to aircrafts, the previous model is suitable if there are no changes of altitude. If the trajectory of the aircraft demands altitude changes, the third component of the three-dimensional Euclidean space is required (Choi (2014)). The Dubins airplane model was introduced by Chitsaz and LaValle (Chitsaz and LaValle (2007)) and includes a climb rate limitation. This model was modified to obtain a more reliable model within a real environment (Beard and McLain (2013)).

The movement equations are as follows:

$$\dot{x} = V\cos\psi\cos\gamma, \quad \dot{y} = V\sin\psi\cos\gamma, \quad \dot{z} = -V\sin\gamma, \tag{3.2}$$

where $\dot{\psi} = \frac{g}{V}\tan\phi$. A new component $\dot{\gamma}$ is the flight-path angle of the aircraft. The airplane's bank angle $\dot{\phi} = u_1$ is also considered in the turning rate (Beard and McLain (2012)). The kinematic model of the airplane proposed in Beard and McLain (2013) is illustrated in Fig. 3.1. However, in the approach presented in this work only the angles ψ and γ are taken as control inputs, without considering the g (the gravity acceleration) and ϕ components: $\dot{\psi} = u_1$ and $\dot{\gamma} = u_2$.

3.2 Path Planning and Trajectory Verification

As introduced in the previous chapter, the approach presented in this work uses the FM^2 path planning method to plan missions for a UAV in a 3D urban environment. The Dubins kinematic control model is used to verify if the planned path is feasible for the UAV. This section presents the methodology followed.

The environment map W_0 where the simulation is carried out is shown in Fig. 3.2. The simulated environment used in our case study is a 3D grid map, which lodges 15 buildings of different dimensions. Each cell of the grid map is equivalent to 5 meters. All the buildings have square shapes and their heights are within a range from 220 to 390 meters, while their widths are within the range from 20 to

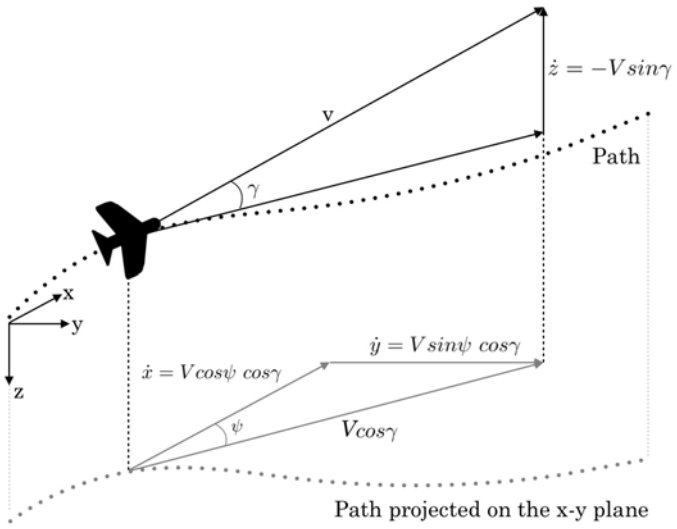


Figure 3.1: *Illustration of the Dubins airplane model.*

60 meters. The scenario is read as a binary map, where the buildings are labeled as the obstacles and the rest of the map is labeled as the free space. The FM method is applied over W_0 and the resulting map W has a different value in each cell related to the distance from obstacles. As a novel approach, after scaling the map, each cell is raised to a power causing different types of trajectories, modifying the safety distance from the obstacles and the smoothness. These trajectories are analyzed in the simulation section. The distances of this map are proportional to the maximum permissible speed of the UAV in each point having a value between 0 and 1, where 0 corresponds to the stall speed and 1 corresponds to the maximum speed. The FM method is applied again from the goal point to the start point over the potential W. This creates a cost map T, where a single global minimum exists. T is also denominated time of arrival map, since each cell corresponds to the time of arrival of the wavefront expansion.

Following this approach, the Dubins airplane model is used to check if the resulting path is feasible for the UAV. When the kinematic control model is applied over the planned path, a realistic execution of the planned path is obtained taking into account the turning rate, climb rate and constant velocity of the aircraft.

In the Dubins airplane model the position of the aircraft is defined by the components x, y and z belonging to the Euclidean space $\mathbb{R}^3$. The heading angle of the aircraft ψ is the angle formed by the x axis and the longitudinal axis of the aircraft (Chitsaz and LaValle (2007)). The angle formed by the longitudinal axis of the aircraft and the $x - y$ plane is denoted as the flight-path angle γ (Beard and McLain (2013)) (see Fig. 3.1). The final path executed by the UAV depends on its kinematic model. It is of vital importance to obtain the orientation of the aircraft in

each point of the planned path resulting from the FM^2 algorithm. The heading angle and flight-path angle are calculated for any point $p\,(x_i, y_i, z_i)$, for $i = 1, 2, ..., n-1$, where n is the current total number of points in the trajectory:

$$\psi_i = atan2(y_{i+1} - y_i,\ x_{i+1} - x_i), \tag{3.3}$$

$$\gamma_i = 90 - \cos^{-1}\left(\frac{z_{i+1} - z_i}{V_i}\right), \tag{3.4}$$

where $V = |v|$. Nevertheless, in order to anticipate the turn in the curves and to avoid the UAV from not following the planned path, it is essential to anticipate the movement. Thus, if the actual position of the aircraft is p_i, the next position to consider will be p_{i+a}, where a will be the number of positions ahead from the current position. Hence, Eq. (3.3) and Eq. (3.4) can be rewritten as

$$\psi_i = atan2(y_{i+a} - y_i,\ x_{i+a} - x_i), \tag{3.5}$$

$$\gamma_i = 90 - \cos^{-1}\left(\frac{z_{i+a} - z_i}{V_i}\right), \tag{3.6}$$

The parameter a is selected depending on the number of total points of the path and the necessary distance to anticipate the movement.

The minimum turning radius for the Dubins airplane model is given by

$$\rho = K_{p\psi} e(\psi), \tag{3.7}$$

where $K_{p\psi}$ is a proportional constant and $e(\psi)$ the error between ψ in the current position ψ_i and the next point of the path $\psi_{(i+1)}$.

The flight-path angle is also bounded as

$$\varphi = K_{p\gamma} e(\gamma), \tag{3.8}$$

where $K_{p\gamma}$ is another proportional constant and $e(\gamma)$ the error between γ in the current position γ_i and the next point of the path $\gamma_{(i+1)}$.

It should be noted that in certain cases where the values of ρ and φ are high, a saturation can be produced. This mainly depends on the conditions at which the path is generated, such as the size and the obstacles distribution in the environment. In our case, the limitation of ρ and φ has not been considered.

According to ρ and φ, Eqs. (3.2) can be rewritten as:

$$\dot{x} = V\cos(\psi - \rho)\cos(\gamma - \varphi),\ \dot{y} = V\sin(\psi - \rho)\cos(\gamma - \varphi),\ \dot{z} = V\sin(\gamma - \varphi). \tag{3.9}$$

These equations are valid when the start point is on the left of the goal point. In the opposite case, if the start point is on the right of the goal point, Eqs. (3.9) can be rewritten as

$$\dot{x} = -V\cos(\psi-\rho)\cos(\gamma-\varphi),\ \dot{y} = -V\sin(\psi-\rho)\cos(\gamma-\varphi),\ \dot{z} = V\sin(\gamma-\varphi). \tag{3.10}$$

3.3 Simulation Results

The approach described above has been implemented in the 3D urban environment described in the previous section. The approach takes the model of the environment, generates the path using the FM^2 method avoiding the buildings and checks the planned path using the Dubins airplane model, obtaining a feasible path (achievable by the UAV) where the aircraft constraints are taken into account. The planned path (FM^2 path) and the feasible path (Dubins path) are compared to verify to what extent the planned path is feasible. It is noted that the smoother the path, the higher the precision to follow it. For each mission, the initial and final points (positions) of the trajectory are given to the method.

Two different experiments are discussed here. On the one hand, the planning is carried out at a low altitude with respect to the floor. In this experiment the FM^2 method is also compared with the RRT algorithm, showing the benefits of using FM^2. On the other hand, the planning is performed at a high altitude, considering the option to plan over the buildings. In both cases the turning rate, the climb rate and the velocity constraints are considered in the Dubins model. In each experiment, the map W resulting from the FM^2 method is raised to different powers. In this way the path is modified depending on the safety margin between the elements of the scenario and the UAV, this affecting also the smoothness of the trajectory. The first potential map also indicates the maximum admissible speed at which the UAV will fly to execute the path. The value 0 is identified as the stall speed and the value 1 as the maximum speed. In the simulations, the chosen speed for the Dubins model is a constant speed, which is interpreted as the cruise speed of the aircraft.

3.3.1 Low Altitude Experiment

The main purpose of this experiment is to plan the optimal path for the UAV between the buildings at a low altitude. Three cases are described providing different trajectories according to the safety distance from the buildings. The first potential map W will be raised to three different powers in order to analyse the effect on the path.

When W is raised to the power of 1, the planned path obtained by the FM^2 method is shown in Fig. 3.2(a). The path has very abrupt curves and maintains

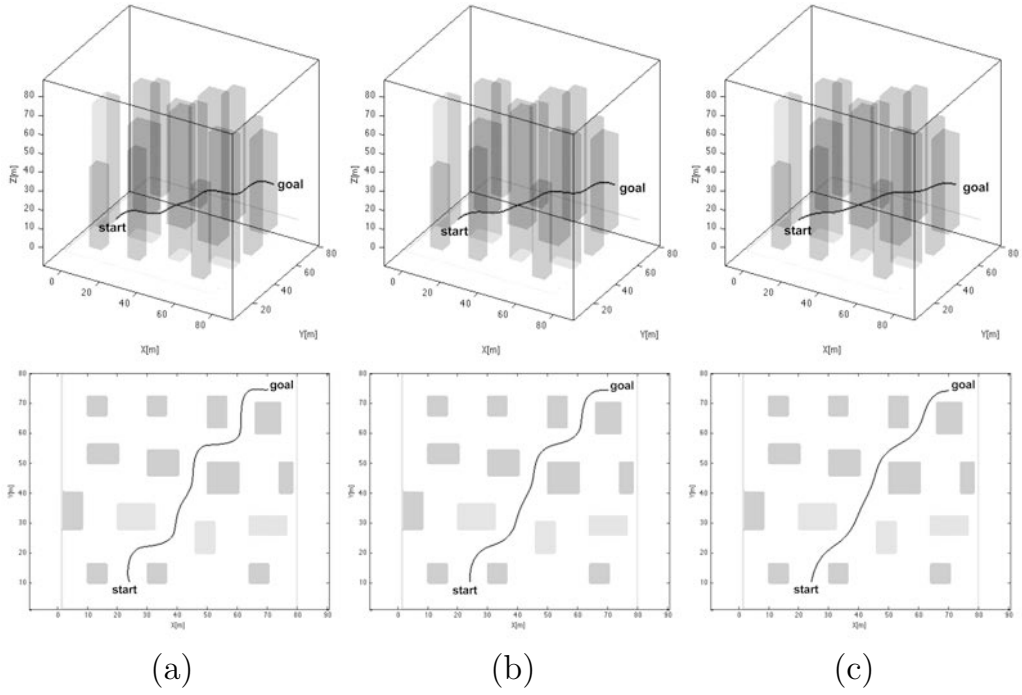


Figure 3.2: *Planned paths at low altitude: (a) path with W^1; (b) path with $W^{1/2}$; (c) path with $W^{1/4}$.*

a safety distance of approximately 20.4 meters from the buildings. In Fig. 3.3(a) the feasible path is shown, which is generated when the Dubins airplane model is applied. The feasible path moves away from the planned path in the curves a maximum distance of approximately 2.1 meters. This is due to the turning rate and velocity constraints defined in the kinematic model.

If the map W is raised to a power of 1/2, the planned path is smoother than in the previous case, presenting less abrupt curves. As shown in Fig. 3.2(b), the safety distance has decreased to a value of 15.8 meters. There is also a smaller separation between the planned path and the feasible path in the curves, reducing the value to 1.6 meters (see Fig. 3.3(b)).

In the third case, where the map W is powered to a value of 1/4, the planned path is far smoother than in the previous cases, as shown in Fig. 3.2(c). The trajectory has a shorter safety distance from the obstacles, around 10.3 meters. The maximum separation between the planned path and the feasible path is also smaller, approximately 1.3 meters (see Fig. 3.3(c)).

Figure 3.4(a) shows the velocities profiles resulting from the FM^2 method in each of the cases defined above. The maximum permissible speed changes according to the smoothness of the path. That is, if the path is very smooth, the maximum

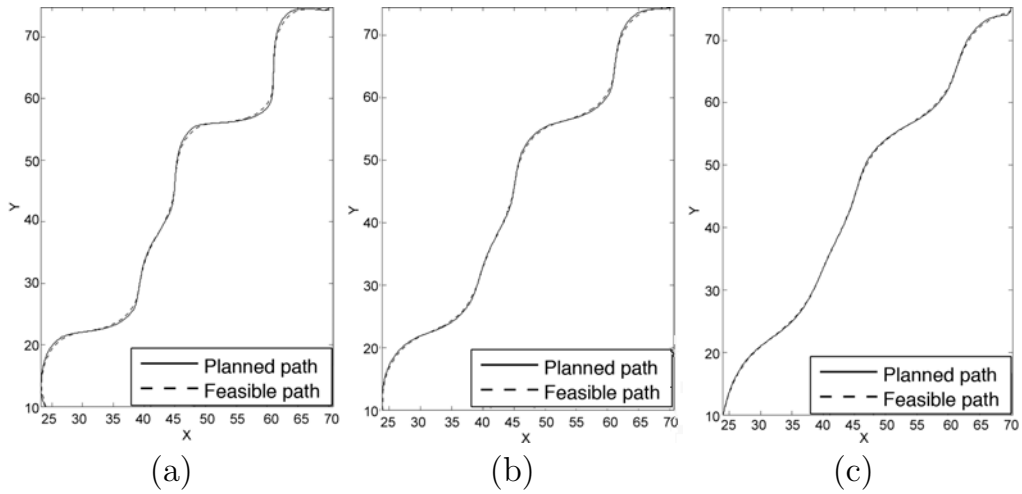


Figure 3.3: *The planned path (FM^2) and the feasible path (Dubins model) at low altitude: (a) path with W^1; (b) path with $W^{1/2}$; (c) path with $W^{1/4}$.*

Table 3.1: *Safety distances and separation distances at low altitude.*

	Safety distance (m)	Separation distance (m)
W^1	20.4	2.1
$W^{1/2}$	15.8	1.6
$W^{1/4}$	10.3	1.3

permissible speed at each point is greater than if the path has abrupt curves. It is also appraised that in the points belonging to the curves, the maximum permissible speed is smaller than in the straight stretches. Consequently, if the UAV flies with a constant speed (for instance, the average between the maximum and minimum speed of the profile), the UAV will tend to diverge from the planned path in the points where its speed is greater than the permissible speed.

Table 3.1 summarizes the safety distance from the planned path to the buildings and the separation distance between both paths (planned and feasible) in each case.

The computational time of the planning in all cases is about 1.2 seconds.

RRT and FM^2 Algorithms Comparison

Before proceeding with more experiments, the RRT algorithm available in Clifton et al. (2008) and the FM^2 algorithm are compared to demonstrate the benefits of using the latter.

The RRT algorithm is implemented under the same conditions imposed in the low altitude experiment, that is to say, same environment and same initial and final points. Figure 3.5 shows the result obtained from applying the RRT algorithm,

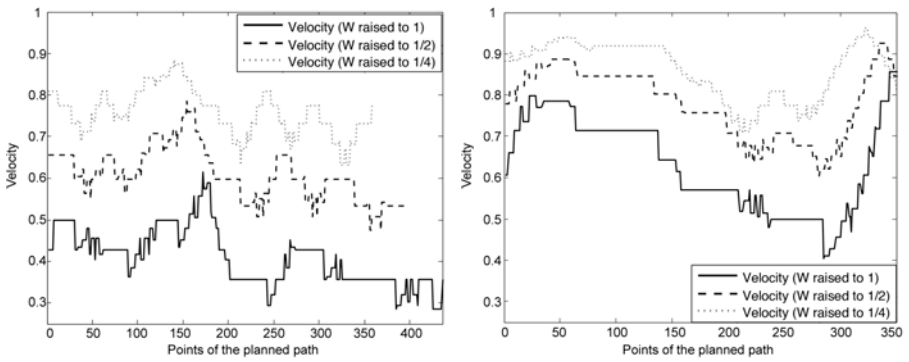


Figure 3.4: *Velocities profiles at different altitudes: (a) velocities profiles of the trajectories at low altitude; (b) velocities profiles of the trajectories at high altitude.*

where the black line represents the trees, the red one represents the initial path, and the green one represents the smoothed path. The generated path does not respect any security margin and does not maintain a smoothness, producing very sharp curves. The necessary time to calculate the path is about 5 seconds, while the FM^2 method generates a path in about 1.2 seconds. This happens because, after planning with the RRT algorithm, it is necessary to smooth the path, which requires more iterations and therefore, more time. This demonstrates that under the same conditions, the RRT algorithm generates paths of inferior quality in terms of security and with a very high computational cost, since many more iterations are required to generate a smooth trajectory.

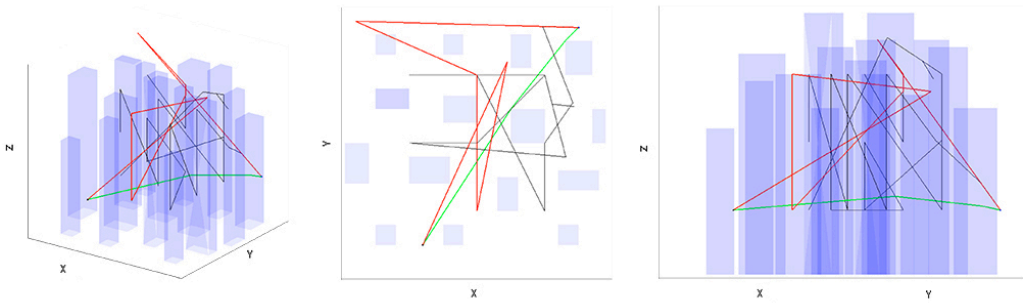
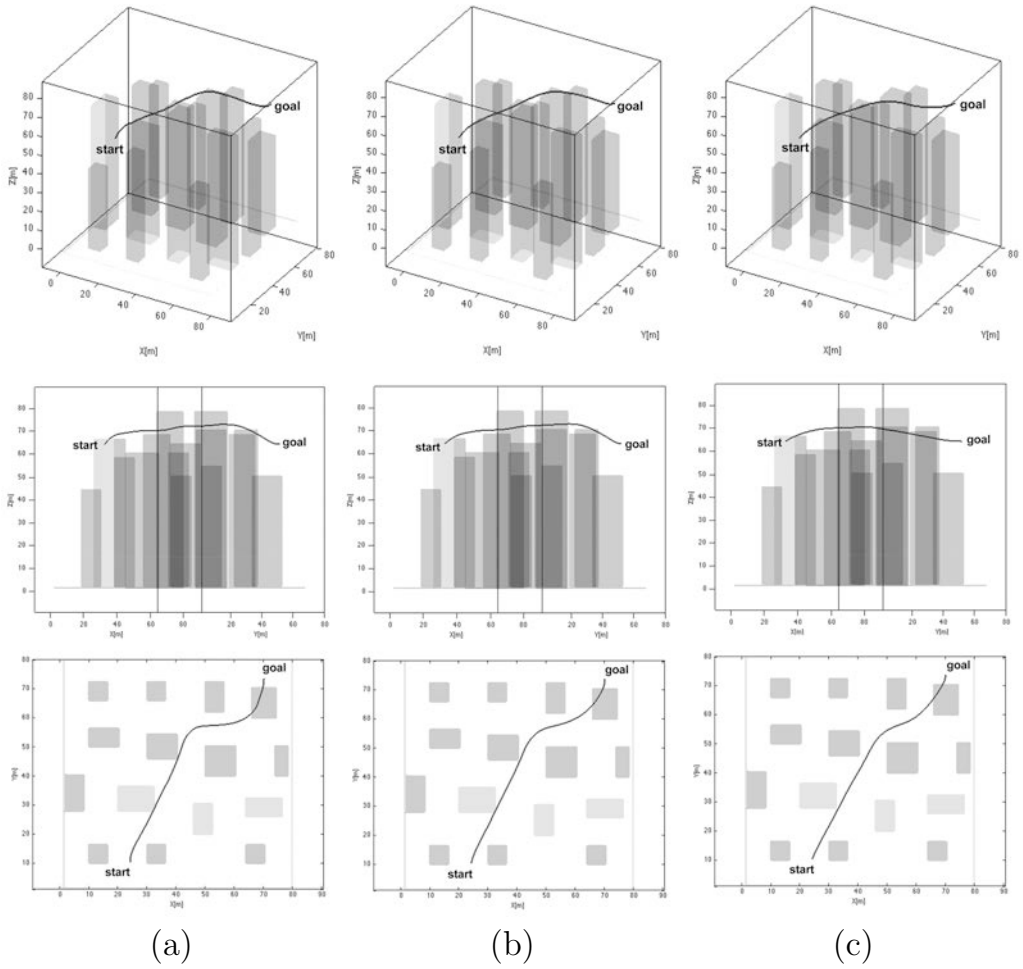


Figure 3.5: *Path generated by the RRT Algorithm. Black line represents the trees; red line represents the initial path; green line represents the smoothed path.*



(a) (b) (c)

Figure 3.6: *Planned paths at high altitude: (a) path with W^1; (b) path with $W^{1/2}$; (c) path with $W^{1/4}$.*

3.3.2 High Altitude Experiment

In the high altitude experiment, the main purpose is to plan the optimal path in order for the UAV to fly over the buildings when the altitude parameter comes into play. Unlike the previous experiment, here there are two perspectives: $x - y$ plane and $z - xy$ plane. The same three cases as in the low altitude experiment will be studied.

For the first case, when W is raised to the power of 1, the path shown in Fig. 3.6(a) is planned. In the perspective of the $z - xy$ plane (Fig. 3.7(a)) the planned path presents sudden changes of altitude. For the perspective $x - y$, Fig. 3.8(a) shows a path with several noticeable curves. The feasible path moves away from the

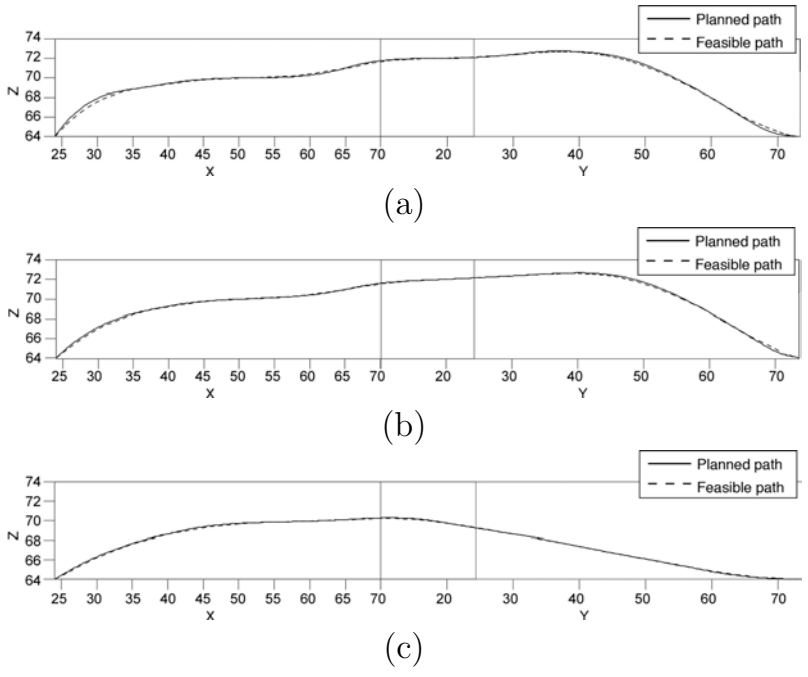


Figure 3.7: *Paths at high altitude in the $z-xy$ plane: (a) path with W^1; (b) path with $W^{1/2}$.;(c) path with $W^{1/4}$.*

planned path a maximum value of 1.6 meters and the safety distance over the buildings is rather high, around 30.4 meters, approximately. These distances mentioned above are expressed in the 3D map.

In the second case, when W is raised to the power of 1/2, the planned path is shown in Fig. 3.6(b). Figure 3.7(b) illustrates the perspective in the $z-xy$ plane. The planned path is softer and has less sudden changes than in the previous case. Figure 3.8(b) shows the perspective in the $x-y$ plane. It is appreciated that the path in this plane is also softer than in the previous case. The separation between both paths is reduced to around 0.8 meters and the safety distance from the buildings is around 26.5 meters.

The last case is shown in Fig. 3.6(c), when W is raised to the power of 1/4. Figure 3.7(c) illustrates the path in the $z-xy$ plane. In contrast with the previous cases, this planned path is very smooth. Figure 3.8(c) also shows a path smoother than in the previous cases. The distance between both trajectories is 0.7 meters, approximately, and the distance from the obstacles is around 20.1 meters above the buildings.

Figure 3.4(b) shows the velocities profiles of the three latter cases. In general, these three paths are smoother than in the low altitude experiment. As a consequence, these velocities profiles have higher values. The values of the permissible

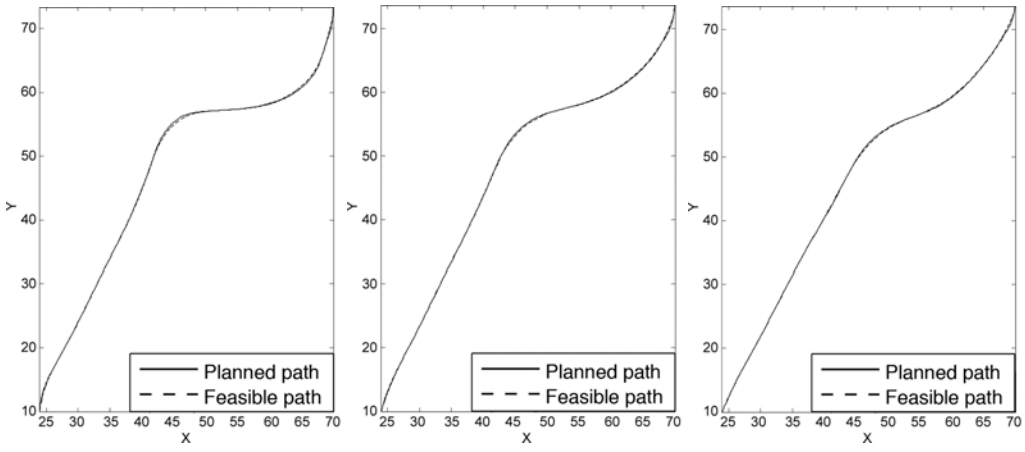


Figure 3.8: *The planned path (FM^2) and the feasible path (Dubins model) at high altitude in the $x-y$ plane: (a) path with W^1; (b) path with $W^{1/2}$; (c) path with $W^{1/4}$.*

Table 3.2: *Safety distances and separation distances at high altitude. Distances expressed in the 3D map.*

	Safety distance (m)	Separation distance (m)
W^1	30.4	1.6
$W^{1/2}$	26.5	0.8
$W^{1/4}$	20.1	0.7

speed also present a descent in the points belonging to the curves of the path. Table 3.2 resumes the distance from the buildings and the separation distance between the planned path and feasible path in each case. One can observe that the lower the value of the raised power, the smaller the distance from the buildings and the smoother the path.

In all cases, the computational time of the planning is approximately 1.2 seconds.

3.4 Conclusions

This chapter has presented a case study of the application of the modified FM^2 method for the mission planning to be carried out by a UAV in a 3D urban environment. The planning method uses the FM^2 algorithm as a planner and the Dubins airplane model to verify if the planned path is feasible for the UAV. It takes into account the UAV physical constraints, such as turning rate, climb rate and cruise speed. In this way, it is possible to compare safety distances from the obstacles and the separation distances between the planned path (FM^2) and the feasible path

(Dubins model). In each proposed experiment three different cases have been considered where the first potential map is raised to different powers, causing distinct effects on the path.

It has been proven in our experiments that the lower the value of the raised power, the greater the smoothness of the trajectory. So, when this happens, the separation distance between the planned path and the feasible path is smaller. That is to say, the lower the value of the raised power, the greater the precision to follow the path. In this way, it is possible to generate trajectories perfectly competitive with the Dubins paths.

However, it should be noted that if our approach is applied in other types of maps, such as environments with a single skinny obstacle, the relation between W and the adjustment parameter can vary. If the adjustment parameter is decreased, the resulting path may be closer to the obstacle, causing a smaller turning radius. However, the path will continue to have a certain smoothness, since the method is based on the expansion of a wave.

The novelty of our approach is that, whatever the constraints of the planning problem, the algorithm provides a high flexibility, since the adjustment parameter can be varied allowing the modification of W according to security and smoothness requirements.

It has also been demonstrated that the FM^2 algorithm has a low computational time in comparison with the RRT algorithm.

The low complexity of our approach makes it easy to adjust the smoothness and safety of the path thanks to the introduction of the power parameter into the potential map, which makes the algorithm very suitable for a wide range of kinematic restrictions.

Chapter 4

Fast Marching Square Approach with Imposition of Flight Level

This chapter presents a novel approach based on the FM^2 method to plan a trajectory for a UAV maintaining a given flight level with respect to the ground. Two adjustment parameters p_1 and p_2 will be used to force the path planning in specified areas of the 3D map determined by a given flight level.

Simulation results will be presented and discussed to demonstrate to robustness of the method proposed both in open field environments with non-uniform terrains and in urban environments.

4.1 Problem Statement

The problem statement is divided into two different issues: first, the environments where the mission is carried out are presented; second, the mission planning (goal) is described.

4.1.1 Environments

The environments where the mission planning is carried out are shown in Fig. 4.1 for open field test and and Fig. 4.2 for urban environment test.

Figure 4.1 represents an open field with mountainous terrain where the surface is rather uneven, this being a 3D grid map with dimensions 120 x 90 x 40 cells. Each cell of the map is equivalent to 15 x 15 x 15 meters. In this case, it is not necessary to consider the size of the UAV, since it is assumed to be smaller than the size of a cell. In the case that the UAV exceeds the size of the cell, the obstacles of the map can be swelled according to the radius of the vehicle, as demonstrated in Gómez et al. (2013).

Figure 4.2 represents an urban environment that lodges 15 buildings of different dimensions. Each cell of the grid map is equivalent to 5 meters. All the buildings have square shapes and their heights are within a range from 220 to 390 meters, while their widths are within the range from 20 to 60 meters.

4.1.2 Mission

Our mission requires a UAV moving throughout an open field or urban environment, flying from a start position to a goal position, avoiding any hill in the terrain or building in the city and maintaining a fixed flight level over the ground.

An approach based on the FM^2 method is used to find a path that joins two points with different configurations. This approach generates a path composed of a set of consecutive points such as $Q = P_0, P_1, ..., P_n$. To maintain a determined flight level with respect to the ground, two adjustment parameters are employed. The values

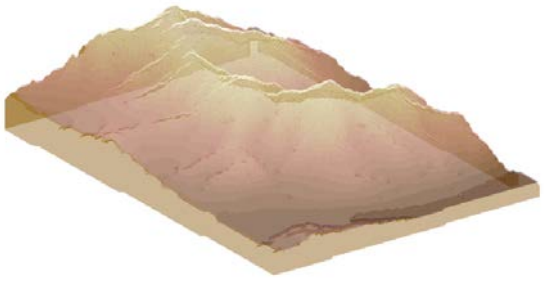


Figure 4.1: *The 3D simulated representation of the open field environment.*

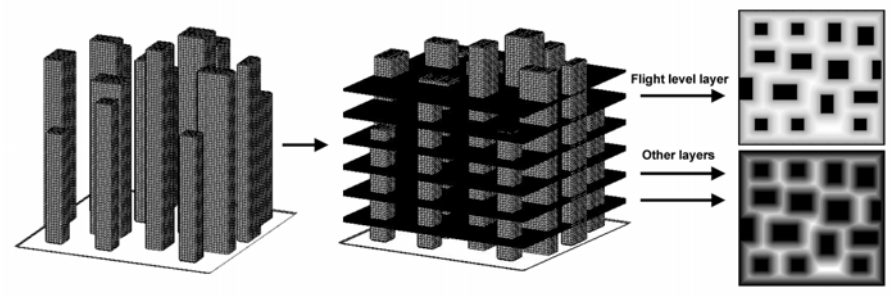


Figure 4.2: *The 3D urban map divided into layers and the 2D representation of the flight level layer and the rest of layers.*

of these parameters affect the compliance of the restriction in flight, depending on the desired smoothness and safety of the trajectory. This entire process is explained next.

4.2 Path Planning with Flight Level Constraint

This section presents our approach to carry out the path planning in a 3D open field or urban environment where a flight level is imposed over the ground. This flight level must be respected by the UAV in the trajectory, considering that the initial and final stretches do not respect this constraint due to the configuration of the start and final points of the path.

The chosen environments to perform the planning are described in Section 4.1.1 and the planner is the FM^2 method. An approach to plan trajectories with an imposed flight level over the ground is added within the FM^2 method.

As stated previously, the wave front used in the FM^2 method tends to expand by the clearer areas of the map W. Therefore, the value of the cells where we want to 'force' the planning must be close to 1. It is noteworthy that the environment map is a 3D grid map, so the different layers of the map W can be clarified or obscured as shown in Fig. 4.3.

To achieve our goal, the corresponding cells with the fixed flight level over the ground have to be clarified. Algorithm 2 presents the approach used within the FM^2 to fix a flight level with respect to the ground. For this purpose, two adjustment parameters, p_1 and p_2, are employed. The mission of each of these parameters is to clarify or darken the map W cells according to the desired planning.

The procedure to modify the value of the map W cells belonging to the given flight level over the ground is as follows:

- Firstly, the terrain elevation for each cell of the map, $Surface(i, j)$, is identified (see Algorithm 2, line 4). This elevation is computed taking as reference the layer 0 of the environment, which can be defined as sea level.

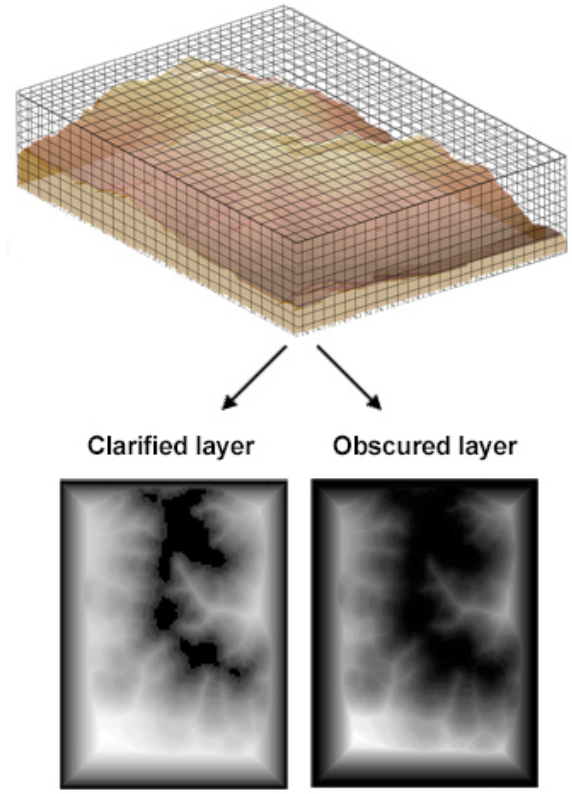


Figure 4.3: *The upper part of the image shows the 3D grid map of the environment. The lower part shows clarified and obscured layers of the map W.*

- Then, the value of the given flight level is added to the value of the terrain elevation (Algorithm 2, line 5). These cells are clarified raising their value to p_1 (Algorithm 2, line 6). Thus, the planning for these zones will be easier.

- The rest of the cells are obscured raising them to p_2 (Algorithm 2, line 8), making the planning harder through them.

The influence of the values of parameters p_1 and p_2 on the smoothness and feasibility of the path will be discussed in detail in next sections of simulation results, both in open field and urban environment.

Though this work does not deal with the uncertainty problem, our approach can be used in case of terrain uncertainties or even dynamic obstacles avoidance. A previous work by the authors (Gómez et al. (2013)) focuses on this topic.

In the case of terrain uncertainties, the procedure will depend on their characteristics. For instance, if there is a total uncertainty on a particular part of the map and that region model is not known, then one extreme alternative could be not to include that region in the planning in order to avoid passing through it. On the

Algorithm 2 FM^2 algorithm with imposition of a fixed flight level.

Initialize: The velocities map W of a gridmap G of size m x n x l.
Initialize: Flight level L_f with respect to the ground.
Initialize: 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 ← Surface(i, j)
      if k = (L_f + SurfaceValue) then
        f_{i,j,k} ← (f_{i,j,k})^{p_1}
      else
        f_{i,j,k} ← (f_{i,j,k})^{p_2}
      end if
    end for
  end for
end for
```

contrary, if the terrain uncertainties can be quantified and the uncertainty range is well defined, then that range can be included in the planning as a security margin, enlarging the corresponding areas in the map so that the algorithm treats them as enlarged obstacles.

When the uncertainties are somehow limited but cannot be defined with precision, then the algorithm can be locally executed during the flight around those regions with uncertainties, with a specific time interval between executions, depending on the rate of change of the environment (Gómez et al. (2013)). For this, the UAV must be equipped with the corresponding sensors to detect and measure the uncertainty in real time. A more detailed study of the treatment of this type of terrain uncertainty will be addressed in future works.

The approach described here has been run under the toolbox Fast Marching provided by Peyre (Peyre (2004)) using Matlab 2013a and the code for N-dimension given in Gómez et al. (2015b). The computer used is a standard one with operating system Ubuntu 14.04. Algorithm 2 for the imposition of the flight level and adjustment of the path smoothness has been programmed in Matlab in the context of the previously cited codes and run for the 3D open field and urban environments described in Section 4.1.1. The results obtained from this approach are presented next.

4.3 Simulation Results: Open Field

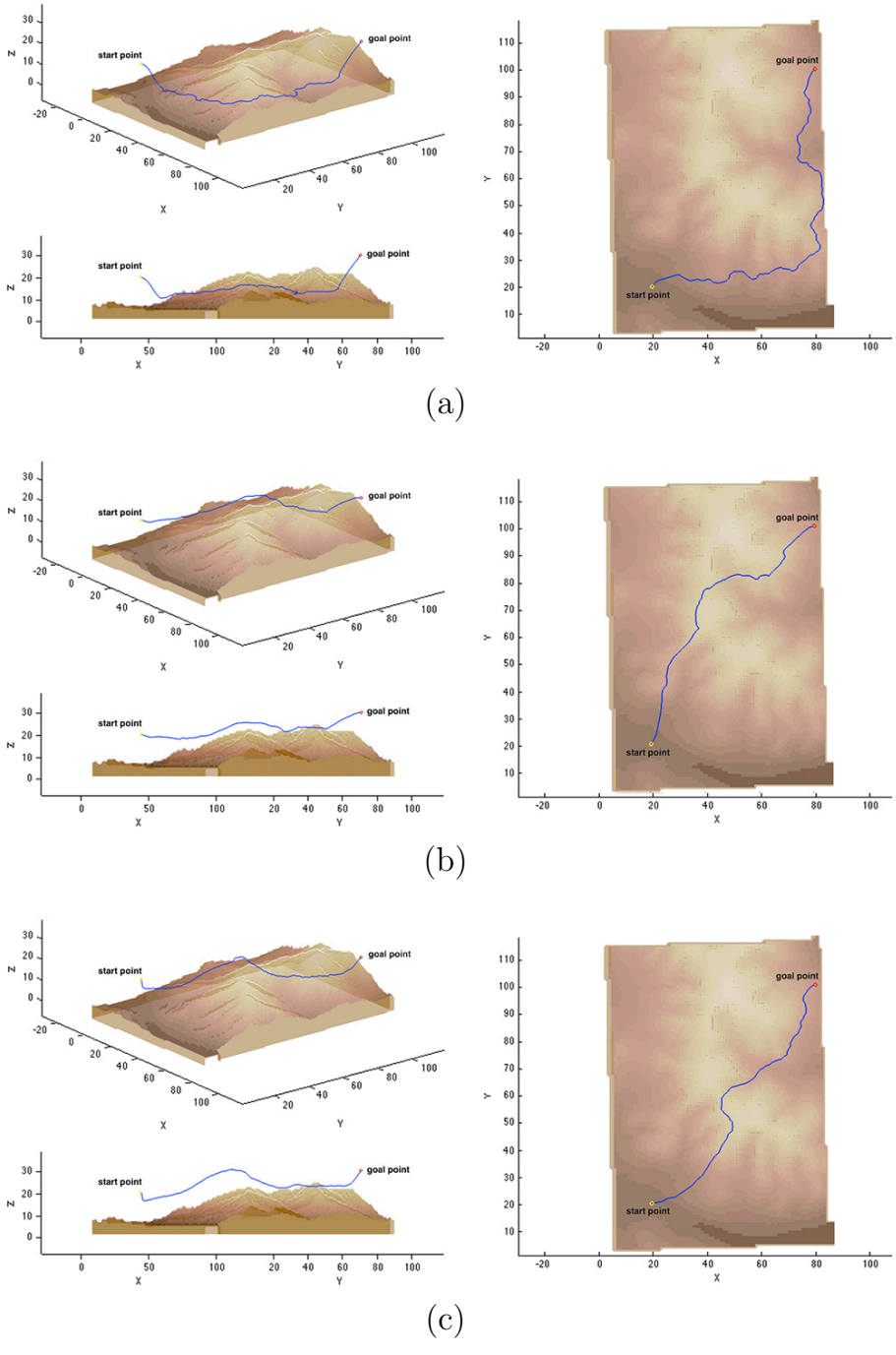


Figure 4.4: *Paths at different flight levels with respect to the ground: (a) path at flight level 2; (b) path at flight level 5; (c) path at flight level 10.*

In this section the simulation results of the proposed method when applied to the open field environment defined in Section 4.1.1 are presented and discussed.

Having taken the start and goal points of the trajectory, the approach based on the FM^2 method takes the simulated environment and plans the trajectory with a fixed flight level with respect to the ground, avoiding any hill above the terrain.

Two experiments have been carried out, where the adjustment parameters p_1 and p_2 are used to modify the map W and thus to impose a constraint over the flight level maintained by the UAV. The first experiment analyzes how the planning maintains a fixed flight level with respect to the ground. In the second experiment, the trajectory smoothness is studied according to the difference between the values of p_1 and p_2. In this latter experiment it is shown how the smoothness of the trajectory can affect the compliance of the flight level constraint, since the terrain is highly non-uniform.

4.3.1 Different Flight Levels with respect to the Ground

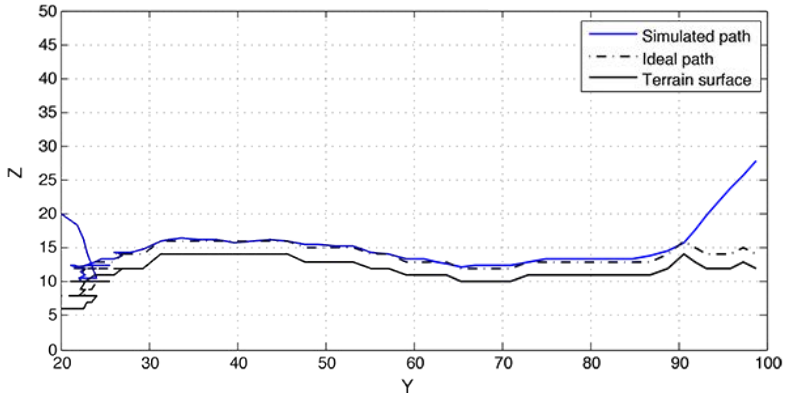
The aim of this experiment is to plan the optimal trajectories respecting a given fixed flight level with respect to the ground. Three different cases are presented in this experiment: the first case is evaluated with a flight level of 2 over the ground; the second one with a flight level of 5; and the third one with a flight level of 10. The start and goal points are the same in all the cases, being $p_s(20, 20, 20)$ and $p_g(100, 80, 30)$, respectively.

To maintain a fixed flight level with respect to the ground, it is necessary to adjust the values of p_1 and p_2, which affect the map W.

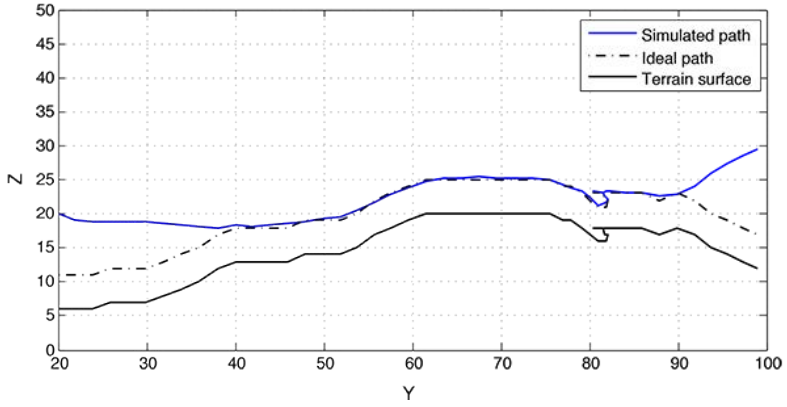
Case 1

For the first case, the values of p_1 and p_2 are adjusted to 0.5 and 2.5, respectively, for a flight level of 2. The result is shown in Fig. 4.4(a). The planned path avoids the central mountain and is planned by the lateral areas of the map, where the terrain is more uniform. That is because the value of p_2 is very high. This produces a much greater darkening of the upper layers, causing a less permissive planning.

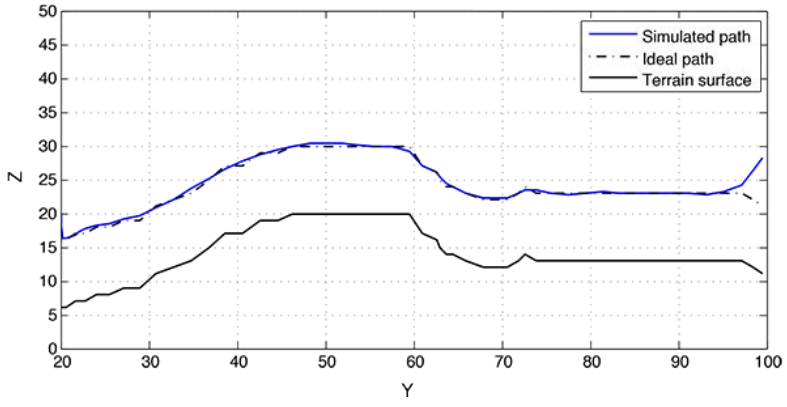
Fig. 4.5(a) shows the representation of the simulated path (resulting path from FM^2) with respect to the ideal path and the terrain in the plane $y - z$ for a flight level of 2. As can be appreciated, the difference between the simulated path and the ideal path is minimal. The only exceptions are the initial descent and the final ascent due to the position of the start and goal points, respectively, whose slopes are imposed by the smoothness of the planning.



(a)



(b)



(c)

Figure 4.5: *Comparison of the resulting path against the ideal path and the profile terrain: (a) path at flight level 2; (b) path at flight level 5; (c) path at flight level 10.*

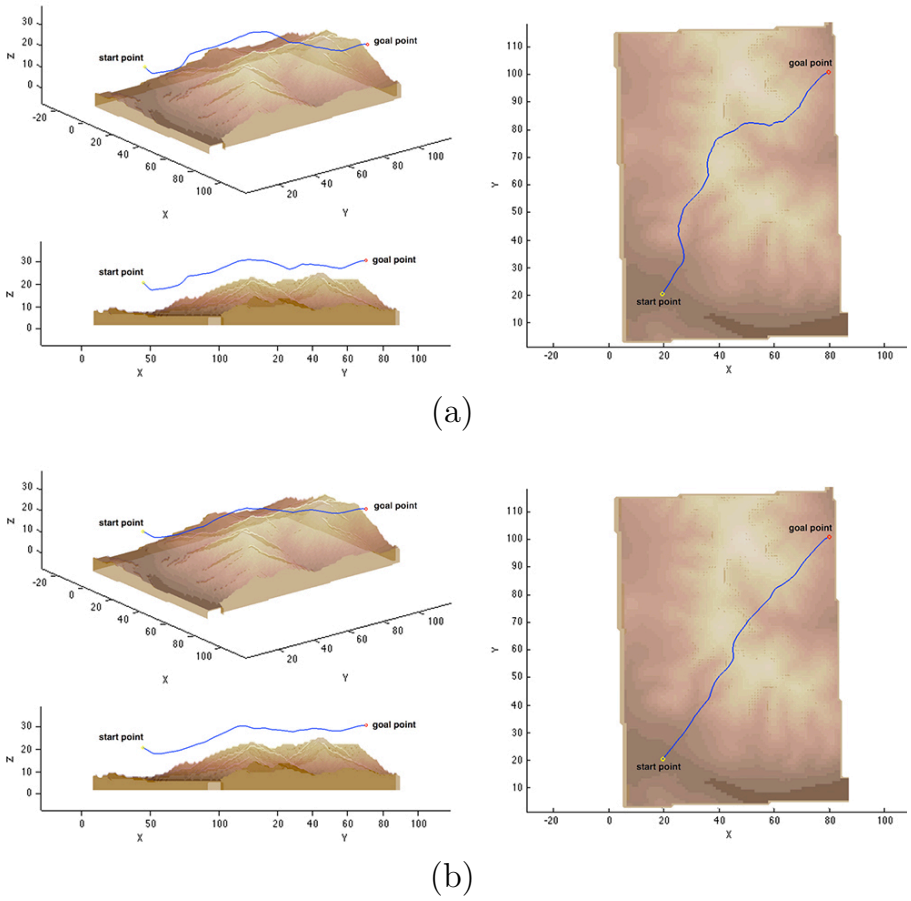


Figure 4.6: *Paths at flight level 10 with different smoothness: (a)* $p_1 = 0.5$ *and* $p_2 = 1$*; (b)* $p_1 = 0.5$ *and* $p_2 = 0.8$*.*

Case 2

In the second case, p_1 and p_2 are adjusted to 0.5 and 1.5, respectively, in order to maintain a flight level of 5. Figure 4.4(b) shows the results. In this case, the path is planned along the center of the map, since the value of p_2 is smaller than in the previous case and the planning is more permissive, allowing to plan the path through the upper layers.

Fig. 4.5(b) shows the simulated path with respect to the ideal path and the terrain in the plane $y - z$ for a flight level of 5. The simulated path tries to follow the terrain. Like in the previous case, the difference between the simulated path and the ideal path is minimal, with the exception of the stretches belonging to the descent and ascent of the trajectory at the start and goal points, respectively.

Case 3

The last case of this experiment is shown in Fig. 4.4(c), where $p_1 = 0.5$ and $p_2 = 1.3$ in order to maintain a flight level of 10. As in the previous case, the trajectory is also planned through the center of the map.

Fig. 4.5(c) shows the simulated path with respect to the ideal path and the terrain in the plane $y - z$ for a flight level of 10. The altitude variation is smaller than in the previous cases, even at the initial and final stretches of the path, since the smoothness of the trajectory is more compatible with the planning requirements.

As a conclusion, it can be seen that in all cases the flight level is respected with small variations, depending of the smoothness imposed by the planning.

4.3.2 Different Smoothness of the Trajectory

This second experiment analyzes how the trajectory can be smoothed according to the difference between the values of parameters p_1 and p_2. Here, two cases are studied where the variation of the trajectory is appreciated when modifying the values of the adjustment parameters. The start and goal points are the same as in the first experiment and the trajectory is planned at a flight level of 10.

Case 1

For this first case, the value of p_1 has been maintained to 0.5 and the value of p_2 has been reduced to 1.0. The result is shown in Fig. 4.6(a). If this figure is compared with Fig. 4.4(c), it can clearly be seen that the planning is smoother. The planning is carried out through zones where the terrain is less pronounced. Furthermore, the curves and the ascents and descents are also less sharp.

Case 2

In the second case of this experiment, the difference between the values of the adjustment parameters has been decreased, being $p_1 = 0.5$ and $p_2 = 0.8$. In this case, the trajectory is much smoother than in the previous cases, as can be seen in Fig. 4.6(b). The curves and the altitude changes are less sharp.

Fig. 4.7 shows the simulated path against the ideal path and the terrain for a flight level of 10 for both experiments. The resulting paths travel through different areas of the map, and the turn angle, climb angle and curvature of the trajectories also vary, as demonstrated in Fig. 4.8, Fig. 4.9 and Fig. 4.10, respectively. As can be appreciated, the turn angle, climb angle and curvature profiles of the path in Case 1 are more pronounced than those in Case 2. These profiles are related to the smoothness of the path and therefore can be adjusted by parameters p_1 and p_2.

From the set of experiments carried out in this section, it can be concluded that the smoothness and feasibility of the path are conditioned to parameters p_1 and

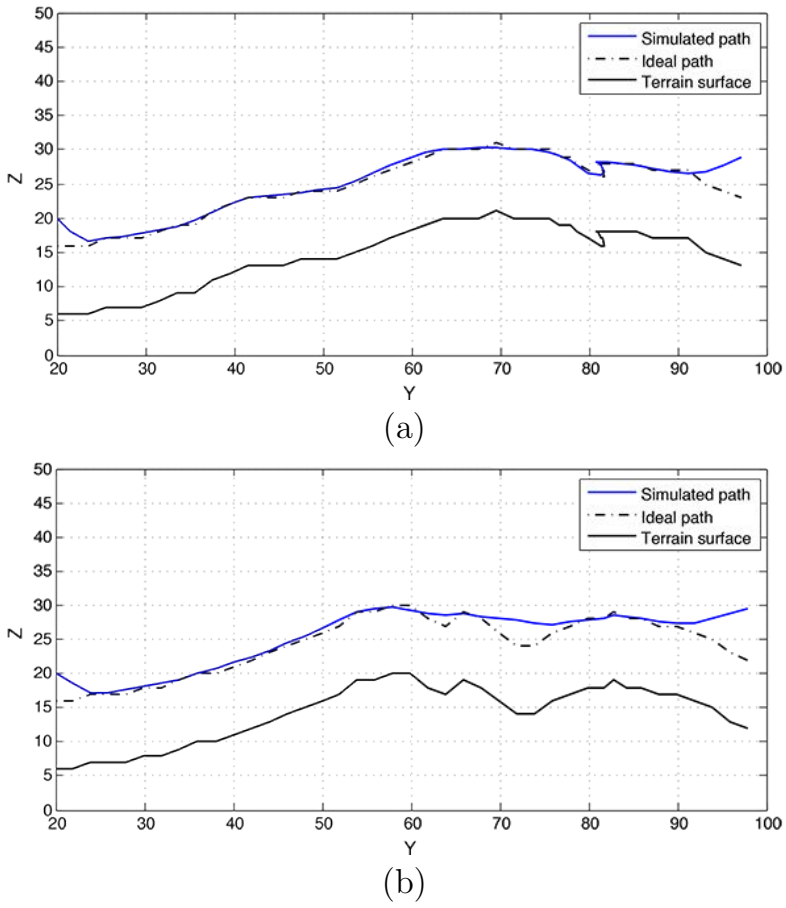


Figure 4.7: *Comparison of the resulting path against the ideal path and the terrain profile: (a)* $p_1 = 0.5$ *and* $p_2 = 1$*; (b)* $p_1 = 0.5$ *and* $p_2 = 0.8$*.*

p_2. Apart from their absolute values, the relationship between them (difference of their values) has an important role. If that difference is small, the trajectory will be more optimal in terms of smoothness and safety. On the contrary, if the difference is bigger, the generated trajectory will have sudden changes of altitude and sharper curves, but will faithfully follow the profile of the terrain.

This can be used as a first design criterion when setting the initial values of these parameters in order to adapt the smoothness of the path and make it more feasible for its execution by the UAV from the very beginning. In case the kinematic constraints of turn angle, climb rate or curvature are not fulfilled at a first run, or on the contrary, allow the obtaining of a more refined path over the terrain, the parameters can be changed according to these design rules in order to meet the requirements.

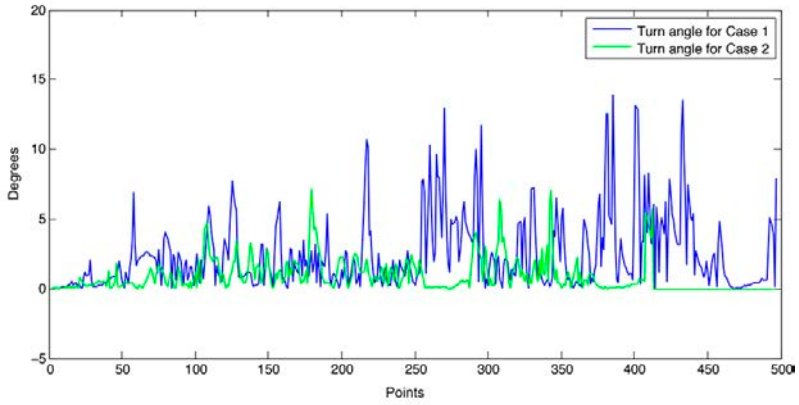


Figure 4.8: *Profile of the turn angle for different smoothness of the trajectory.*

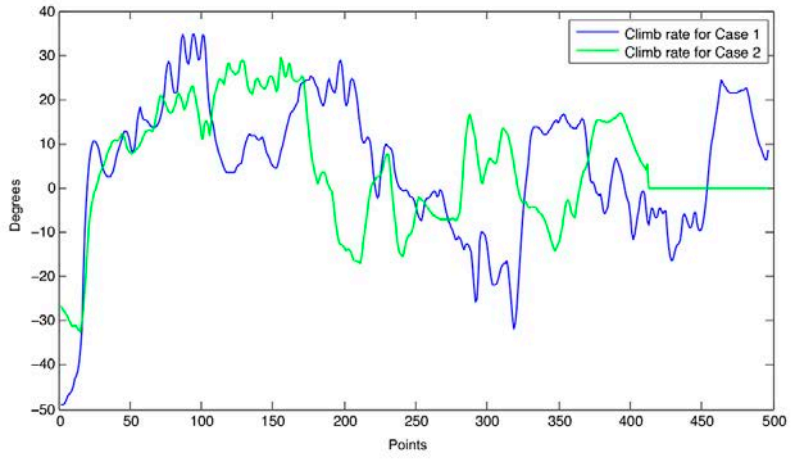


Figure 4.9: *Profile of the climb angle for different smoothness of the trajectory.*

It is clear that the optimality problem in this case depends very much on restrictions for the algorithm coming from optimization constraints such as the kinematics of the UAV or its dynamics. And even more, other constraints such as fuel consumption could be considered and will affect the optimal p_1 - p_2 sets of values. We are somehow dealing with a Pareto optimality problem that needs an accurate definition of the optimization criteria.

A further research step is currently focusing on the study of the explicit relationship between the UAV kinematics and the value of the adjustment parameters, as a first approach to the optimality study.

With our approach so far, it is expected that a feasible path can be obtained at a first run, as demonstrated in Chapter 3, where the paths resulting from FM^2 are compared with those from considering the Dubins kinematic model of a UAV. In any case, the turn angle, climb angle and curvature curves in Fig. 4.8, Fig. 4.9

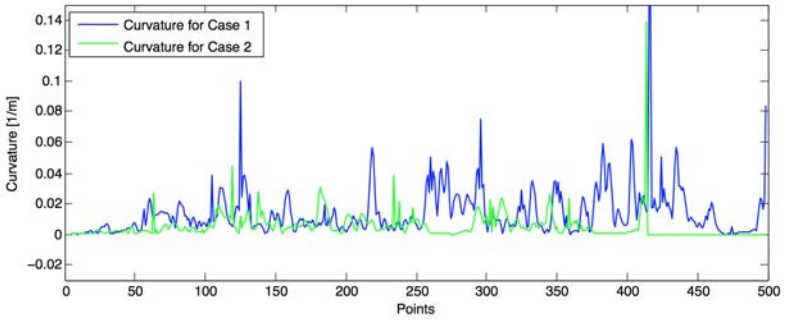


Figure 4.10: *Profile of the curvature for different smoothness of the trajectory.*

and Fig. 4.10, respectively, can be obtained and used to check the feasibility of the paths for every specific 3D environment and particular kinematic restrictions.

Besides, from the profiles in Fig. 4.8, Fig. 4.9 and Fig. 4.10, and after several mathematical computations, the dynamic restrictions for the UAV with respect to speed and acceleration ranges can be determined. This dynamic study will be done in a future work in order to reach an analytical and explicit relation between parameters p_1 and p_2 and the permissible dynamic ranges of the UAV.

In order to simplify the study on the relationship between the kinematic and dynamic restrictions and the values of the adjustment parameters, these two can be established by fixing $p_1 = 1$ and varying p_2, or setting $p_1 = k$ and $p_2 = 1 - k$ and then setting k to different values to create different paths. The introduction of parameter k to rule that relationship can help to simplify future studies.

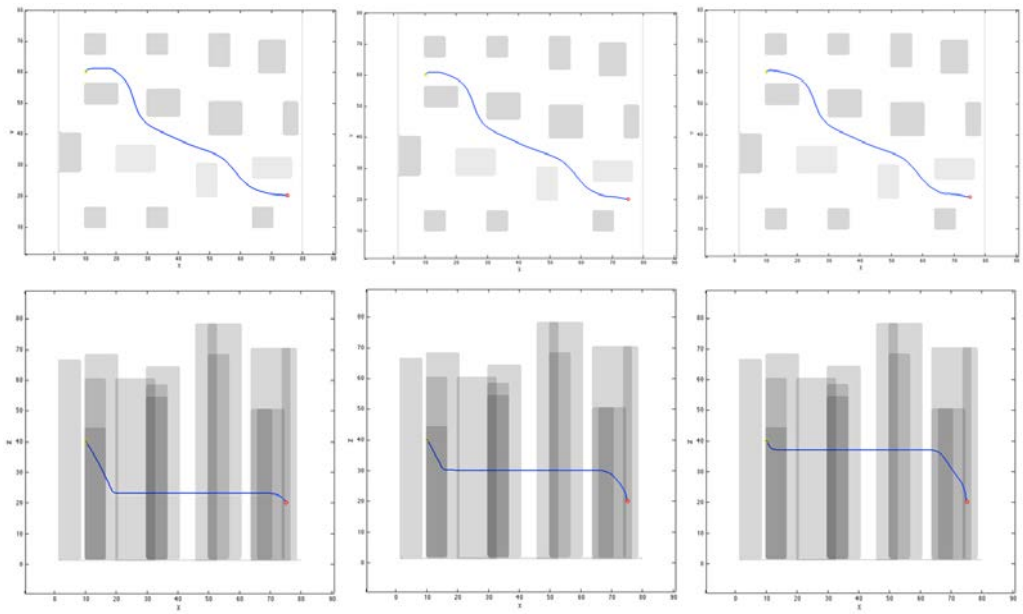
4.4 Simulation Results: Urban Environment

In this section the simulation results of the proposed method when applied to the urban environment defined in Section 4.1.1 are presented and discussed.

Our approach takes the model of the environment, generates the path using the FM^2 method avoiding the buildings of the environment and maintains an imposed flight level. The flight level and the coordinates of the start and goal points of the trajectory are given to the method.

Two different experiments are analyzed in this section. In the first experiment, the approach to fix a certain level in the planning is analyzed. In the second experiment, the approach to modify the path in terms of smoothness and safety is studied.

In both experiments, each layer of the map W is raised to different adjustment parameters. This directly affects the planning, since the wave expansion tends to expand throughout the clearer zones.



(a) (b) (c)

Figure 4.11: *Paths with different flight levels: (a) path with flight level 23; (b) path with flight level 30; (c) path with flight level 37.*

4.4.1 Control over Flight Levels

The main purpose of this experiment is to plan the optimal path between the buildings, maintaining a certain flight level.

In this first case, the start point is situated in coordinates (60, 10, 40) and the goal point in (20, 75, 20).

The layer chosen as the flight level is clarified raising it to $p_1 = 0.2$ and the rest of layers are raised to $p_2 = 0.8$. Figures 4.11(a), 4.11(b) and 4.11(c) show different trajectories, each for a particular flight level.

Besides, it is appreciated that the trajectory in the plane $x - y$ barely suffers changes. It maintains the same smooth and distance margin from the buildings in the three cases.

However, none of these trajectories are suitable to be flown by a fixed wing UAV, since the path has rapid changes of altitude, causing abrupt ascents and descents. This problem is solved in the next section.

4.4.2 Control over Path Smoothness

In the second experiment, the UAV will fly using the same goal and start points of the previous experiment at an altitude of 30. In this case, the procedure followed

consists on varying the difference between the adjustment parameters p_1 and p_2 to obtain trajectories with different degrees of smoothness.

Figure 4.12(a) shows the result when $p_1 = 0.47$ and $p_2 = 0.53$. The resulting trajectory is too smooth and the imposed flight level is not appreciable.

However, if the values of p_1 and p_2 are modified to 0.45 and 0.55, respectively, the imposed flight level is more predominant. In this case, the trajectory is still very smooth, with ascents and descents slightly pronounced, as shown in Fig. 4.12(b).

If the values are modified again by values with a higher difference, $p_1 = 0.5$ and $p_2 = 1.5$, the trajectory changes significantly as shown in Fig. 4.12(c). The imposed flight level is maintained in the majority of the trajectory, causing sharp ascents and descents. This path could be only flown by an aircraft with vertical take-off and landing (VTOL) capability.

The trajectories in the plane $x - y$ are similar in all the cases, since p_1 keeps a similar value. Nevertheless, the trajectories in the plane $x - y$ will be even smoother if p_1 acquires a lower value.

4.5 Conclusions

This chapter has presented a novel approach based on the FM^2 method to plan a trajectory for a UAV maintaining a given flight level with respect to the ground. This planning has been carried out in both an open field environment with non-uniform terrain and a urban environment.

Two adjustment parameters p_1 and p_2 have been used to force the path planning in specified areas of the 3D map determined by a given flight level. As a result, the modification of p_1 and p_2 causes variations in the cells of the velocities map W, producing its lightening or darkening. To maintain an altitude over the ground, the gridmap cells whose value is the sum of the terrain elevation plus the flight level are clarified, while the rest of the cells are obscured. Consequently, the path planning is performed through the clearer zones, where the wave front of the FM^2 has much more facility to expand, thus maintaining the specified flight level.

The results have also shown that the modification of the path in terms of smoothness and safety is caused by the variation of the difference between the values of the adjustment parameters. Besides, the greater the smoothness, the lower the probability that the path faithfully follows the profile of the terrain.

In a further research, it would be interesting to study the explicit relationship between the adjustment parameters and the kinematic and dynamic constraints of the UAV, as a first approach to the optimality study.

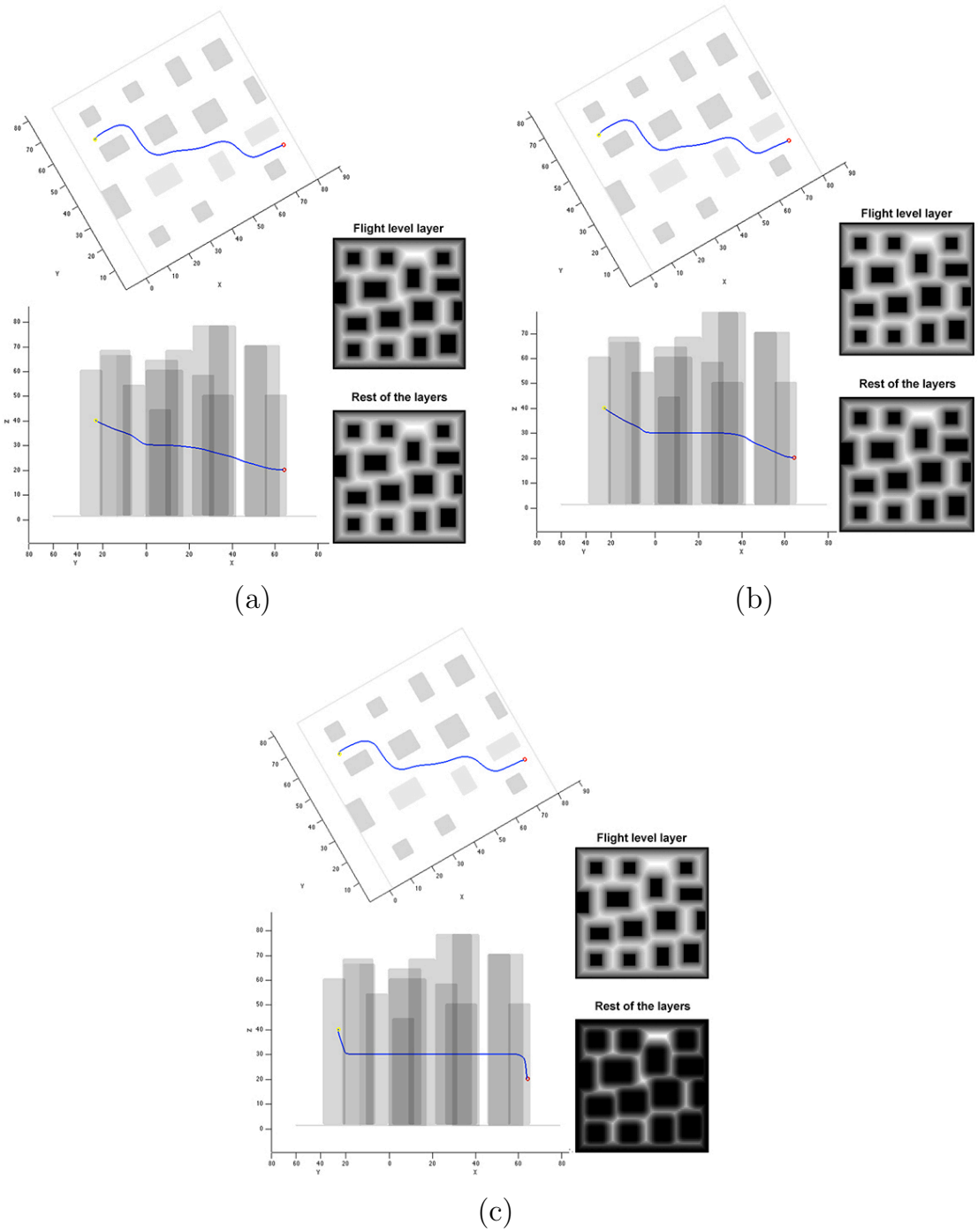


Figure 4.12: *Paths with different smoothness degrees in the plane $xy-z$: (a) $p_1 = 0.47$ and $p_2 = 0.53$; (b) $p_1 = 0.45$ and $p_2 = 0.55$; (c) $p_1 = 0.5$ and $p_2 = 1.5$.*

Chapter 5

Coverage Path Planning Approach

This chapter presents a novel approach for missions of Coverage Path Planning (CPP) carried out by UAVs in a 3D environment. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks.

The methodology followed uses an optimization process based on the Differential Evolution (DE) algorithm in combination with the FM^2 planner. The DE algorithm allows achieving the most optimal zigzag path in terms of distance travelled by the UAV to cover the whole area. Then, the FM^2 method is applied to generate the final path according to the steering angle of the zigzag bands resulting from the DE algorithm.

The approach generates a feasible path free from obstacles, keeping a fixed flight level over the ground. The flight level, smoothness and safety of the path can be modified by the two adjustment parameters previously discussed in our approach.

The simulated experiments carried out in this chapter demonstrate that the proposed approach generates the most optimal zigzag path in terms of distance, safety and smoothness to cover a certain whole area, keeping a determined flight level with successful results.

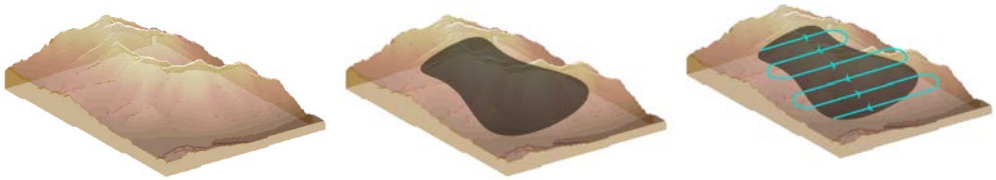
5.1 Problem Statement

The use of UAVs for area coverage applications is emerging in recent years. There are different fields where the Coverage Path Planning (CPP) is being used. For instance, a solution to perform aerial imaging applied to precision agriculture is presented in Barrientos et al. (2011); in Torresa et al. (2016) a path is found in order to completely visit an area for 3D terrain reconstruction application. The area coverage problem is also studied in Acevedo et al. (2013b) for a team of aerial mobile robots; besides, the CPP problem is used for in-detail inspection of 3D natural structures on the ocean floor in Galceran and Carreras (2013).

Nevertheless, when an area is tracked, it is important to take into account several factors with the aim of saving energy or reducing the flight time. The main factor is that the planned trajectory that covers the whole area has to be an optimal path. For this purpose, methods focused on Fuzzy control (Xixia Sun and Shen (2014)) and Genetics Algorithms (GA) are studied in different situations to evaluate the paths according to a cost function in order to achieve the most optimal result. For instance, in Trujillo et al. (2016) a GA combined with space partitioning methods is used to find the most optimal path to cover a desired area. The Differential Evolution (DE) algorithm is utilized in Zamuda and Sosa (2014) to optimize a short-term sea trajectory for an underwater glide. Another work that proposes an improved constrained DE algorithm is Zhanga and Duana (2015), where an optimal feasible route for UAVs is generated. In Nikolos et al. (2007) the DE algorithm is also used in order to find optimal paths for coordinated UAVs, where the paths are modeled with straight line segments.

Most of the techniques of CPP outlined above have been proposed for environments without obstacles. In the case of environments where obstacles are considered, an adequate path planning is required. In this way, the UAV can fly from one point to another avoiding any obstacle in the environment. Many planners have been used to resolve the path planning and obstacle avoidance problem for UAVs, such as Voronoi diagrams (Ok et al. (2013)), RRT algorithms (Lee and Shim (2014)), searching algorithms (Xue et al. (2014), Zhang et al. (2014)), nature inspired methods such as ant colony (Cekmez et al. (2016)) or algorithms based on Probabilistic Roadmap Method (PRM) (Babiarz and Jaskot (2013)), among others.

The path planning requirements depend on the specific task. The majority of the CPP applications require to capture videos or a large number of overlapping images with a high quality and with the same resolution. This implies maintaining a fixed flight level with respect to the ground. This problem has been denoted as Terrain Following Flight (TFF) and has been studied in several works over the last decade. For instance, Amirreza et al. (2014) presents path planning and optimization methods in TFF, though the problem is treated only in 2D. A 3D terrain following/avoidance trajectory optimization is performed in Kamyar and Taheri (2014), where the problem is solved generating a path and later, optimizing the resulting



(a) (b) (c)

Figure 5.1: *3D Environment: (a) the 3D simulated representation of the environment; (b) the representation of the area to be covered by the UAV; (c) the zigzag path to cover the surface.*

path. In our work (González et al. (2017b)), the path is planned at a constant height among the buildings of a city or above some of the buildings, according to the specific requirements of the mission. However, there are few works which join the CPP, the obstacle avoidance and the TFF problems.

The main objective of this work is to plan a feasible path for a UAV to cover a whole area, keeping a flight level with respect to the ground (TFF). In our approach, the CPP problem is solved using a coverage algorithm. The resulting paths from this algorithm are evaluated by the DE algorithm, choosing the most optimal one in terms of distance cost. Once the most optimal path is obtained, the FM^2 algorithm is used as the planner.

Our approach has the following inputs: an environment map where the mission is carried out and a surface which represents the area for the CPP. The environment is represented by an open field map with mountainous terrain, where the surface is rather uneven, as shown in Fig. 5.1(a). This map is a 3D grid map whose size is 118 x 87 x 40 cells. The size of each cell is equivalent to 10 x 10 x 10 meters. Here, it is not necessary to consider the size of the UAV, since it is assumed to be smaller than the size of a cell.

An instance of the surface is shown in Fig. 5.1(b), which represents the area to be covered in the mission by a zigzag path. The zigzag path for that surface is represented in Fig. 5.1(c), which is planned with a certain width between the bands and with a determined steering angle. Here, the steering angle is defined as the direction of each band with respect to the x and y axis, and the width between the bands corresponds with the camera visual field of the UAV.

The execution of the path by the UAV can be affected by its kinematics and dynamics. However, large environments are considered in general inspection cases and it is assumed that the turns of the path are compatible with the turning radius of the UAV and do not compromise its dynamic stability.

The lenght of the trajectory resulting from the coverage algorithm depends mainly on the steering angle. The DE algorithm will evaluate iteratively a cost

function defined by the distance of the zigzag path, obtaining as a result the steering angle of the path with minimum distance cost. Then, the FM^2 method is used to plan the final path according to the resulting zigzag path, avoiding any obstacles in the environment and maintaining a certain altitude with respect to the ground. The final path is composed of a set of consecutive points ($Q = P_0, P_1, ..., P_n$) and its smoothness and safety distance can be modified by the two adjustment parameters used in the FM^2 method. In this way, the path could be adapted to the kinematics of the UAV.

5.2 Methods for Coverage Path Planning

Our proposal for CPP is divided into three different methods: the Zigzag Path method, the DE algorithm and the FM^2 method.

5.2.1 Zigzag Path Method

The area to be covered is a surface, which can be represented by an irregular form (for instance, see Fig. 5.2(a)). To begin the zigzag path planning in the area, the tangent point to the surface is determined according to a certain angle α. This angle corresponds with the steering angle of the bands. The tangent point is found within the enclosed area shown in Fig. 5.2(a), and the angle α is delimited between [1, 89] degrees.

A parallel band (b_1) to the tangent line is calculated with a distance of $\frac{D}{2}$ (see Fig. 5.2(b)), where D is the distance corresponding to the visual field of the UAV camera. Here, b_1 cuts the surface in two points denoted as cut points Pc_1 and Pc_2. In our case, the direction of the zigzag path will always be from the greatest values of x to the smallest values. For this reason, Pc_1 will be the initial point of the trajectory and Pc_2 will be the point that connects with the arc towards the second band.

The next band (b_2) is calculated with a distance D from b_1, where the cut points Pc_3 and Pc_4 are obtained (see Fig. 5.2(c)). The cut points Pc_2 and Pc_3 will be connected through an arc, so both points must be on a perpendicular line to any band. For this reason, two perpendicular lines are calculated for these points to check which one has to be modified (see Fig. 5.2(d)). The point outside the surface is replaced by the new one, in this case Pc_2.

To ensure that all area is covered in the opposite side of the arc according to the vision field, a parallel line to b_1 is calculated with a distance of $\frac{D}{2}$, bv_1 (see Fig. 5.2(e)). Then, a perpendicular line is calculated above the cut point of bv_1 with the surface. If the points where this perpendicular line cuts with b_1 and b_2 cover a larger area, these ones will be considered. Here, Pc_1 will be modified.

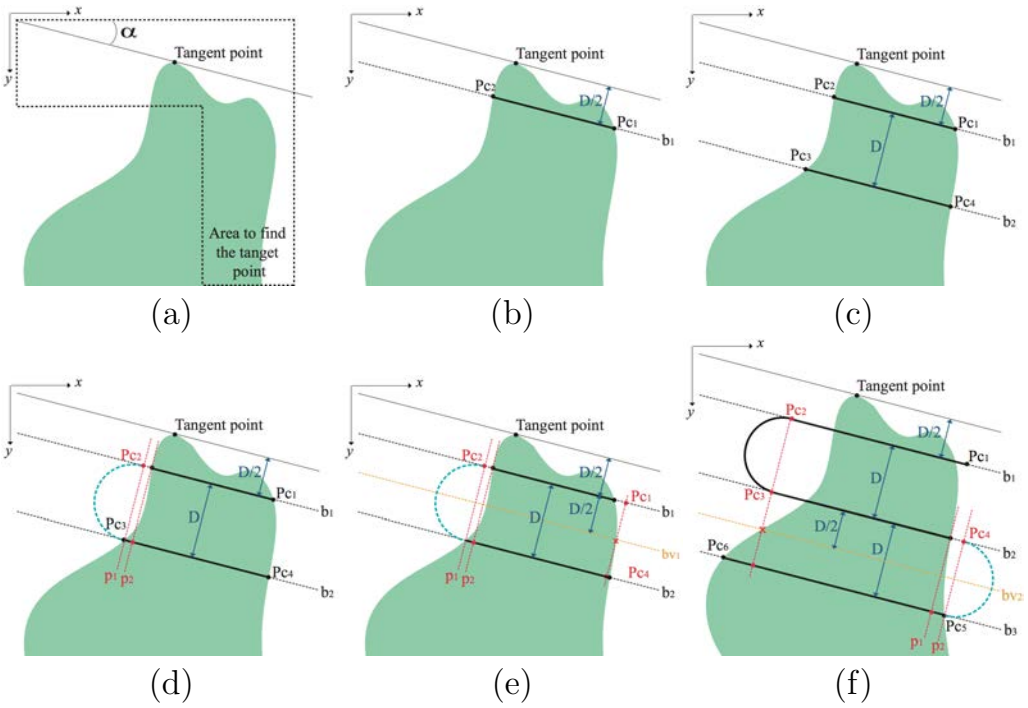


Figure 5.2: *Process to cover all the surface with a zigzag path: (a) area to find the tangent point; (b) the first band is calculated; (c) the next band is calculated; (d) creation of the cut points for the arc; (e) adaptation of the cut points for the UAV visual field; (f) repetition of the process for the rest of the bands.*

All this process is repeated for the next bands, as shown in Fig. 5.2(f). In this figure it can be observed that Pc_3 is modified because of the perpendicular line above the cut point of bv_2, which entails a modification of Pc_2.

To finish with the process, two cases can be given. On the one hand, a last band outside the surface, b_{n+1}, may be necessary, where n is the total number of bands, to cover the whole area of the surface. In this case, a parallel band to b_n is calculated outside the surface with a distance D. According to Fig. 5.3(a), a perpendicular line is calculated above the last cut point (Pc_m). The cut point of this perpendicular line with b_{n+1} will be the initial point of this band (Pc_{m+1}). Then, a parallel line bv_n is calculated with a distance $\frac{D}{2}$ from b_n. A new perpendicular line is calculated above the point where bv_n cuts with the surface (this point depends on the flight direction). The point when this perpendicular line cuts with b_{n+1} will be the final point of the path (Pc_{m+2}).

On the other hand, bv_n could be outside the surface. Here, a perpendicular line is calculated above the cut point of bv_{n-1} with the surface (on the opposite side of the arc). The cut point of this perpendicular line with b_n will be the final point of

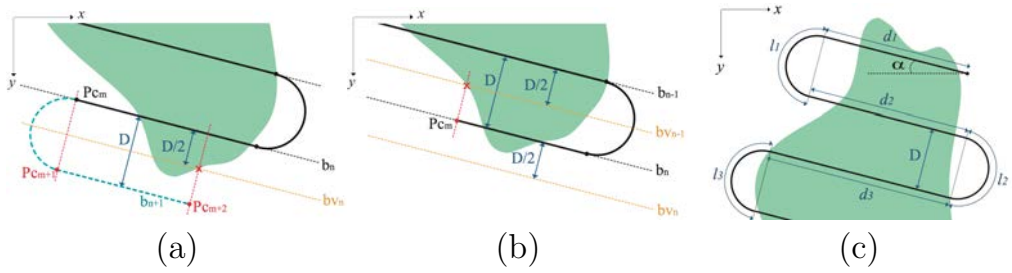


Figure 5.3: *Two final cases to cover all the surface: (a) a new band is necessary to cover the whole area; (b) no further bands are needed to cover the whole area; (c) illustration of the final zigzag bands over the surface.*

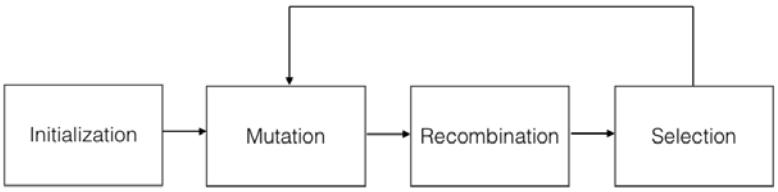


Figure 5.4: *Flow of the DE algorithm.*

the path.

5.2.2 Differential Evolution Algorithm

Once the zigzag path is generated, the DE algorithm evaluates iteratively a cost function defined by the distance of the path, according to its steering angle. In this way, the most optimal zigzag path in terms of distance cost is obtained, and therefore the best steering angle.

The DE algorithm was proposed in 1996 by Storn and Price (Storn and Price (1997)) and it is a relatively new population-based stochastic global optimization algorithm. It is based on a genetic algorithm and combines concepts of *mutation*, *crossover* and *selection* (see Fig. 5.4), which are described in the literature in Goldberg (1989). In order to optimize a certain function, this method uses n D-dimensional parameter vectors, $x_i^G = \{x_{i,1}^G, x_{i,2}^G, ..., x_{i,d}^G\}$; i = 1, 2, ..., n, where d is the number of real parameters of the function and G the generation number. It is noted that n is a fixed parameter during the minimization process.

The initial population is chosen randomly, defining initial limits $x_j^{min} \leq x_{j,i,1} \leq x_j^{max}$ and must cover the entire parameter space. A uniform probability distribution is assumed for all random decisions. DE generates new parameter vectors through the addition of the weighted difference vector between two population vectors to a third vector. This process is denoted as *mutation* and is illustrated in Fig. 5.5. The

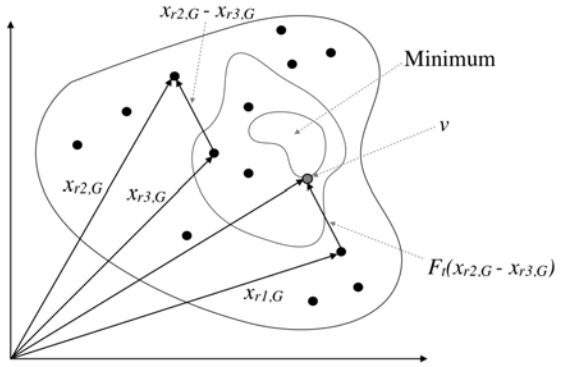


Figure 5.5: *Generation of the new population vector.*

resulting vector is called *trial* vector and is defined by

$$v_i^{G+1} = x_{r1}^G + Ft(x_{r2}^G - x_{r3}^G), \tag{5.1}$$

where $Ft \in [0, 2]$ is a factor which controls the amplification of the differential variations $x_{r2}^G - x_{r3}^G$, which is determined by the user, and $r1$, $r2$, $r3 \in \{1, 2, ..., n\}$, being $r1 \neq r2 \neq r3$. The vector v_i^{G+1} is called *donor* vector.

A *crossover* is introduced with the aim of increasing the diversity of the new generation of parameter vectors. The new parameter vector $u_i^{G+1} = (u_{i,1}^{G+1}, u_{i,2}^{G+1}, ..., u_{i,d}^{G+1})$ is defined according to

$$u_i^{G+1} = \begin{cases} v_{i,j}^{G+1} \text{ if } \text{rand}_{i,j} \leq CR \text{ or } j = I_{rand}, \\ x_{i,j}^G \text{ if } \text{rand}_{i,j} > CR \text{ and } j \neq I_{rand}, \end{cases} \tag{5.2}$$

where $i = 1, 2, ..., n$ and $j = 1, 2, ..., d$. The term $\text{rand}_{i,j}$ is the jth evaluation of a uniform random number generator with outcome $\in [0, 1]$, CR is the crossover probability $\in [0, 1]$, which is determined by the user, and I_{rand} is a randomly chosen index $\in 1, 2, ..., d$, which ensures that $u_i^{G+1} \neq x_i^G$.

As shown in Eq. (5.3), the comparison between the trial vector u_i^{G+1} and the population element x_i^G is indispensable for the *selection* process. In this case the greedy criterion is used: if the element u_i^{G+1} has a cost value lower than x_i^G, the value of x_i^{G+1} will be replaced by u_i^{G+1}; on the contrary, x_i^G keeps its initial value.

$$x_i^{G+1} = \begin{cases} u_i^{G+1} \text{ if } f(u_i^{G+1}) \leq f(x_i^G) \\ x_i^G \text{ otherwise} \end{cases} \quad i = 1, 2, ..., n. \tag{5.3}$$

The process of *mutation*, *crossover* and *selection* is repeated depending on the criteria established by the user, such as the maximum number of iterations and the population number.

5.2.3 Fast Marching Square Method

When the best zigzag path in terms of distance cost is obtained, the FM^2 algorithm described in Chapter 4 is used to plan the final path at a fixed altitude with respect to the terrain, avoiding any obstacles found in the environment.

5.3 Coverage Path Planning Approach

This section presents our approach, whose main objective is to generate a feasible zigzag path with the minimum distance cost in order to cover a certain area of an environment. The path is planned according to the visual field of the UAV, maintaining a fixed flight level with respect to the ground. The area to be covered, the flight level and the distance between the bands are chosen by the user. The environment where the CPP is carried out is the one described in Section 5.1.

First, the Zigzag Path method is used to generate the zigzag path that will cover all the surface, and then the DE algorithm is used to determine what the best steering angle of the zigzag bands (α) is, providing the most optimal zigzag path in terms of distance cost. Thanks to the evolutionary optimization technique of this algorithm, a minimization of the cost function is possible.

The bands of the zigzag are disposed to cover the whole area according to a certain distance (D) between each band, as shown in Fig. 5.3(c). This distance is equivalent to the vision field of the UAV. Each strech of the bands is composed of the longitude of a straight line (d_i) and the longitude of an arc (l_i). The cost of the path is determined by the sum of each of the longitudes. So, the cost function of this problem to be minimizated is described as follows:

$$Ct(\alpha) = \sum_{i=1}^{n} (d_i + l_i), \tag{5.4}$$

for $i = 1, 2, ..., n$, n being the maximum number of stretches in the planning. Next, it is explained how the DE algorithm is used in our approach, whose procedure is described in Fig. 5.6.

- *Initialization*: The population is formed by a specific number of individuals $N_p = 20$, which represents each possible solution. For the initial generation $g = 0$, the initial population is calculated using a random number distribution according to

  $$x_{i,j}^{0} = \alpha_{max} \cdot \tau, \tag{5.5}$$

  where $\alpha_{max} = 89^{o}$ is the maximum permissible angle and τ is a random value between [0, 1].

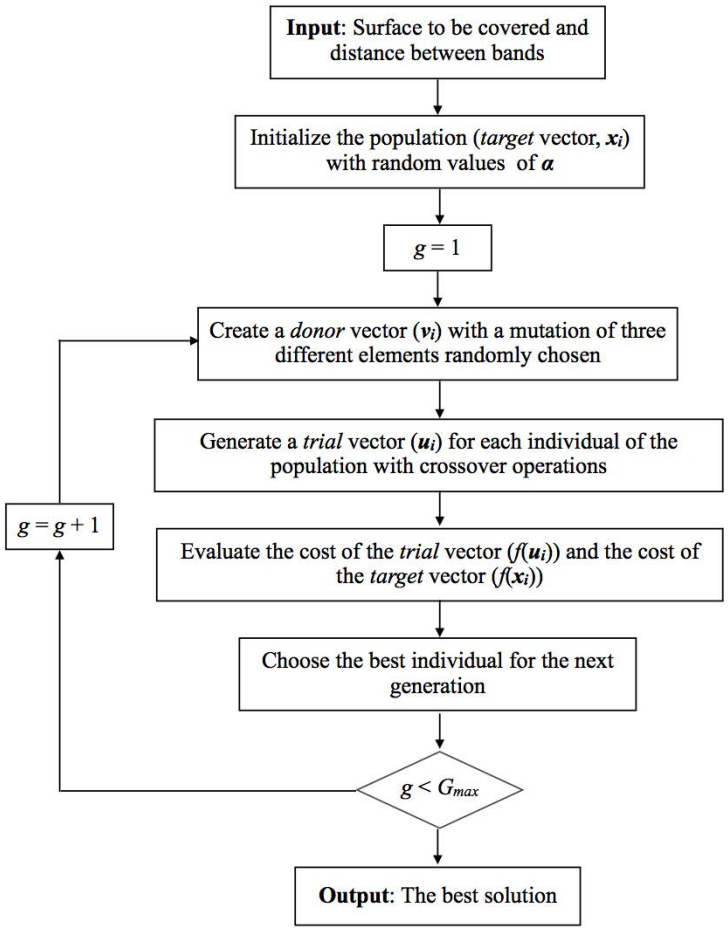


Figure 5.6: *Flowchart of the DE algorithm for our approach.*

- *Mutation*: The mutation process generates the *donor* vector from the linear combination of different members of the population according to

$$v_i^g = x_{r_1}^g + F \cdot (x_{r_2}^g - x_{r_3}^g), \tag{5.6}$$

where the parameter related to the speed of convergence is a positive constant $F = 0.8$, whose value gives a good precision of the solution, and values r_1, r_2 and r_3 are random individuals, being $r_1 \neq r_2 \neq r_3 \neq i$. In this way, the mutation is based on the randomness.

- *Crossover*: The crossover process is used to increase the diversity of our generation, where the *trial* vector u_i^g is calculated. This vector is generated from the elements of the *target* and *donor* vectors according to

$$\{u_{i,j}^g\} = \begin{cases} v_{i,j}^g \ if \ Cr_{rand} < 100 \cdot CR \\ x_{i,j}^g \ otherwise, \end{cases} \tag{5.7}$$

where Cr_{rand} is a random value generated for each j between [0, 100] and $CR = 0.7$, which means a greater probability of selecting the elements of the *donor vector*. If the value of Cr_{rand} is greater than 89 or lower than 1, u_i^g is calculated again as $u_{i,j}^g = \alpha_{max} \cdot \tau$.

- *Selection*: The selection mechanism compares the individuals of the cost of the *trial* vector and the cost of the *target* vector. Here, the best result is chosen. If the value of the *trial* vector cost is smaller than the value of the *target* vector cost, the result is the value of u^g. On the contrary, the best result will be the value of the individual of the population x^g, maintaining in both cases this value for the next population.

$$\{x^{g+1}\} = \begin{cases} u^g \ if \ f(u^g) < f(x^g) \\ x^g \ if \ f(u^g) \geq f(x^g). \end{cases} \tag{5.8}$$

Fig. 5.7(a) shows the specific surface to track in the map and Fig. 5.7(b) shows the resulting zigzag path from the DE algorithm to cover the whole area of the surface with a minimum cost.

Once the zigzag path have been obtained, the FM^2 method is used to generate the feasible path, thus avoiding any obstacle in the environment (with security margins) and maintaining a fixed fligh level with respect to the ground.

The map W_0 is modified according to the resulting zigzag path, as shown in Fig. 5.7(c), where the path will be planned through the zone drawn by the bands. Each strech of the bands is enlarged according to the distance between the bands in order to allow the path to avoid obstacles in the plane $x - y$, and not only in the plane z. Fig. 5.7(d) shows how the FM^2 is applied as described in Chapter 4, presenting the final resulting path in Fig. 5.7(e) and its 3D perspective in Fig. 5.7(f).

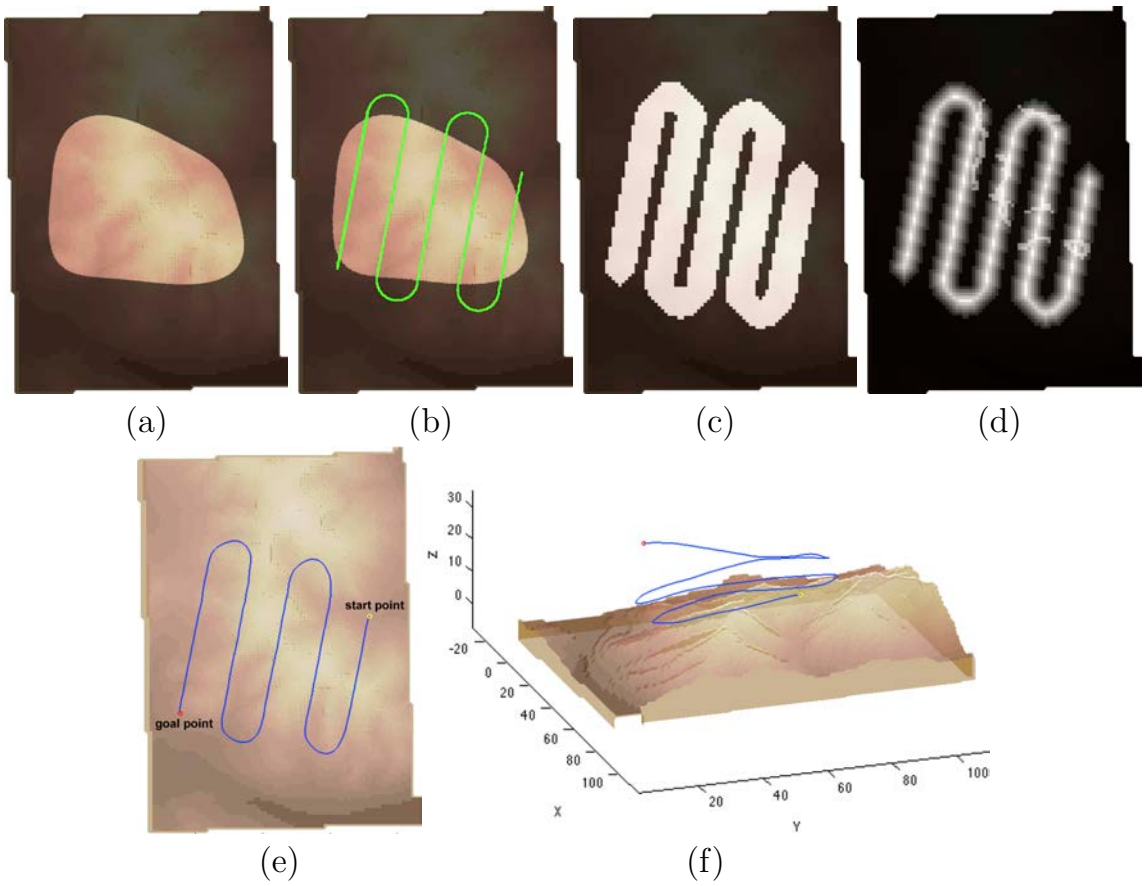


Figure 5.7: *Process to obtain the path with minimum cost: (a) surface to be covered; (b) result from the DE algorithm; (c) modification of W; (d) applying FM²; (e) final resulting path; (f) 3D perspective of the resulting path from our approach.*

From Fig. 5.7(f), it is appreciated that a flight level with respect to the ground is maintained. The influence of the values of parameters p_1 and p_2 on the smoothness and feasibility of the path has been discussed in Chapter 4.

5.4 Results and Discussion

Having defined the environment, the approach based on the DE and FM^2 algorithms takes a certain surface and plans a feasible zigzag trajectory to cover the whole area with the lowest possible cost in terms of distance at a fixed flight level with respect to the ground, avoiding any static obstacle in the environment.

Here, two different experiments have been carried out. In the first experiment, different paths are calculated for four different irregular surfaces. In the second

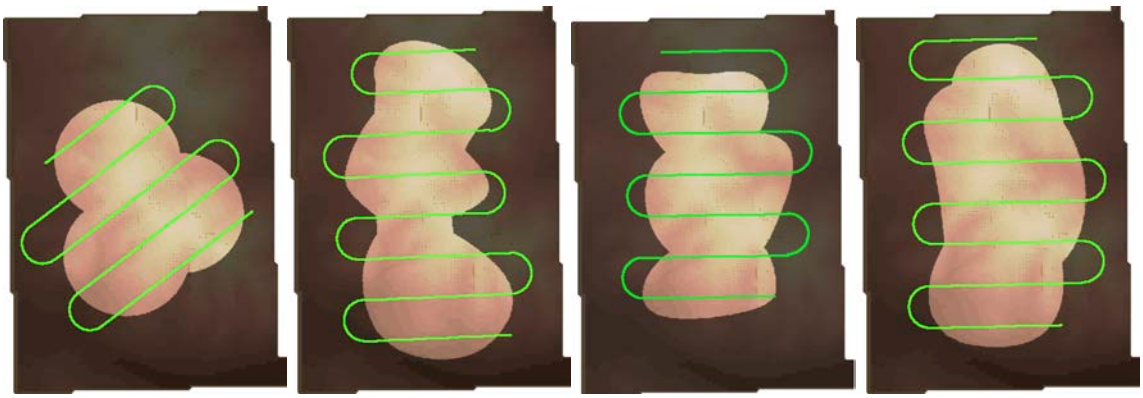


Figure 5.8: *Zigzag bands resulting from the DE algorithm for different surfaces: (a) zigzag bands for S1; (b) zigzag bands for S2; (c) zigzag bands for S3; (d) zigzag bands for S4.*

Table 5.1: *Costs and steering angles of the bands for different surfaces.*

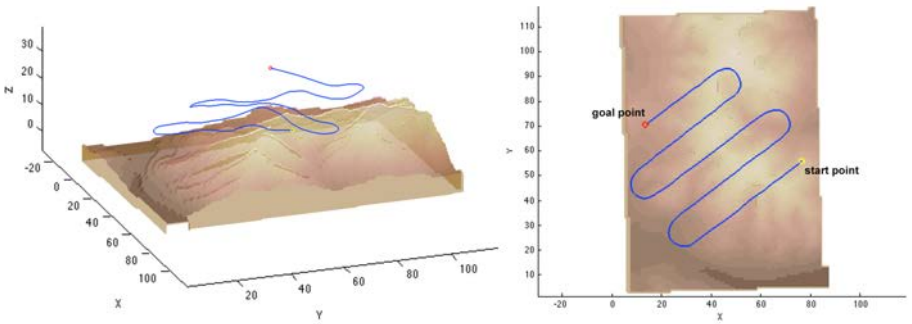
	Minimum		Maximum		Median		Average	
Surface	Cost (m)	α (°)	Cost (m)	α (°)	Cost (m)	α (°)	Cost (m)	α (°)
S1	5446.504	37.134	6104.040	11.499	5464.316	37.684	5616.718	37.214
S2	6748.057	2.503	8373.462	71.434	6768.528	3.389	6870.913	8.597
S3	5908.607	1.012	7493.887	65.506	5980.716	2.086	6061.788	9.452
S4	7176.987	1.349	8223.283	44.610	7215.164	2.581	7290.188	7.953

experiment, the distance between the bands that constitute the zigzag is modified, generating different zizag paths. Both experiments show how the smoothness of the trajectory can affect the compliance of the flight level constraint, since the terrain is highly non-uniform.

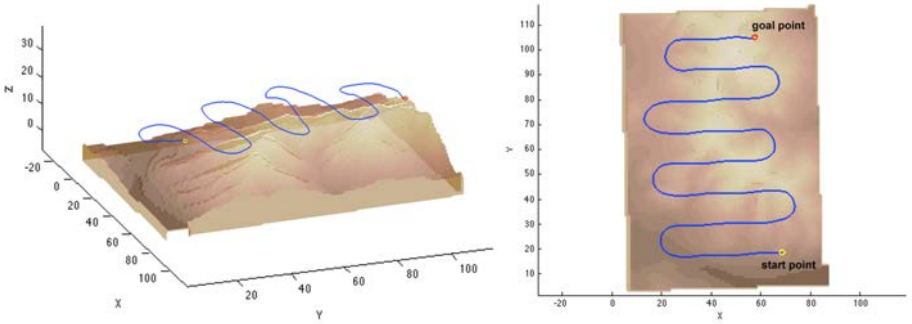
For both experiments, the DE algoritm runs over 30 iterations with a population of 20 individuals and the fixed flight level is 10, with values of p_1 and p_2 of 0.5 and 1.3, respectively.

5.4.1 Different Areas to be Covered

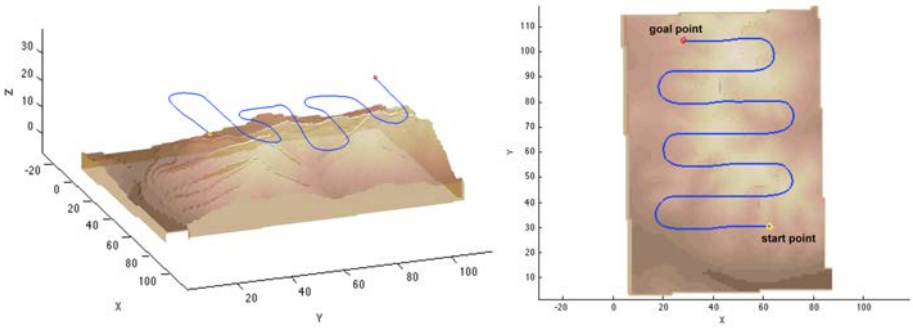
The aim of this experiment is to plan the optimal trajectory in terms of distance cost to cover a whole area. The path has to be feasible by an UAV avoiding any obstacle in the environment and keeping a fixed altitude with respect to the ground. Four different cases are presented in this experiment, evaluating in each one a different area. The different areas to be covered are presented in Fig. 5.8, which have irregular forms. The chosen distance between the bands is 80 cells, because it is assumed that the visual field of the UAV is 800 meters. The size of each cell of the map is equavalent to 10 x 10 x 10 meters.



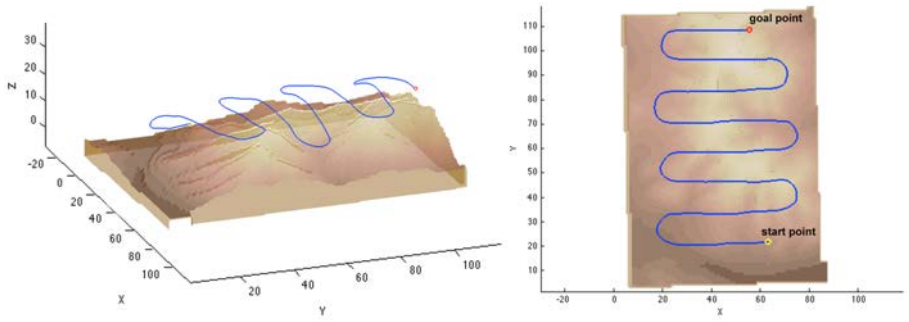
(a)



(b)

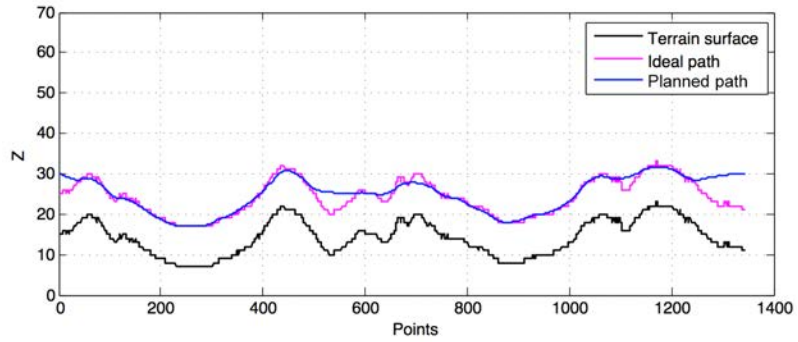


(c)

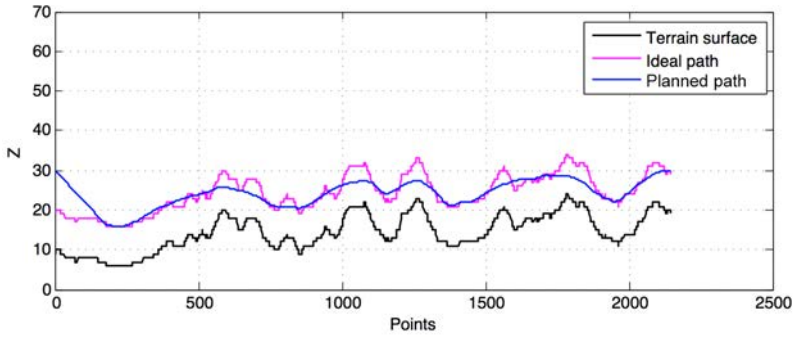


(d)

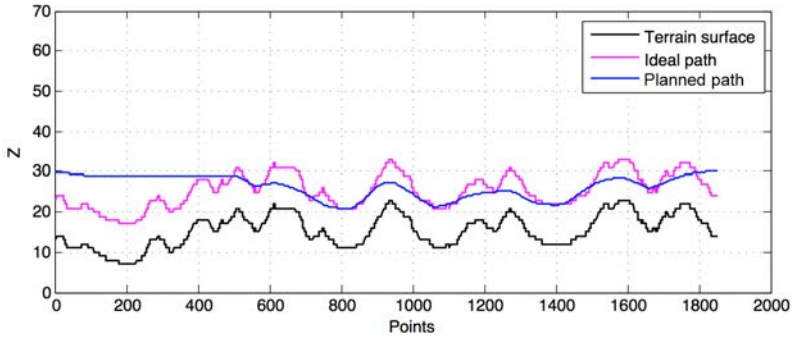
Figure 5.9: *3D perspective (left) and $x-y$ view (right) of the resulting path: (a) path for S1; (b) path for S2; (c) path for S3; (d) path for S4.*



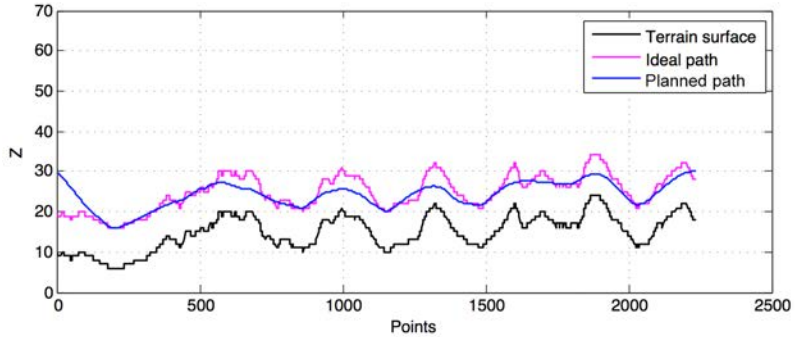
(a)



(b)



(c)



(d)

Figure 5.10: *Comparison of the resulting path against the ideal path and the terrain profile: (a) path for S1; (b) path for S2; (c) path for S3; (d) path for S4.*

Fig. 5.8(a) shows the resulting zigzag path for the surface S1. Table 5.1 shows the results obtained from the algorithm DE, where the minimum (best), maximum (worst), average and median of the cost and angles are listed. The first line of the table shows the values for the surface S1. In this case, the best angle to obtain the most optimal path is 37.134 degrees with a trajectory cost of 5446.504 meters. The worst cost is 6104.040 meters for an angle of 11.499 degrees. The median and average are maintained in values closer to minimum. The result of applying FM^2 for a path with a steering angle of 37.134 degrees is shown in Fig. 5.9(a). Here, the 3D perspective and $x - y$ view of the path are shown. In Fig. 5.10(a) it can be appreciated how the planned path maintains the flight level in the majority of the trajectory, the difference between the planned path and the ideal path being minimal.

The result for the surface S2 is shown in Fig. 5.8(b). The second row of Table 5.1 corresponds to the values of the cost and steering angles for S2. Here, the minimum cost of the trajectory is 6748.057 meters with an angle of 2.503 degrees. The worst cost is 8373.462 meters with an angle of 71.434 degrees. Fig. 5.9(b) shows the resulting zigzag path when FM^2 is applied with a steering angle of 2.503. Fig. 5.10(b) shows the altitude profile of the trajectory for the surface S2. There are several points where the UAV is not capable to fly over due to the values of p_1 and p_2, fixed for a certain kinematic of the UAV. For a major fulfilment of the fixed altitude, parameters p_1 and p_2 must be changed so that the difference between them is higher, as explained in Chapter 4.

In the case of the surface S3, the result is shown in Fig. 5.8(c). Here, it is appreciated that one last band is necessary to cover the whole area. The best angle for the path is 1.012 degrees, providing a cost of 5908.607 meters. This values can be found in the third row of Table 5.1. The worst angle is 65.506 degrees, providing a cost of 7493.887 meters. In Fig 5.10(c) the altitude profile of the path can be appreciated. As in the previous case, the UAV is unable to reach several points of the trajectory, due to its kinematic limitations, fixed by p_1 and p_2.

Finally, the case for the surface S4 is presented in Fig. 5.8(d), where it is appreciated that a final band of the zigzag is necesary to cover the whole area, as in the previous case. The zigzag bands have an angle of 1.349 degrees, whose trajectory has a cost of 7176.987 meters. The worst angle for this case is 44.610 degrees with a cost of 8223.283 meters. The maintained altitude for the path is shown in Fig. 5.10(d). As in the two cases described above, the terrain has significant irregularities, being impossible for the UAV to maintain a constant level with respect to the ground.

5.4.2 Different Widths between the Bands

This second experiment analyzes how the trajectory changes depending on the visual field of the UAV. Three different cases are studied for the surface S1, modifying the

Table 5.2: *Costs and steering angles for different distances between the bands for the surface S1.*

	Minimum		Maximum		Median		Average	
Distance (cells)	Cost (m)	α (°)	Cost (m)	α (°)	Cost (m)	α (°)	Cost (m)	α (°)
40	9679.496	38.474	10885.701	86.879	10150.668	36.968	10066.730	33.238
60	6909.084	37.878	7696.142	79.164	6940.380	36.990	6982.500	36.189
100	4615.506	38.368	5394.695	57.202	4658.287	37.670	4761.862	29.829

distance between the bands of the zigzag and checking how the steering angle varies according to the cost of the trajectory.

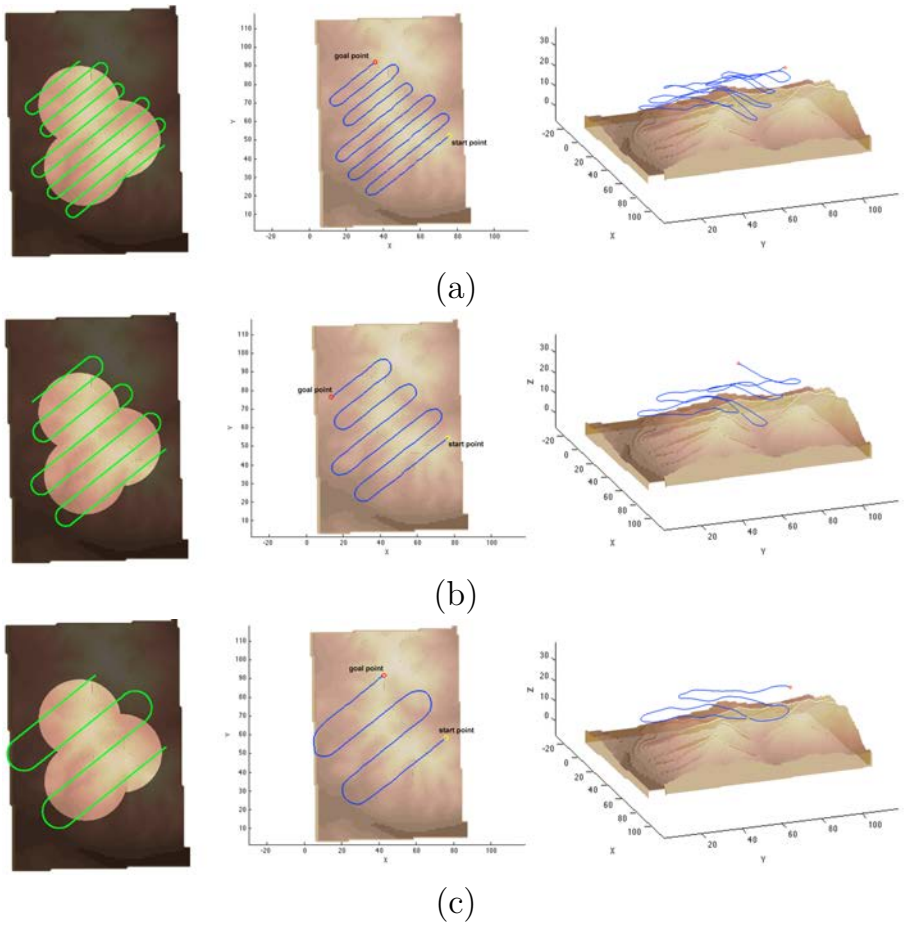


Figure 5.11: *Path planning for surface S1 with different distances between bands: (a) path for distance 40 cells; (b) path for distance 60 cells; (c) path for distance 100 cells.*

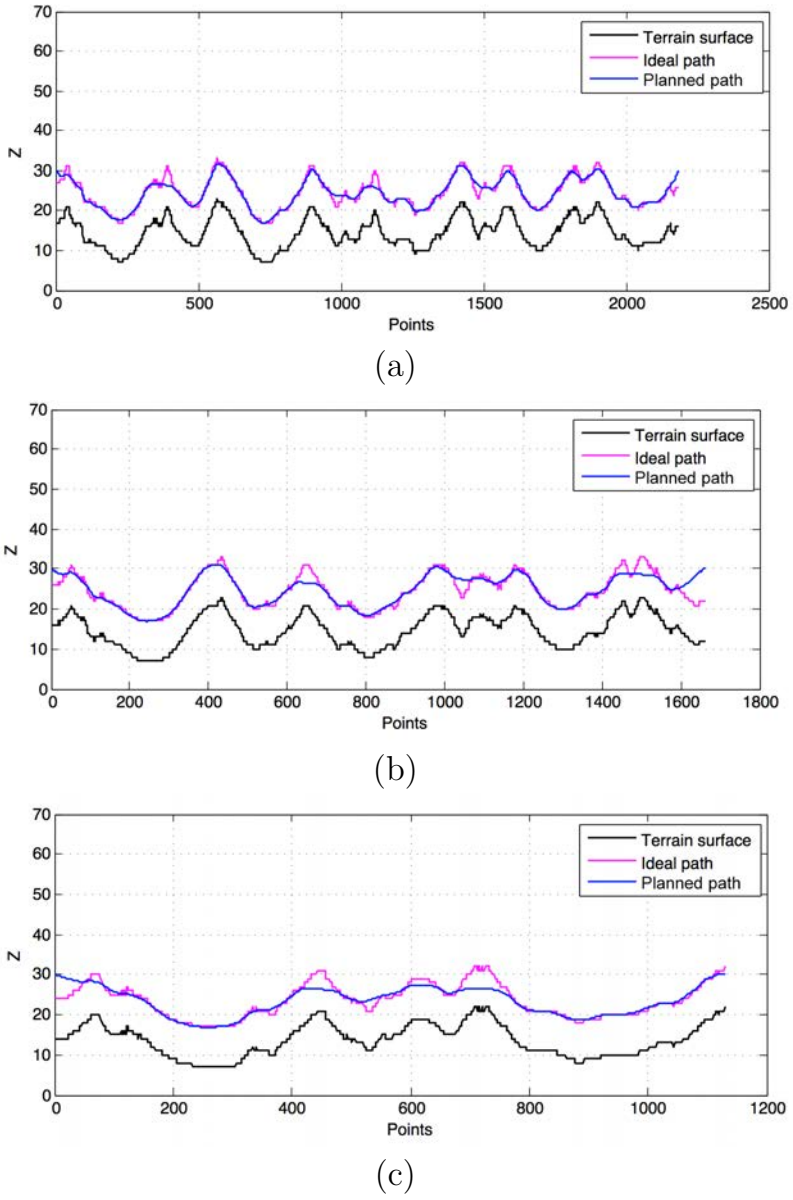


Figure 5.12: *Path planning for surface S1 with different distances between bands: (a) path for distance 40 cells; (b) path for distance 60 cells; (c) path for distance 100 cells.*

Fig. 5.11(a) shows the trajectory with minimum cost for a visual field of 40 cells. Table 5.2 shows the minimum, maximum, median and average values for the costs and steering angles of the trajectory. If the first row is observed, the best angle is 38.474 degrees with a cost of 9679.496 meters, the worst cost being for an angle of 86.879 degrees.

The result for a visual field of 60 cells is shown in Fig. 5.11(b). Here, the best

steering angle is 37.878 degrees providing a cost of 6909.084 meters, as appreciated in the second row of Table 5.2. The maximum cost for this case is 7696.142 meters, provided by an angle of 79.164 degrees.

The last case is shown in Fig. 5.11(c), with a visual field of 100 cells. Acording to the third row of Table 5.2, the minimum cost is 4615.506 meters with an angle of 38.368 degrees and the maximum cost is 5394.695 meters with an angle of 57.202 degrees.

Fig. 5.12 shows the planned path against the ideal path and the terrain profile. In these three cases, the flight level is maintained in the majority of the trajectory with minimal differences.

As a conclusion, it can be seen that, in all cases, the minimum cost is achieved with similar steering angles, independenly of the visual field of the UAV. The flight level is respected with small variations, depending on the smoothness imposed by the planning.

From the set of experiments carried out in this section, it can be concluded that the approach is valid for any surface, independently of the distance between the bands. The smoothness and feasibility of the path are conditioned to parameters p_1 and p_2. If their difference is small, the trajectory will be more optimal in terms of smoothness and safety. On the contrary, if the difference is bigger, the generated trajectory will have sudden changes of altitude and sharper curves, as demonstrated in Chapter 4, but will faithfully follow the profile of the terrain.

5.5 Conclusions

This chapter has presented a novel approach based on the DE algorithm and the FM^2 method to plan a feasible zigzag path for UAVs in order to cover a certain area of an environment. The method generates an optimal path in terms of smoothness and safety with a minimum distance cost, maintaining a fixed flight level with respect to the ground and avoiding any obstacle in the terrain. The cost of the path corresponds with its length.

First, the Zigzag Path method is used for the generation of the zigzag bands, and then the DE algorithm optimizes them so that the steering angle is optimal, ensuring a minimal distance cost. Later, the map W_0 is modified according to the obtained zigzag path and the FM^2 method is applied over this map. Two adjustment parameters p_1 and p_2 have been used to adjust the path to the kinematics of the UAV and to force the path planning in specific areas of the 3D map determined by a given flight level. The generated path is optimal not only in terms of distance cost, but also in terms of safety and smoothness, and therefore, feasible for the UAV.

Our approach has been proven in two experiments 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 the algorithm generates paths to save energy or fuel, and in addition, generates trajectories compatibles with the kinematics of the UAV.

Chapter 6

UAVs Formation Approach

This chapter introduces the path planning problem for 3D UAV formations based on our FM^2 algorithm. The approach is based on a leader-followers scheme, where the reference pose of the follower AUV is defined by the geometric equations that place the goal pose of each follower as a function of the leader pose. The solution presented here allows the UAV formation to adapt its shape so that the obstacles can be avoided and a flight level with respect to the ground can be kept.

Simulation results will be presented in different environments to show the validity and robustness of the approach.

6.1 Introduction

It is clear that, in many cases of application, the use of multi-agent systems improves the performance, flexibility and robustness of the mission thanks to the introduction of different agents in the planning (Martin et al. (2001)). Very common applications are exploration (Dewan et al. (2013)), search and rescue (Hauert et al. (2009)), and surveillance (Acevedo et al. (2013a), Likhachev et al. (2013)), among others.

The formation problem requires to address important research topics, such as modelling and control of agents (Bouabdallah (2007)), collision avoidance (Krabar (2011)), mapping and state estimation (Shen et al. (2011)), and formation control and planning (Hino (2010)). 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 one of key research topics.

When the operation is performed in limited spaces or for collaborative tasks, the movements of the agents have to be planned and coordinated efficiently. Besides, a computationally fast solution is also required so that the travel speed can be kept.

There exist several strategies that describe how to control the evolution of a formation. For instance, the multi-agent coordination problem is studied in Ogren et al. (2002) 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 (Zhang et al. (2010)). In other behavior-based approaches (Cao et al. (2003)) each agent has basic primitive actions that generate the desired behaviors in response to sensory inputs. For the case of leader-followers approach, a common solution is the model predictive controller (Ahmad et al. (2013)) which was recently introduced for holonomic robots (Kanjanawanishkul and Zell (2008)).

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 the previous chapters, our FM^2 approach shows a robust performance when it comes to these two issues. This is the reason why we have taken a step towards its application to UAVs formations.

In this chapter a leader-followers approach is presented. In this configuration, a UAV is designated as the leader of the formation and follows a trajectory towards the goal point. At the same time, follower UAVs are positioned behind the leader according to a default geometry shape that can change, within a given range, trying to accommodate to the environment conditions (Alvarez et al. (2014), Garrido et al. (2011), Yu et al. (2010)).

A detailed description of the approach is given in Section 6.3.

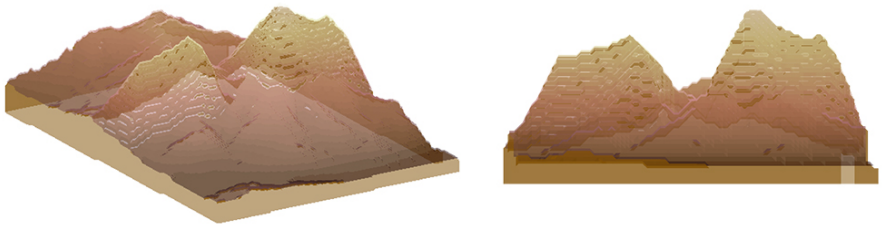


Figure 6.1: *The left part of the image shows the 3D simulated representation of the open field environment. The right part shows the front view of the environment.*

6.2 Problem Statement

In this chapter 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.

6.2.1 Environment

The 3D environment where the path planning for UAVs formation is carried out is represented in Fig. 6.1 . The left part of the figure represents an open field with mountainous terrain where the surface is rather uneven. The 3D grid map has a dimension of 120 x 90 x 40 cells, where each cell of the map is equivalent to 15 x 15 x 15 meters. As mentioned in previous experiments, 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.

6.2.2 Mission for the UAVs formation

The mission presented in this chapter requires a UAVs formation moving throughout an open field, avoiding any obstacle in the terrain.

There are several type of formations; however, this work focuses on a leader-follower 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 work is formed by three UAVs that compose a triangular shape among them. The formation will avoid any obstacle in the environment, deforming and adapting to its characteristics, and also taking into account the rest of the UAVs of the formation.

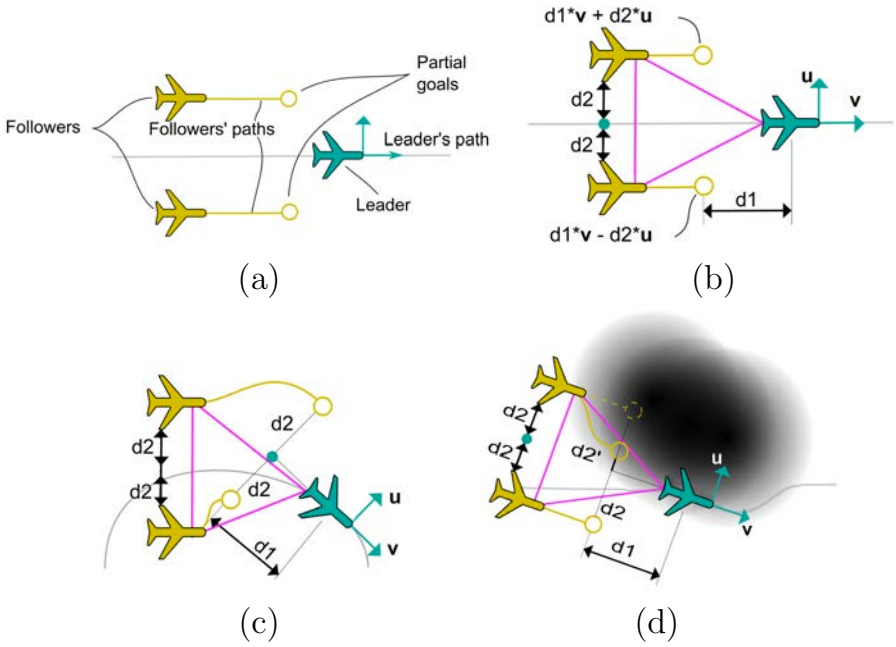


Figure 6.2: *Behavior of the UAV formation algorithm: (a) main components; (b) triangular-shaped UAV formation; (c) partial goals according to the leader position; (d) partial goals according to the obstacles of the environment.*

The approach implemented to find the trajectory for the leader and its followers is based on our FM^2 approach. The method described in Chapter 4 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 next.

6.3 UAVs Formation Approach

This section presents the approach to create a formation composed of three UAVs with triangular shape, which operates in an 3D 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 Gómez et al. (2013). The improvements included here are: 1) the adaptation of the formation to a 3D 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

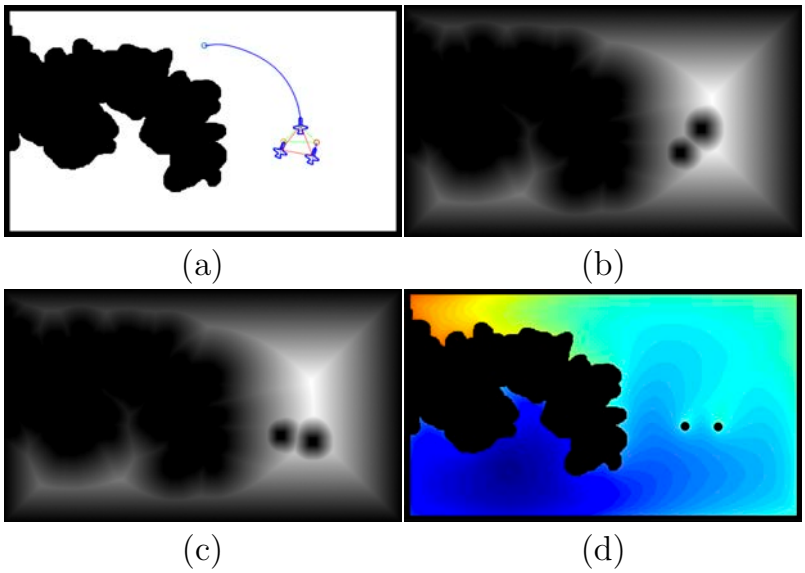


Figure 6.3: *UAVs formation approach: (a) UAV formation following a path; (b) first potential map taking the leader and one follower as obstacles; (c) first potential map taking the followers as obstacles; (d) second potential map taking the followers as obstacles.*

the position of the leader as reference. These geometric relations are shown in Fig. 6.2, 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 next section a detailed description of our approach is presented.

6.3.1 UAVs Formation Algorithm

As described in Chapter 2, the FM^2 technique uses the FM method twice to create two different potential maps. The first time, the FM method creates a potential map identified as W, and the second time it generates a wave front growing into W and given 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 obstacles in the environment. Each of the UAVs, like the rest of obstacles in the environment, must have additional *repulsive forces*, preventing them from colliding with each other. Therefore, the integration of the potential given by the FM method 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).

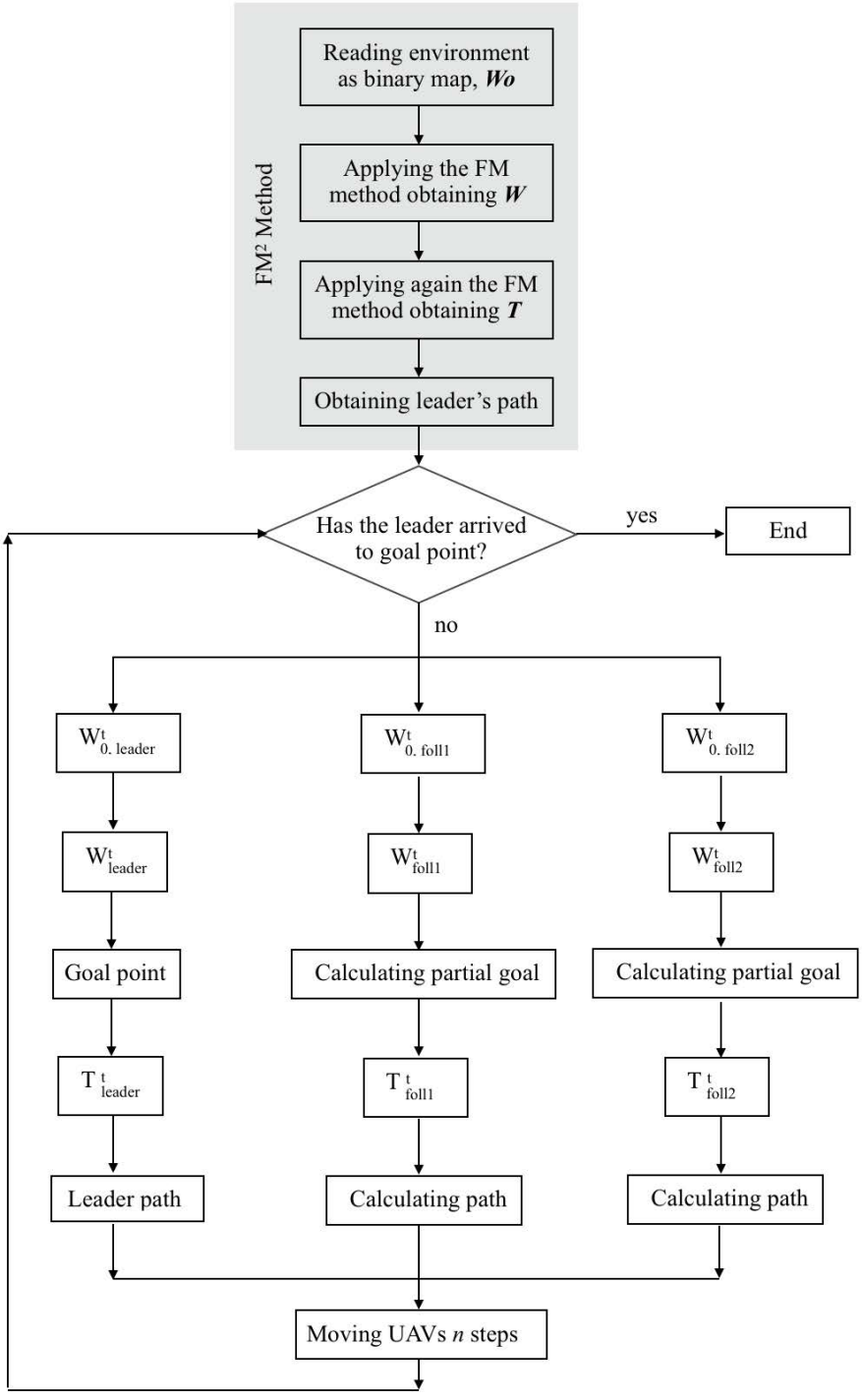


Figure 6.4: *The left part of the image shows the 3D simulated representation of the open field environment. The right part shows the front view of the environment.*

2. The FM method is applied to W_0 generating the first potential map W.

3. The FM method 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 towards 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 next:

   (a) Each UAV$_i$ of the formation is included in its binary map $W^t_{0,i}$ together with the rest of the UAVs, leaders and followers, labeled as obstacles. Fig. 6.3 (a) shows the binary map with the leader and followers.

   (b) For each UAV$_i$, a new first potential W^t_i is generated from $W^t_{0,i}$ in each cycle t. Figures 6.3 (b) and 6.3 (c) shows how the rest of UAVs of the formation are labeled as obstacles.

   (c) The *partial goals* (x_{gj}, y_{gj}, z_{gj}) for each follower$_j$ are calculated from the position of the leader. These *partial goals* indicate the next position for each follower; when that position is calculated, the obstacles of the environment (including the rest of UAVs) are taking into account. For this reason, the grey level of the 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 6.2 shows how the geometry of the formation is affected and how the next *partial goals* are calculated, where the triangular shape distances for the follower 1 and follower 2 are given by Eq. (6.1) and Eq. (6.2), respectively.

$$\begin{aligned} d_1 &= 2 \cdot dist \cdot v, \\ d_2 &= 2 \cdot dist \cdot (B_i) \cdot u, \end{aligned} \tag{6.1}$$

$$\begin{aligned} d_1 &= 2 \cdot dist \cdot v + (2 - B_{i+1}) \cdot dist, \\ d_2 &= 2 \cdot dist \cdot u - (B_{i+1}) \cdot dist, \end{aligned} \tag{6.2}$$

   where *dist* is the safety distance between the UAVs and B_i is the grey level of the follower's current position.

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.

(d) The map T_i^t is obtained applying FM method into W_i^t. Figure 6.2 (d) shows how the FM method is applied taking into account the followers as obstacles.

(e) The gradient descent is applied into T_i^t obtaining the path for each UAV_i from its current position until its *partial goal*.

(f) 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 Fig. 6.4

6.4 Simulation Results

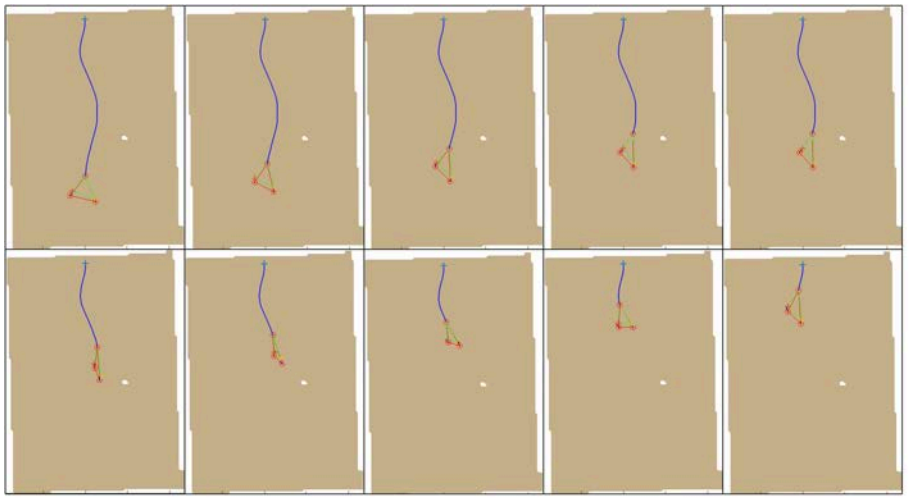
Simulation results have been run under the same software and hardware requirements defined in Chapter 4. The results obtained from this approach are presented next.

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.

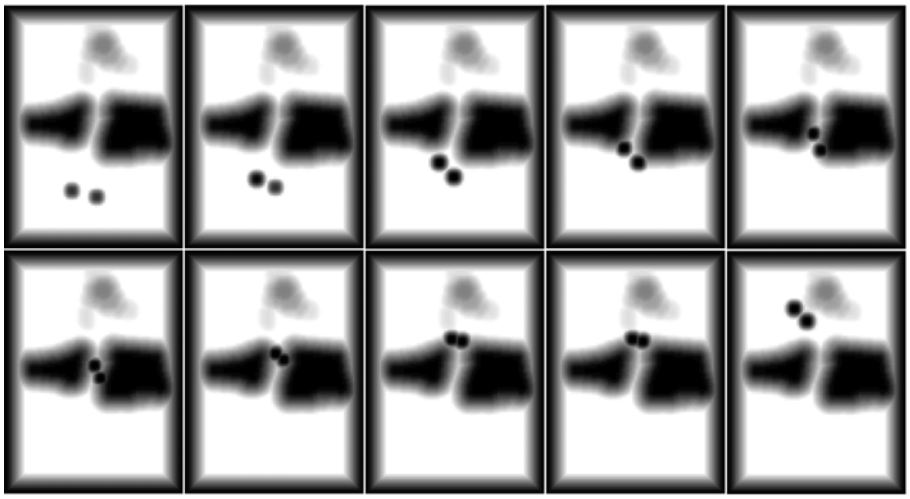
The simulation results carried out show how the formation of three UAVs is maintained during the whole planning and is only deformed when the UAVs avoid the obstacles of the environment. Here, two cases are presented: the first case presents a planning without altitude constraint for the UAVs formation; 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 the 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.

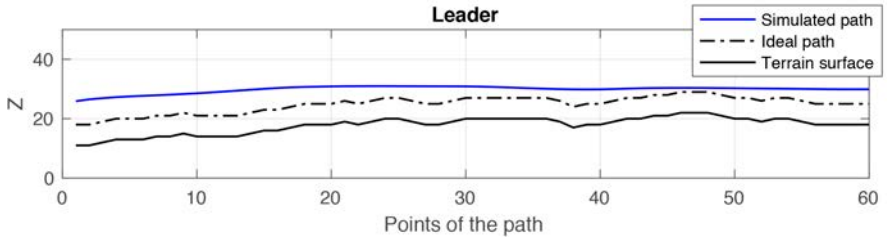


(a)

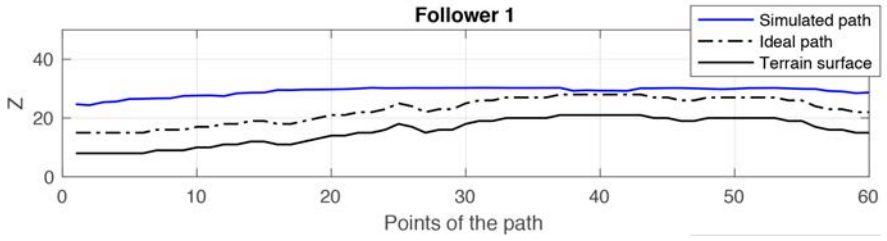


(b)

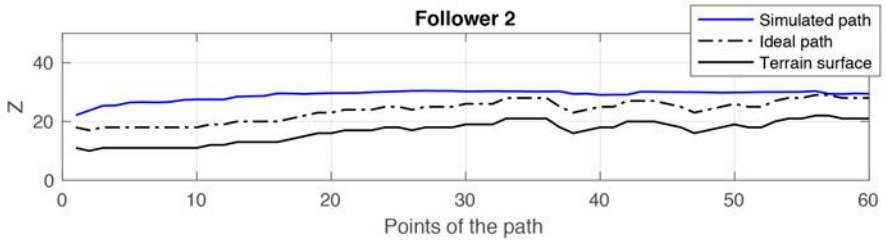
Figure 6.5: *Sequence without flight level restriction: (a) normal map; (b) map W from leader perspective.*



(a)



(b)



(c)

Figure 6.6: *Comparison of the resulting path without flight restriction against the ideal path and the terrain profile: (a) leader; (b) follower 1; (c) follower 2.*

6.4.1 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 6.5 shows the results after running the algorithm for this particular case, presenting the resulting sequence of movements. The distance between UAVs is 8 cells and each frame of the sequence is chosen in time intervals of 20 seconds, approximately.

Fig. 6.5 (a) shows the movements of each elements 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 de real position for each UAV. Figure 6.5 (b) shows the same movements that 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, Fig. 6.6 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 seconds.

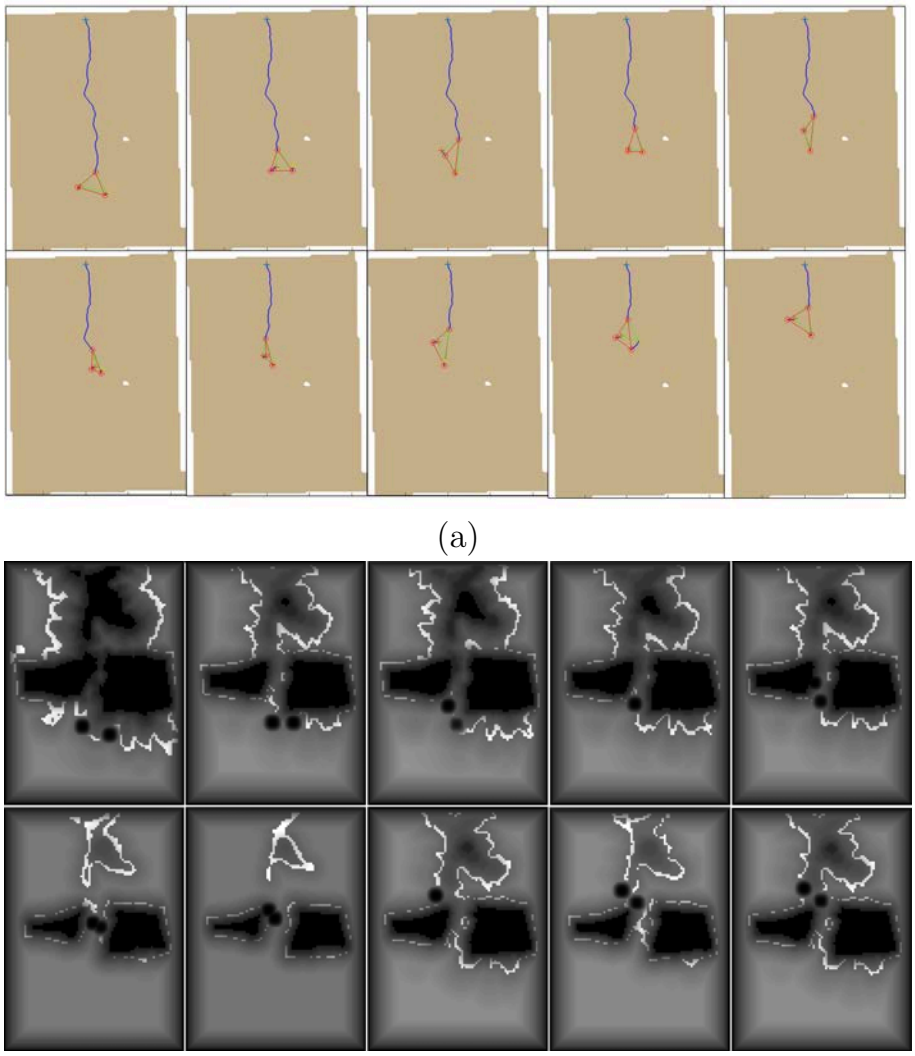
6.4.2 Case 2: Formation with Flight Level Constraint

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 4. In this case, the values of p_1 and p_2 has been fixed to 1 and 0.5, respectively, and the altitude with respect to the ground is 7 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 6.7 (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 seconds. This time increment is due to a higher computation cost imposed by the introduction of the flight level into the algorithm. Figure 6.7 (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.



(a)

(b)

Figure 6.7: *Sequence with flight level 7 restriction: (a) normal map; (b) map W from follower 1 perspective.*

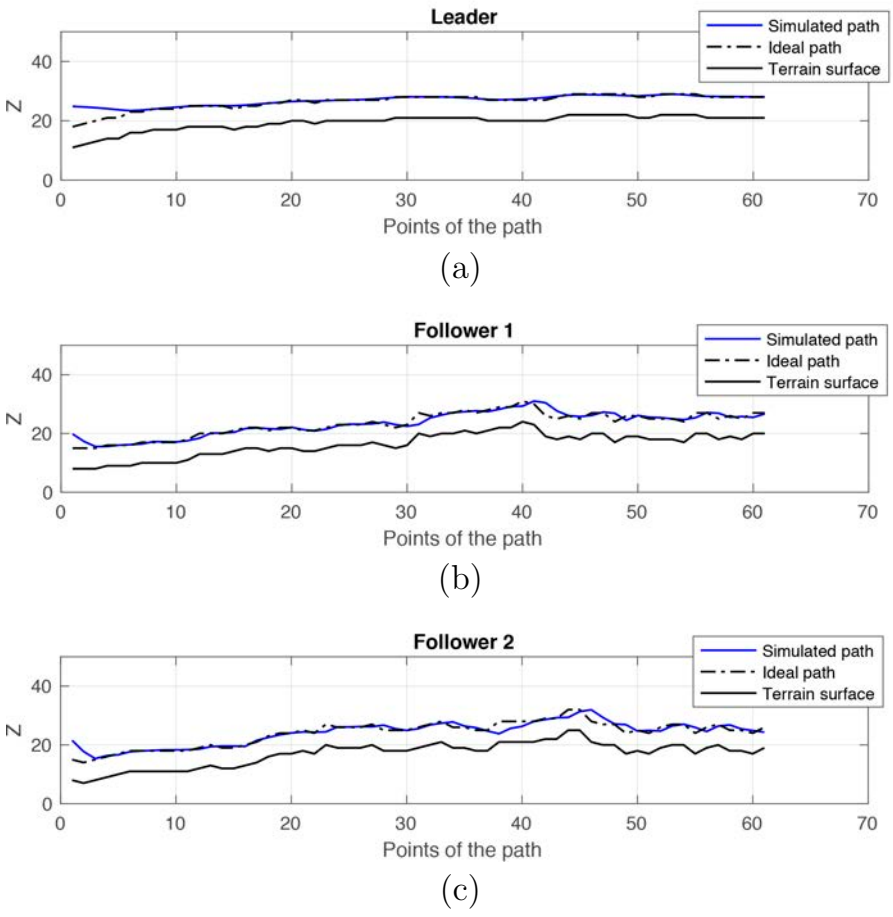


Figure 6.8: *Comparison of the resulting path with flight restriction against the ideal path and the terrain profile: (a) Leader; (b) Follower 1; (c) Follower 2.*

The paths of each agent of the formation can be seen in Fig. 6.8. 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 seconds.

6.5 Conclusions

This chapter has introduced the path planning problem for 3D 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 has shown that the formation is able to adapt its shape so that the obstacles are avoided at the same time that it fulfills the flight level restriction.

Chapter 7

Conclusions and Future Works

This thesis has attempted to contribute in some areas related to path planning for aerial vehicles in 3D environments. The planner approach presented here is based on an improved version of the FM^2 method, which allows to plan UAVs missions taking into account constraints such as flight level, smoothness of the path, safety issues and kinematic restrictions.

Besides, a novel approach for CPP missions has been presented, based on an optimization process that combines the DE algorithm and the FM^2 planner.

Finally, the UAVs formation problem has been introduced and addressed in a first stage using the planner proposed in this thesis.

A wide variety of simulated experiments have been used to illustrate the efficiency and robustness of the approaches presented here when applied to different urban and open field 3D environments.

7.1 Conclusions

The problem of UAVs mission planning is an open research field today and is being addressed by the scientific community with great interest. If future UAVs are going to share our living space, they will need to be able to move safely and have a high level of autonomy when it comes to planning tasks. Every chapter of this thesis attempts to make a step forward in the development of path planning for aerial vehicles in 3D environments. Next, the summary and conclusions of every chapter is presented.

Chapter 2 introduced the FM and FM^2 methods and their mathematical formulations, together with a discussion on the advantages of using FM^2 versus FM. This chapter is fundamental for the understanding of the planning problem and the planner approach presented in this thesis.

Chapter 3 presented a first approach of the planner proposed in this thesis based on FM^2. An adjustment parameter was introduced in order to modify the smoothness and safety distance of the resulting paths. The Dubins airplane model was used to check if the path resulting from the method is feasible according to the constraints of the UAV: its turning rate, climb rate and cruise speed. This first approach gives the reader a perspective of the full potential of the planning tool proposed here, which is later improved in the following chapter.

Chapter 4 introduced the flight level constraint in the planning problem and discussed how the planner proposed fulfills this restriction with the introduction of two parameters in the FM^2 algorithm. Depending on the values of these parameters, the restriction of flight level can be modified, as well as the smoothness and safety margins from the obstacles of the generated paths. The final algorithm is very powerful and flexible, allowing to consider constraints such as flight level, smoothness of the path, safety issues and kinematic restrictions within the planning.

Chapter 5 presented the CPP approach, where the missions focus on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the DE algorithm in combination with the FM^2 planner. The simulation results showed that the approach is robust and can work efficiently in a wide variety of environments.

Chapter 6, finally, extended the planning problem to UAVs formations, addressing the problem in a first stage by using the planner proposed in this thesis. The simulation results presented in the chapter validates the proposal.

7.2 Contributions

The main contributions presented in this thesis are summarized as follows:

- We proposed a novel path planner based on the FM^2 method to be used for mission planning of UAVs in 3D environments. Our approach is robust and feasible enough to be used for whatever aerial vehicle, adapting to its kinematic constraints and ensuring that the path fulfills smoothness and safety restrictions.

- Our planner is intuitive to use. It is based on the natural movement of a wave, so conceptually, it is very easy to understand. Besides, the introduction of two adjustment parameters makes the planning procedure easy to understand and easy to adapt to kinematic, smoothness and safety conditions. Very simple and intuitive relationships between these two parameters are established, and the planning map can be easily conformed following those relationships.

- The computation cost of the algorithm is very low. Though the expansion time of the wave depends on the complexity of the environment, the planner has a fast and efficient computational speed. This fact makes it suitable not only for planning in static environments but also for planning in dynamic maps, where the conditions of the environment are changing.

- Related to the previous contribution, the planner can be used: 1) as a local planning method, for instance in the cases of obstacles avoidance in a dynamic environment, and 2) as a global planning method, with path planning purposes. In both types of applications, the algorithm will find the optimal solution in terms of safety and smoothness. Though the local planning method is not the case of study presented in this thesis, for dynamic environments the algorithm would need to be executed several times with a specific time interval between executions, depending on the rate of change of the environment.

- Our planner has demonstrated to be successful for coverage missions. These missions are focused on path planning to cover a certain area in an environment in order to carry out tracking, search or rescue tasks. The methodology followed uses an optimization process based on the DE algorithm in combination with the FM^2 planner. The DE algorithm evaluates a cost function to determine what the zigzag path with the minimum cost is, according to its steering angle. This optimization process allows achieving the most optimal zigzag path in terms of distance travelled by the UAV to cover the whole area. Then, the FM^2 planner is applied to generate the final path according to the steering angle of the zigzag bands resulting from the DE algorithm.

- The planner proposed can be also used with success for UAVs formations planning. It has been demonstrated that the formation is able to adapt its shape so that the obstacles are avoided at the same time that it fulfills the flight level restriction. All this with a very competitive computational cost.

7.3 Future Works

Path planning with application to aerial vehicles is an open research field and there is a wide range of challenges to be faced. Regarding those arising from this thesis, we can remark the following ones:

- Even if very simple and intuitive relationships between the adjustment parameters of our planner have been established in order to conform the planning, a further research step is currently focusing on the study of the explicit relationship between the UAV kinematics and the value of these parameters, as a first approach to the optimality study.

  Besides, from our study the dynamic restrictions for the UAV with respect to speed and acceleration ranges can be determined. A further study will deal with the analytical and explicit relationship between the adjustment parameters and the permissible dynamic ranges of the UAV.

  In order to simplify the study on the relationship between the kinematic and dynamic restrictions and the values of the adjustment parameters, these two can be established by fixing $p_1 = 1$ and varying p_2, or setting $p_1 = k$ and $p_2 = 1 - k$ and then setting k to different values to create different paths. The introduction of parameter k to rule that relationship can help to simplify future studies.

- The application of our planner to dynamic environments needs to be studied. As stated before, the algorithm is suitable for being applied to changing environments, acting in this case as a local planning approach. However, aspects such as the dynamic range of the environment and the computational cost need to be approached to determine to what extent the algorithm is applicable.

- In this thesis just a first approach to UAVs formations has been presented. This just opens the path to a deeper analysis on this field, that will be treated in a future Ph.D thesis.

- Finally, the planner approach needs to be implemented in a real UAV. A preliminary study of this implementation has been done in the Master Thesis by David Rodríguez (Rodríguez (2017)), under the context of this Ph.D work. First results show that the planning algorithm is easy to implement, though the UAV platform selected for experiments (Parrot's Bebop Drone) was not robust enough to be used for planning applications. As a future work, the planner will be tested in other real platforms equipped to accomplish planning missions.

Bibliography

Acevedo, J., Arrue, B., Maza, I., and Ollero, A. (2013a). Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. *J. Intell. Robot. Syst.*, 70(1–4):329–345.

Acevedo, J. J., Arrue, B. C., Maza, I., and Ollero, A. (2013b). Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints. *International J. Advanced Robotic Systems*, 10:1–13.

Ahmad, A., Nascimento, T., Conceicao, A., Molina, A., and Lima, P. (2013). Perception-driven multi-robot formation control. In *IEEE International Conference on Robotics and Automation*, page 1851–1856, Karlsruhe.

Alvarez, D., Gómez, J., Garrido, S., and Moreno, L. (2014). 3D robot formations planning with fast marching square. In *IEEE Int. Conf. Auton. Robot Syst. and Compet.*, pages 59–64.

Amirreza, K., Hossein, M., and Mehdi, F. (2014). A new method for trajectory optimization in terrain following flight. In *22nd Iranian Conference on Electrical Engineering (ICEE 2014)*, pages 1203–1208, Tehran, Iran.

Arismendi, C., Álvarez, D., Garrido, S., and Moreno, L. (2015). Nonholonomic motion planning using the fast marching square method. *Int. J. Advanced Robotic Syst.*, 12(56):1–14.

Babiarz, A. and Jaskot, K. (2013). The concept of collision-free path planning of UAV objects. *Advanced Technologies for Intelligent Systems of National Border Security*, 440:81–94.

Barrientos, A., Colorado, J., del Cerro, J., Martínez, A., Rossi, C., Sanz, D., and Valente, J. (2011). Aerial remote sensing in agriculture: A practical approach to

area coverage and path planning for fleets of mini aerial robots. *J. Field Robotics*, 28(5):667–689.

Barrientos, A., Gutiérrez, P., and Colorado, J. (2009). Advanced UAV trajectory generation: Planning and guidance. In Lam, T. M., editor, *Aerial Vehicles*, chapter 4, pages 55–82. InTech, London.

Beard, R. and McLain, T. (2012). In *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press.

Beard, R. and McLain, T. (2013). Implementing Dubins airplane paths on fixed-wing UAVs. In *Handbook of Unmanned Aerial Vehicles*, pages 1677–1701. Springer Netherlands.

Bhatia, A., Graziano, M., Karaman, S., Naldi, R., and Frazzoli, E. (2008). Dubins trajectory tracking using commercial off-the-shelf autopilots. In *AIAA Guidance, Navigation and Control Conference and Exhibit*.

Bidabad, B. and Sedaqat, M. (2010). Geometric modeling of Dubins airplane movement and its metric. *Amirkabir International Journal of Electrical and Electronics Engineering*, 42(1):9–16.

Bo-Bo, M. and Xiaoguang, G. (2010). UAV path planning based on Bidirectional Sparse A* Search Algorithm. In *Int. Conf. on Intelligent Computation Technology and Automation*, pages 1106–1109.

Bouabdallah, S. (2007). *Design and Control of Quadrotors with Applicationto Autonomous Flying*. PhD thesis, Ecole polytechnique federale de Lausanne, Lausanne.

Cao, Z., Xie, L., Zhang, B., Wang, S., and Tan, M. (2003). Formation constrained multi-robot system in unknown environments. In *IEEE International Conference on Robotics and Automation*, pages 735–740.

Cekmez, U., Ozsiginan, M., and Sahingoz, O. K. (2016). Multi colony ant optimization for UAV path planning with obstacle avoidance. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 47–52, Arlington, VA USA.

Chitsaz, H. and LaValle, S. (2007). Time-optimal paths for a Dubins airplane. In *IEEE Conf. on Decision and Control*, pages 2379–2384.

Choi, H. J. (2014). *Time-optimal paths for a Dubins car and Dubins airplane with a unidirectional turning constraint*. PhD thesis, Michigan University, EE.UU.

Clifton, M., Paul, G., Kwok, N., Liu, D., and Wang, D. L. (2008). Evaluating performance of multiple RRTs. In *2008 IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications*, pages 564–569.

Dewan, A., Mahendran, A., Soni, N., and Krishna, K. (2013). Heterogeneous ugv-mav exploration using integer programming. In *AIAA/USU Conference on Small Satellites on Intelligent Robots and Systems*, pages 5742–5749.

Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516.

Galceran, E. and Carreras, M. (2013). Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4159–4164, Karlsruhe, Germany.

Garrido, S., Malfaz, M., and Blanco, D. (2013). Application of the Fast Marching method for outdoor motion planning in robotics. *J. Robot. Auton. Syst.*, 61(2):106–114.

Garrido, S., Moreno, L., Abderrahim, M., and Blanco, D. (2009). FM2: a real-time sensor-based feedback controller for mobile robots. *Int. J. Robot. Autom.*, 24(1):3169–3192.

Garrido, S., Moreno, L., and Alvarez, D. (2015). Path planning for Mars Rovers using the Fast Marching method. In *Robot 2015: Second Iber. Robot. Conf.*, volume 1, pages 93–105. Springer International Publishing.

Garrido, S., Moreno, L., and Lima, P. (2011). Robot formation motion planning using Fast Marching. *J. Robot. Auton. Syst.*, 59(9):675–683.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company.

González, V., Monje, C. A., Moreno, L., and Balaguer, C. (2016). Fast Marching Square method for UAVs mission planning with consideration of Dubins model constraints. In *20th IFAC Symp. Autom. Control in Aerosp. (ACA)*, volume 49, pages 164–169, Sherbrooke, Quebec, Canada.

González, V., Monje, C. A., Moreno, L., and Balaguer, C. (2017a). UAVs mission planning with flight level constraint using Fast Marching Square method. *Robotics and Autonomous Systems*, 94:162–171.

González, V., Monje, C. A., Moreno, L., and Balaguer, C. (2017b). UAVs mission planning with imposition of flight level through Fast Marching Square. *J. Cybern. Syst.*, 48(2):102–113.

Gómez, J., Lumbier, A., Garrido, S., and Moreno, L. (2013). Planning robot formations with Fast Marching square including uncertainty conditions. *J. Robot. Auton. Syst.*, 61(2):137–152.

Gómez, J., Vale, A., Garrido, S., and Moreno, L. (2015a). Performance analysis of Fast Marching-based motion planning for autonomous mobile robots in ITER scenarios. *J. Robot. Auton. Syst.*, 63:36–49.

Gómez, J. V. (2015). *Fast Marching Methods in path and motion planning: improvements and high-level applications*. PhD thesis, Dept. Syst. Eng. Autom., Univ. Carlos III, Madrid, Spain.

Gómez, J. V., Pardeiro, J., and Gely, P. (2015b). N-Dimensional Fast Marching method V1.0. URL: goo.gl/iabfn3. Accessed 15.02.17.

Hanson, C., Richardson, J., and Girard, A. (2011). Path planning of a Dubins vehicle for sequential target observation with ranged sensors. In *Control Conf.*, page 1698–1703.

Hauert, S., Zufferey, J., and Floreano, D. (2009). Reverse-engineering of artificially evolved controllers for swarms of robots. In *IEEE Congress on Evolutionary Computation*, pages 55–61.

Hino, T. (2010). Simple formation control scheme tolerant to communication failures for small unmanned air vehicles. In *International Congress of the Aeronautical Sciences*, pages 1–9.

Kamyar, R. and Taheri, E. (2014). Aircraft optimal terrain/threat-based trajectory planning and control. *J. Guidance, Control and Dynamics*, 37(2):466–483.

Kanjanawanishkul, K. and Zell, A. (2008). A model-predictive approach to formation control of omnidirectional mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 2771–2776, Nice.

Krabar, S. (2011). Reactive obstacle avoidance for rotorcraft uavs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4967–4974, San Francisco.

Lee, D. and Shim, D. H. (2014). RRT-based path planning for fixed-wing UAVs with arrival time and approach direction constraints. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 317–328, Orlando, FL, USA.

Likhachev, M., Keller, J., Kumar, V., Dobrokhodov, V., Jones, K., Wurz, J., and Kaminer, I. (2013). Planning for opportunistic surveillance with multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5750–5757, Tokyo.

Lin, Y. and Saripalli, S. (2014). Path planning using 3D Dubins curve for unmanned aerial vehicles. In *Int. Conf. Unmanned Aircr. Syst.*, pages 296–304.

Martin, M., Klupar, P., Kilberg, S., and Winter, J. (2001). Techsat 21 and revolutionizing space missions using microsatellites. In *IAIAA/USU Conference on Small Satellites*, pages 1–10.

McGee, T., Spry, S., and Hedrick, J. (2005). Optimal path planning in a constant wind with a bounded turning rate. In *AIAA Guidance, Navigation and Control Conference and Exhibit*.

Nikolos, I. K., Zografos, E. S., and Brintaki, A. N. (2007). UAV path planning using evolutionary algorithms. *Innovations in Intelligent Machines-1*, 70:77–111.

Ogren, P., Egerstedt, M., and Hu, X. (2002). A control lyapunov function approach to multiagent coordination. *IEEE Trans. Robot. Autom.*, 18(5):847–851.

Ok, K., Ansari, S., Gallagher, B., Sica, W., Dellaert, F., and Stilman, M. (2013). Path planning with uncertainty: Voronoi uncertainty fields. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4596–4601, Karlsruhe, Germany.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed-algorithms based on hamilton-jacobi formulations. *J. Computational Phys.*, 79(1):12–49.

Peyre, G. (2004). Toolbox Fast Marching. URL: goo.gl/cor8cs. Accessed 17.02.17.

Rodríguez, D. (2017). *Fast Marching Square Applied to Parrot's Bebop Drone*. PhD thesis, Universidad Carlos III de Madrid, Spain.

Sethian, J. (1996). A Fast Marching level set method for monotonically advancing fronts. In *Natl. Acad. Science*, volume 93, page 1591–1595, USA.

Shen, S., Michael, N., and Kumar, V. (2011). 3d estimation and control for autonomous flight with constrained computation. In *IEEE International Conference of Robotics and Automation*, pages 4967–4974, Shanghai.

Storn, R. and Price, K. (1997). Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11(14):341–359.

Torresa, M., Pelta, D. A., Verdegaya, J. L., and Torres, J. C. (2016). Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Systems with Applications*, 55:441–451.

Trujillo, M. M., Darrah, M., Speransky, K., DeRoos, B., and Wathen, M. (2016). Optimized flight path for 3D mapping of an area with structures using a multirotor. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 905–910, Arlington, VA USA.

Valero-Gómez, A., Gómez, J. V., Garrido, S., and Moreno, L. (2013). Fast Marching methods in path planning. *IEEE Robot. Autom. Mag.*, 20(4):111–120.

Xixia Sun, Chao Cai, J. Y. and Shen, X. (2014). Route evaluation for unmanned aerial vehicle based on type-2 fuzzy sets. *Engineering Applications of Artificial Intelligence*, 39:132–145.

Xue, Q., Cheng, P., and Cheng, N. (2014). Offline path planning and online replanning of UAVs in complex terrain. In *2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 2287–2292, Yantai, China.

Yan, F., Liu, Y., and Xiao, J. (2014). Path planning in complex 3D environments using a probabilistic roadmap method. *Int. J. Autom. Comput.*, 10(6):525–533.

Yu, W., Chen, G., and Cao, M. (2010). Distributed leader-follower flocking control for multi-agent dynamical systems with time-varying velocities. *Syst. Control Lett.*, 59(9):543–552.

Zamuda, A. and Sosa, J. D. H. (2014). Differential Evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures. *Applied Soft Computing*, 24:95–108.

Zhang, B., Tang, L., and Roemer, M. (2014). Probabilistic weather forecasting analysis for unmanned aerial vehicle path planning. *J. Guidance, Control, and Dynamics*, 37(1):309–312.

Zhang, M., Shen, Y., Wang, Q., and Wang, Y. (2010). Dynamic artificial potential field based multi-robot formation control. In *IEEE Instrumentation and Measurement Technology Conference*, page 1530–1534, Austin.

Zhanga, X. and Duana, H. (2015). An improved constrained Differential Evolution algorithm for unmanned aerial vehicle global route planning. *Applied Soft Computing*, 26:270–284.