sensors

MDPI

Article

FM^2 Path Planner for UAV Applications with Curvature Constraints: A Comparative Analysis with Other Planning Approaches

Santiago Garrido *, Javier Muñoz, Blanca López, Fernando Quevedo, Concepción A. Monje and Luis Moreno

Robotics Lab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Av. Universidad 30, 28911 Madrid, Spain; jamunozm@ing.uc3m.es (J.M.); bllopezp@ing.uc3m.es (B.L.); fquevedo@ing.uc3m.es (F.Q.); cmonje@ing.uc3m.es (C.A.M.); moreno@ing.uc3m.es (L.M.)

* Correspondence: sgarrido@ing.uc3m.es

Abstract: This paper studies the Fast Marching Square (FM^2) method as a competitive path planner for UAV applications. The approach fulfills trajectory curvature constraints together with a significantly reduced computation time, which makes it overperform with respect to other planning methods of the literature based on optimization. A comparative analysis is presented to demonstrate how the FM^2 approach can easily adapt its performance thanks to the introduction of two parameters, saturation α and exponent β, that allow a flexible configuration of the paths in terms of curvature restrictions, among others. The main contributions of the method are twofold: first, a feasible path is directly obtained without the need of a later optimization process to accomplish curvature restrictions; second, the computation speed is significantly increased, up to 220 times faster than other optimization-based methods such as, for instance, Dubins, Euler–Mumford Elastica and Reeds–Shepp. Simulation results are given to demonstrate the superiority of the method when used for UAV applications in comparison with the three previously mentioned methods.

Keywords: fast marching; Dubins; Euler–Mumford Elastica; Reeds–Shepp; curvature constraint

check for updates

Citation: Garrido, S.; Muñoz, J.; López, B.; Quevedo, F.; Monje, C.A.; Moreno, L. FM^2 Path Planner for UAV Applications with Curvature Constraints: A Comparative Analysis with Other Planning Approaches. *Sensors* **2022**, *22*, 3174. https://doi.org/10.3390/s22093174

Academic Editor: Sergio Toral Marín

Received: 1 March 2022
Accepted: 19 April 2022
Published: 21 April 2022

1. Introduction

Autonomous robots and ground, marine and aerial vehicles have increased their technological level over the last 20 years, today being a mature technology in some fields and increasing their maturity level in others. Their capabilities and autonomy are continuously increasing with new applications, operating for longer periods and in more complex environments and situations. One of the factors that significantly affects the vehicle autonomy level is the path-planning ability to guarantee accurate, safe and executable trajectories. This path-planning capability is critical for most robots, and particularly challenging for UAVs.

Motion planning of robots is an extensively researched area [1]. In most domains, and for different reasons, planners take into consideration only geometric constraints. Widely used approaches for the path-planning problem are grid-based methods based on graph-search strategies to extract the optimal path, such as Dijkstra's Algorithm [2] and A* [3]. However, these methods depend on the discretization that should balance computational requirements and accuracy, and that, on occasion, can lead to non-optimal solutions or even to failures if the grid does not have enough resolution.

A different and widely used class of algorithms are sampling-based methods, or stochastic ones, such as probabilistic roadmap (PRM) [4] and rapidly exploring random tree (RRT) [1]. These methods rely on randomly sampling configurations from the C-Space and connecting them to each other. These approaches are not limited by the resolution of the grid, as in the case of search-based methods, but are strongly limited by the sampling process. The solution will converge towards the optimal as the number of iterations goes towards infinity. A drawback of sampling-based algorithms is the lack of guarantees to find a solution and optimality.

For non-holonomic robots such as UAVs, finding a collision-free path is not sufficient: the path also needs to be executable, which means that the vehicle's execution of the path should verify the vehicle's motion constraints. Different variations of the previously mentioned methods have dealt with the way to take into account these constraints at planning time. Yan et al. [5] used A* with a circular search to ensure path feasibility for a UAV. Pivtoraiko et al. [6] extended A* to connect states based on Reeds–Shepp curves. Another approach to solve the path planning problem under motion constraints with a search-based method is the hybrid A* (HA*) [7]. In a similar way, different sampling-based algorithms consider the motion constraints of the robot at planning. RRT was initially developed for kinematic planning [8]. Expansive space trees (EST) [1] performs random sampling from a random state to explore the search space with kinematic constraints. Hernández et al. added Dubins curves [9] to enable RRT* to be used with kinematic constraints [10]. Stable sparse-RRT (SST) was used for AUVs by Pairet et al. [11].

However, an important question arises: if the planning method is able to generate smooth enough paths to verify the vehicle's motion constraints, is it then necessary to introduce these constraints in the path planning method? Moreover, if we introduce them, what are the advantages and disadvantages of doing so? These questions are addressed in this work.

Other common approaches try to solve the path-planning problem include minimising the first and second derivatives of the line curvature by means of a quadratic optimisation problem [12] or solving an additional Optimal Control Problem in order to find the curvature of the centre line [13]. In contrast to these approaches, Heilmeier et al. [14] performed an approximate spline regression and generated a smooth reference line, which varies slightly from the centre line, and thus offers more scope for action in the smoothing procedure. However, these methods are not appropriate for path planning and collision risk management strategy for multi-UAVs in 3D environments [15], where maps with a grid of 1000 $\times$ 1000 $\times$ 40 or similar dimensions are used and multiple UAVs have to be planned. Our approach focuses on solving this problem avoiding the use of optimization and providing a very fast computation of the resulting paths.

Throughout the years, the Fast Marching Square (FM^2) method has demonstrated its robust performance when it comes to the calculation of the paths to be followed by UAVs during the execution of their missions. As a path planner, the method has been demonstrated to obtain trajectories which are smooth, with no sharp turns, and safe (far enough from obstacles) to be executed by ground, submarine and aerial robots. All this is achieved without the need to include the vehicle's motion constraints in the computation of the algorithm.

In this work, a study is carried out to demonstrate how FM^2 can overperform other motion constraints based approaches such as Dubins, Euler–Mumford Elastica and Reeds–Shepp Forward methods when applied for the path planning of quadcopters and fixed-wing drones. These methods have been selected due to the fact that they are a reference in the field when it comes to optimization-based planners and many others are based on them. Aspects such as the curvature restriction of the trajectories and the computation time involved in the obtaining of the paths will be addressed.

In 1691, J. Bernoulli [16,17] formulated the classic version of the deformation problem of a flexible rod, or isotropic ideal elastic bar with a uniform elastic stiffness, which bends due to being subjected to external forces and moments applied at its ends. This is called the Euler–Mumford Elastica problem.

Dubins curves are those that connect two points with the shortest path, with the restriction that the curvature cannot exceed a given value and that the vehicle can only go forward. In 1957, Dubins [9] showed that these curves are composed of arcs of circumference with the maximum allowed curvature and line segments.

The Reeds–Shepp problem (1990) [18] is similar to that of Dubins, but the arcs can have different curvatures. Usually, the vehicle can also travel in reverse direction. However, in this paper, only the Reeds–Shepp Forward method is considered.

The study presented in this paper will compare the trajectories obtained by each of these three methods with those obtained using the FM^2 approach when varying two performance parameters, α and β, that will be defined and discussed later. In order to do this, the distance matrix will be calculated using the Fréchet distance [19] between two curves. This distance is a measure of similarity between two discrete curves, P and Q (see Figure 1). It is defined as the minimum cord-length sufficient to join a point traveling forward along P and one traveling forward along Q, although the rate of travel for either point may not necessarily be uniform. Moreover, the area between two discrete curves in an XY plane will be also used as a distance for comparison.

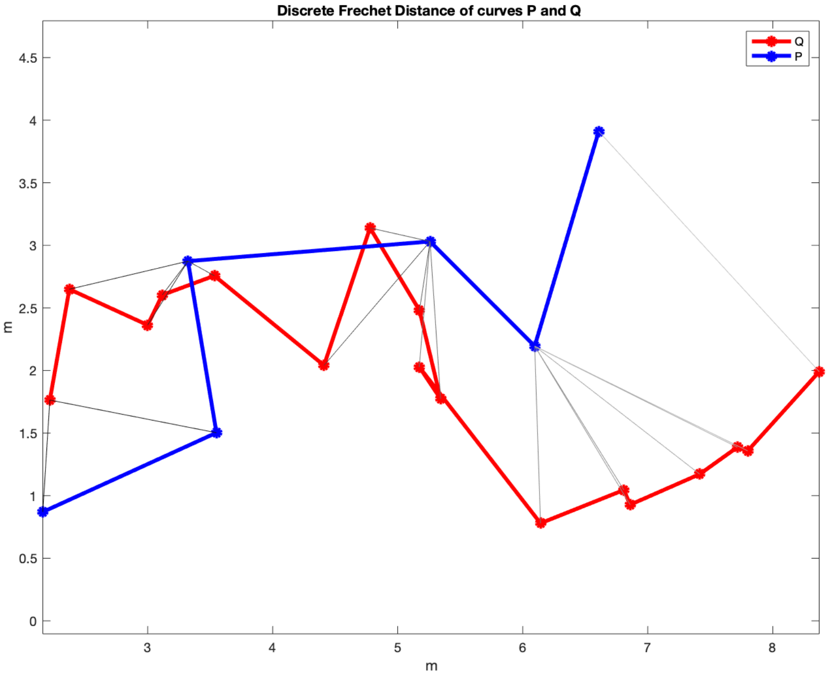


Figure 1. The Fréchet distance between two curves is the maximum length of all the gray lines.

As a first step for comparison, Table 1 shows the computation times of the four path-planning methods of the study taken to calculate the different trajectory solutions (they will be addressed in the following sections). The time difference is of three orders of magnitude in favor of the FM^2 method, which is approximately 220 times faster than the rest. This is the reason why a deeper analysis of the results is of interest and a further study of the curvature problem will be carried out.

Table 1. Computation times (in seconds) of the four path planning methods, Dubins, Euler–Mumford Elastica, Reeds–Shepp and FM^2, taken to calculate the 20 trajectories addressed in the paper.

	Dubins	Euler–Mumford Elastica	Reeds–Shepp	FM^2
Computation time (secs)	18.936	19.199	19.577	0.087

2. Description of the Planning Approaches

This section describes the four path-planning approaches studied and compared in this work, namely, FM^2, Dubins, Euler–Mumford Elastica and Reeds–Shepp. First of all, our FM^2 approach will be presented, followed by the introduction of the other three well-known planning methods.

2.1. *FMM and FM*^{2} *Methods*

The Fast Marching method (FMM) was developed by J.A. Sethian [20] in 1996 to solve the Eikonal Equation (1), which is the equation that models the propagation of light and other electromagnetic waves in a medium with refractive index $1/F(x)$, where $F(x)$ is the velocity of propagation of the medium and $T(x)$ is the arrival time of the wave front.

$$|\nabla T|F = 1 \tag{1}$$

The objective of the FMM is to solve the discretized form of this equation in an orthogonal mesh. Once the equation is solved, the result is a funnel-shaped surface and the path of minimum distance can be found using the gradient method [21].

The FMM models phenomena based on the evolution of a normally self-propagating wavefront, such as the propagation of light or that of an oil slick in the sea. The following explanation is in two dimensions, but it can be extended to three or more dimensions. Let T_{ij} be the time in which the wavefront crosses the point (i, j) of a two-dimensional map, satisfying $|\nabla T|F = 1$, the Eikonal equation. $F = F_{ij}$ represents the velocity function of the wave at that point on the map. The main idea of the algorithm is to compute the arrival time map using only upwind values adjacent to the wavefront which, as shown in [22], leads to a simplified solution as follows:

$$\left(\frac{T - T_x}{\Delta x}\right)^2 + \left(\frac{T - T_y}{\Delta y}\right)^2 = \frac{1}{{F_{ij}}^2}, \tag{2}$$

where T_x and T_y are the minimum arrival time between the neighbors in X and Y axes, respectively, and F_{ij} is the speed of propagation in the cell for which the arrival time is being computed. Equation (2) is a regular quadratic equation of the form $aT^2 + bT + c = 0$, being:

$$\begin{aligned} a &= \Delta_{\mathrm{x}}^2 + \Delta_{\mathrm{y}}^2, \\ b &= -2(\Delta_{\mathrm{y}}^2 T_{\mathrm{x}} + \Delta_{\mathrm{x}}^2 T_{\mathrm{y}}), \\ c &= \Delta_{\mathrm{y}}^2 T_{\mathrm{x}}^2 + \Delta_{\mathrm{x}}^2 T_{\mathrm{y}}^2 - \frac{\Delta_{\mathrm{x}}^2 \Delta_{\mathrm{y}}^2}{F_{\mathrm{ij}}^2}, \end{aligned} \tag{3}$$

where, in order to simplify the notation, we assume that the grid is composed of unit square cells, that is, $\Delta x = \Delta y = 1$.

The complete procedure for calculating the FMM solution is detailed in Algorithm 1. While running, the algorithm classifies map points into three sets: frozen, open, and unvisited. Frozen points are those for which the arrival time can no longer change. Unvisited points are those that have not been processed yet. Finally, the open points are those that can be considered as an interface between frozen and unvisited regions of the map, belonging to the propagating wave front.

In the first step of the algorithm, initialization, all cells in the map are initialized to infinity (or the maximum value possible in the computer) and are set to unvisited, except for the starting point (the destination point of the path planning), which is set with an arrival time of 0 and is considered as the first open point.

Algorithm 1 Fast Marching Method

```
Require: A grid map $X$ of size $m \times n$, source point $x_0$.
Ensure: The grid map $X$ with the $T$ value set for all cells.
    Initialization.
for all $x \in X$ do
    $T(x) \leftarrow \infty$;
end for
$T(x_0) \leftarrow 0$;
$frozen \leftarrow x_0$;
$open \leftarrow \mathcal{N}(x_0)$;                                ▷ Neighbors of $x_0$.
$open \leftarrow X \backslash (frozen \cup open)$;
    Iteration.
while $frozen \neq X$ do
    $x_1 \leftarrow \underset{x \in open}{\arg\min} d(x)$;
    for all $\mathbf{x}_i = \mathcal{N}(x_1) \in T \cap \notin frozen$ do
        UPDATE($\mathbf{x}_i$);
        $open \leftarrow open \cup \{\mathbf{x}_i\}$;
    end for
    $open \leftarrow open \backslash \{x_1\}$;                      ▷ Updating sets.
    $frozen \leftarrow frozen \cup \{x_1\}$;
end while
```

At each iteration, the open point with the smallest value of $T(x)$ is chosen and set as frozen. Then, the arrival time of its von-Neumann neighbors (if they are not labeled as frozen) is analyzed by solving Equation (2) and checking if the calculated value is less than the real one and labeled as open (called *UPDATE* in the algorithm). This procedure continues until all points are set to frozen or the start point of the route is reached.

The minimal distance path has the problem that it can be too close to corners and has sharp turns (Figure 2). To avoid this drawback, the FM^2 method was developed [23–25], which consists of applying the FMM twice. The first takes as the origin of the expansion of the wave all the points corresponding to the walls and obstacles. This provides a kind of distance or first potential W of gray levels, or wave expansion velocities, darker near the walls and obstacles (slower) and lighter at the furthest points (faster).

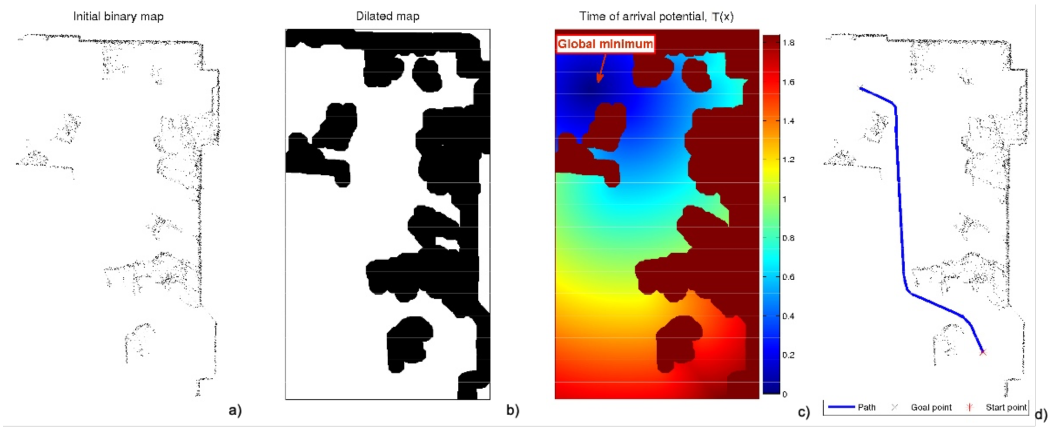


Figure 2. Process of obtaining the FMM trajectory. (**a**) Raw data obtained by the laser sensor; (**b**) enlargement of the previous subfigure; (**c**) expansion of the wave starting from the destination point; (**d**) path obtained by the gradient method over the potential corresponding to the previous subfigure.

On this first potential W, the FMM can be applied again, taking the destination point as the origin of the expansion of the wave and expanding the waves until reaching the origin point. Next, the gradient method is applied to obtain the fastest path according to the metric given by W. In this way, a smoother trajectory is obtained which moves away from walls and corners (Figure 3).

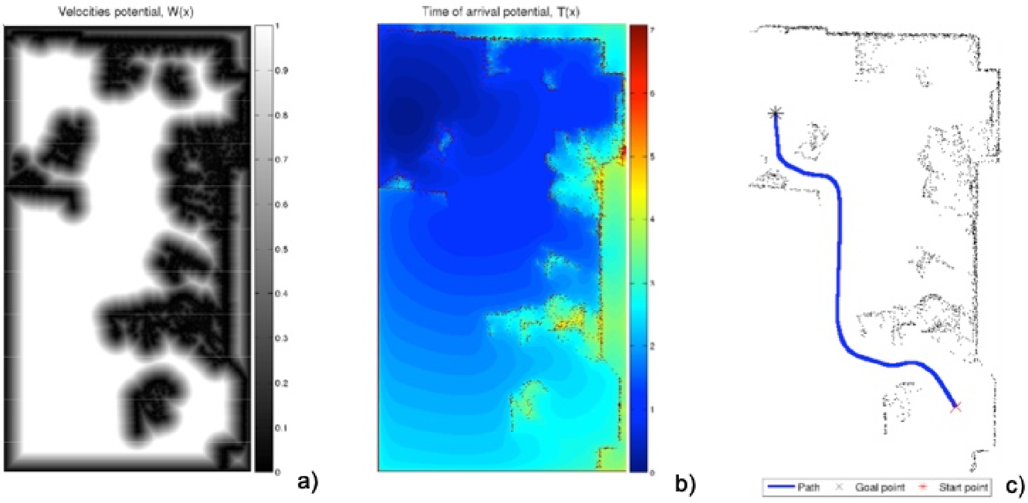


Figure 3. Process of obtaining the FM^2 trajectory with the same raw data obtained by the laser sensor of Figure 1. (**a**) First potential $W(x)$ obtained with the expansion of the wave from the walls and obstacles, saturated with $\alpha = 0.8$; (**b**) expansion of the wave starting from the destination point; (**c**) path obtained by the gradient method over the potential corresponding to the previous subfigure.

There are two parameters affecting the curvature of the resulting paths from the FM^2 approach:

- The first is the saturation parameter α, which consists of setting a maximum value, between 0 and 1, from which all the values of the matrix W are equal to 1.
  In Figures 3 and 4, for α = 0.8 and α = 0.6, respectively, it can be seen that the white area away from obstacles is more or less large. A saturation value closer to zero implies less curved trajectories, but with more angle at certain points and closer to obstacles. The trajectory of Figure 3c) also presents a curvature in some points bigger than that of Figure 4c).
- The second parameter, β, is an exponent between 0 and 1 to which each coefficient of matrix W of the first potential is raised. Figures 5 and 6 show the behaviour of the algorithm for varying values of β. The closer the exponent to zero, the clearer the image and the less curvature of the trajectories. The closer the exponent to one, the more similar the curvature to the Medial Axis Transform (MAT), although the smoother the trajectories and the more separated from obstacles (safer paths).
  It is also possible to raise each coefficient of the matrix to numbers greater than one, but the results are not of interest because they are more and more similar to the trajectories obtained with a Voronoi diagram that have sharp edges.

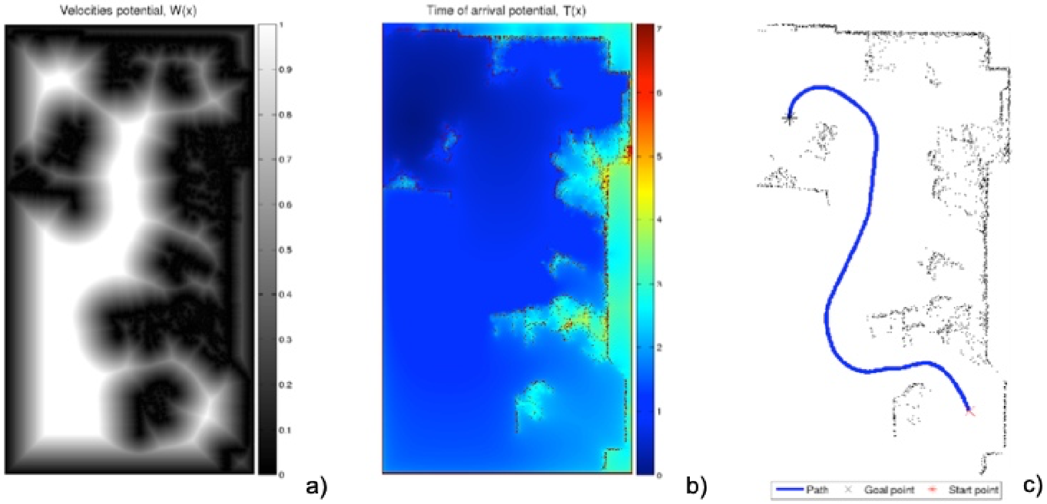


Figure 4. Process of obtaining the FM^2 trajectory with the same raw data obtained by the laser sensor of Figure 1. (**a**) First potential $W(x)$ obtained with the expansion of the wave from the walls and obstacles, saturated with $\alpha = 0.6$; (**b**) expansion of the wave starting from the destination point; (**c**) path obtained by the gradient method over the potential corresponding to the previous subfigure.

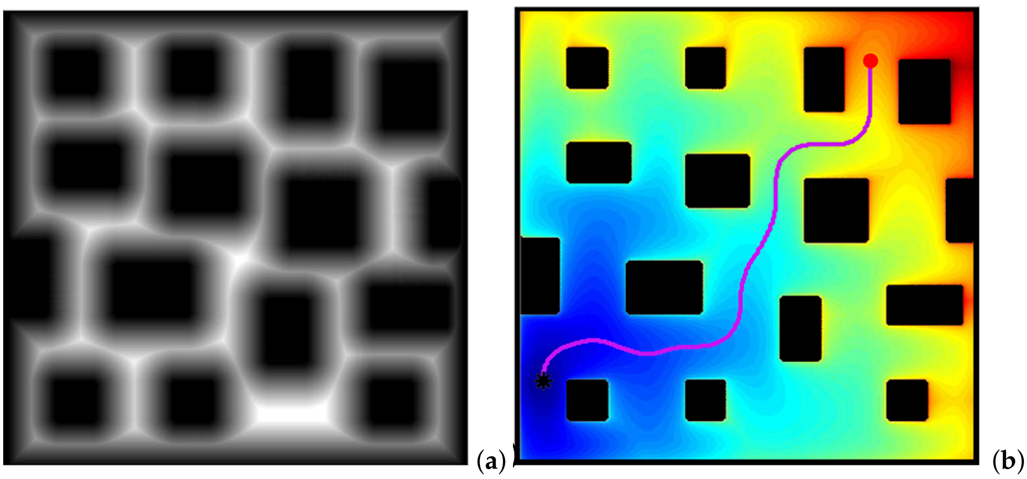


Figure 5. Obtaining FM^2 path in a city environment. (**a**) First potential $W(x)$; (**b**) second potential and the path obtained with the gradient method.

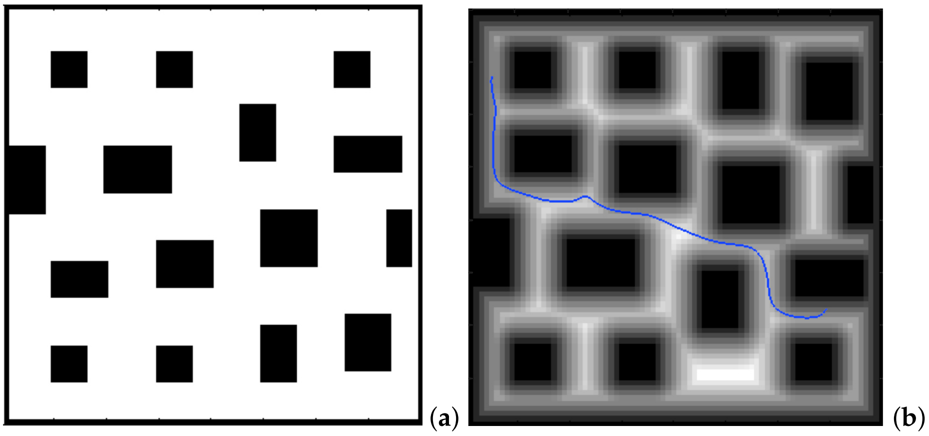


Figure 6. *Cont.*

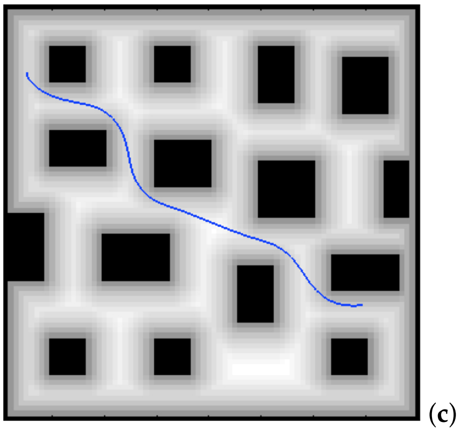


Figure 6. Two FM^2 paths in a city environment for different values of the exponent β of the first potential $W(x)$. (**a**) City environment; (**b**) first potential $W(x)$ and the path for $\beta = 0.9$; (**c**) first potential $W(x)$ and the path for $\beta = 0.4$.

2.2. Dubins, Euler–Mumford Elastica and Reeds–Shepp methods

Mirebeau [26,27] developed numerical methods to calculate trajectories that minimize a cost function to obtain the paths corresponding to the models of the Dubins car, the Euler–Mumford Elastica and the Reeds–Shepp car using the Fast Marching approach with stencils method. For this, he discretizes the generalized Eikonal equation, also called the first-order static Hamilton–Jacobi–Bellman equation. The total cost function for a curve $\mathbf{x} : [0, T] \to \mathcal{R}^{\in}$, parameterized with unit velocity, is

$$\int_0^T \gamma(\mathbf{x}(s), \dot{\mathbf{x}}(s))\mathcal{C}(|\ddot{\mathbf{x}}(s)|))ds, \tag{4}$$

where $\gamma : R^2xS^1 \to]0, \infty[$ is a continuous cost function, which in our case is taken as the unit, and the path curvature $|\xi_\kappa x(s)| > 0$ is penalized using a second cost function $\mathcal{C} : \mathcal{R} \to \mathcal{R}_+$, which for the three models is

$$\begin{aligned} &\mathcal{C}^{\mathcal{D}}(\kappa) := \begin{cases} 1, & if\ |\xi_\kappa| \leq 1, \\ +\infty, & otherwise. \end{cases} \\ &\mathcal{C}^{EM}(\kappa) := 1 + |\xi_\kappa|^2, \\ &\mathcal{C}^{RS}(\kappa) := \sqrt{1 + |\xi_\kappa|^2}, \end{aligned} \tag{5}$$

The Dubins cost function penalizes only the length of the path, unless a certain threshold curvature value is exceeded, in which case the path is discarded. The minimum trajectories are obtained for $\gamma = 1$. In Equation (5), $\mathcal{C}^{\mathcal{D}}(\kappa)$ can have only two values, 1 for the circumference arcs and $+\infty$ for straight line segments.

The Euler–Mumford Elastica cost function has the physical meaning of the energy required to bend an elastic rod. In Equation (5), $\mathcal{C}^{\mathcal{EM}}(\kappa)$ has the meaning of unit energy.

The Reeds–Shepp cost function is used to model slow vehicles, especially for wheelchairs. In Equation (5), $\mathcal{C}^{\mathcal{EM}}(\kappa)$ has the meaning of square root of unit energy.

For both the Euler–Mumford Elastica and the Reeds–Shepp cost functions, the curvature changes continuously along the curve.

3. Discussion of Simulation Results

This section addresses the simulation results obtained when applying the four planning methods described before. First, some considerations regarding the execution of the different algorithms are given.

3.1. Simulation Execution Conditions

The following considerations apply for the simulations:

1. The map in Figure 7, corresponding to the city area of the Centre Pompidou, is used for the simulations.
2. A total of 24 different trajectories are executed by each method, covering the main central area of the map.
3. For the execution of the Dubins car, the Euler–Mumford Elastica and the Reeds–Shepp approaches, the codes proposed in [26,27] by Mirebeau et al. are used, which apply the Fast Marching with stencils method (see Section 2.2 and Equations (4) and (5) for a detailed description of the curvature cost functions). The initial and end points of the 24 trajectories are specified, and parameter ξ is taken as 1.
4. The FM^2 method is applied using the same initial and end points defined before, varing parameters α and β as described in the following subsection.
5. For the sake of comparison of the resulting paths and their curvatures using these four methods, two measures of similarity will be used: the Fréchet distance and the area between two curves, as defined before. The results are given and discussed in the following subsections.

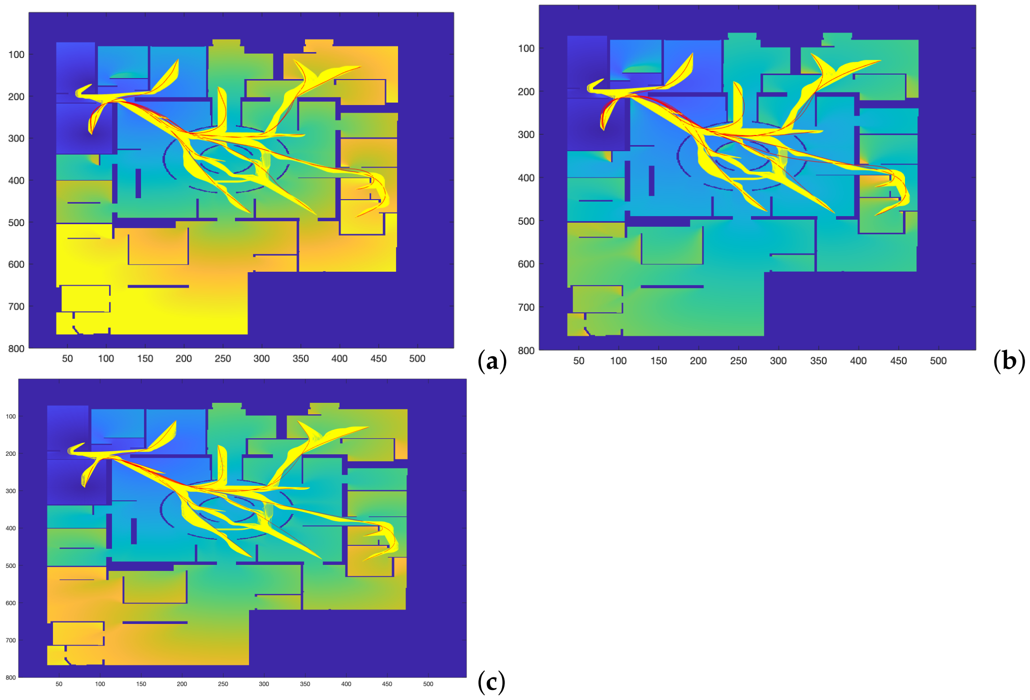


Figure 7. Trajectories obtained by varying α and β parameters to study which set of parameters makes the FM^2 trajectories more similar to those obtained by the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp methods.

3.2. Minimization of the Fréchet Distance

In this section, the comparison of the four planning methods is addressed studying the minimization of the Fréchet distance, as described in the introduction (Figure 1).

The map used for planning is the one represented in Figure 7. This figure shows the comparison of the trajectories obtained by the FM^2 method in yellow, for α = 0.02:0.02:1.0 and β = 0.02:0.02:1.0, with those obtained by the other three methods in red. The computation times involved in the calculation of these trajectories are the ones presented and discussed in Table 1 (significantly faster computation for the FM^2 case).

As can be seen, the trajectories and their curvatures are quite similar at a first sight. The objective is to calculate the values of α and β that better approximate the Dubins, Euler–Mumford Elastica and Reeds–Shepp curves. To do that, it is necessary to have metrics for the distance between two curves. The Fréchet distance can be used for this purpose, as described previously.

The distance matrices between the FM^2 curves and the curves obtained by each of the other three planning methods are shown in Figure 8. Using these matrices, the values of α and β that make this distance minimum can be obtained. As can be seen in the three sub-figures, parameters α and β are coupled and a result similar to the Pareto fronts is obtained.

Figure 9 shows the FM^2 curves corresponding to the best α and β values in yellow and the curves from the other three methods. Figure 10 shows the comparison of the curvatures of the curves corresponding to the best α and β values for the FM^2 method (right) and those obtained by the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp methods (left, from top to bottom, respectively). The curvatures in blue correspond to the point-to-point representations, while the red ones are obtained after applying a smoothing process to the original representations. This treatment is necessary in order to obtain curvatures to be executed by real UAVs. The smoothing method used (DCT-PLS) [28] is based on a penalized least squares approach (PLS), and combines the use of the discrete cosine transform (DCT) and the generalized cross-validation, thus allowing a fast unsupervised smoothing of the data.

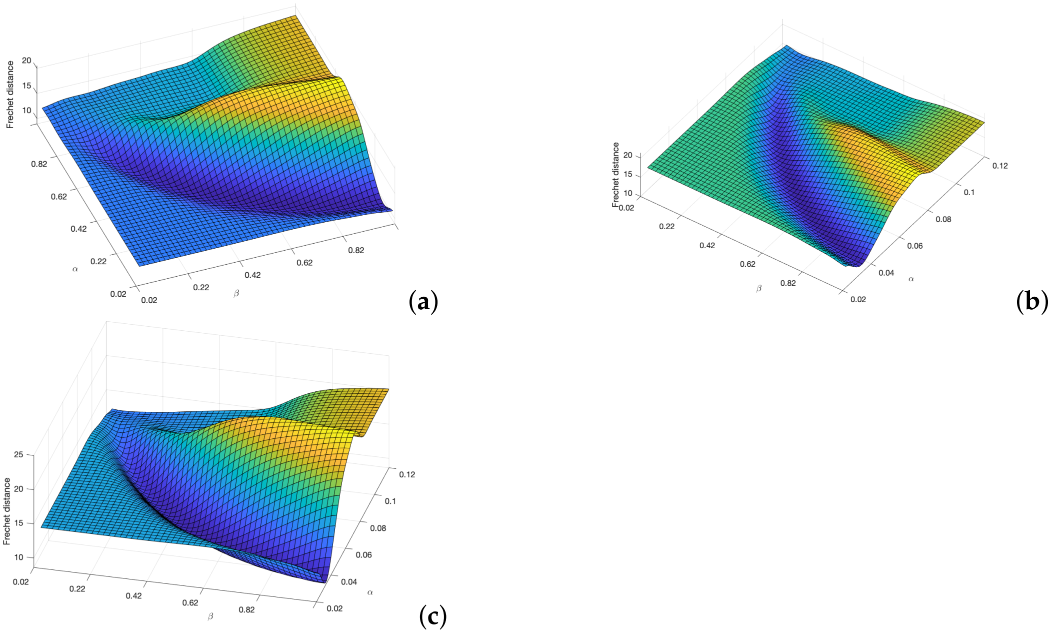


Figure 8. Cost functions (Fréchet distance) obtained by varying α and β parameters to study which set of parameters makes the FM^2 trajectories more similar to those obtain by the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp methods.

Table 2 shows the best values of parameters α and β and the Fréchet cost functions when FM^2 is used to approximate the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp curves.

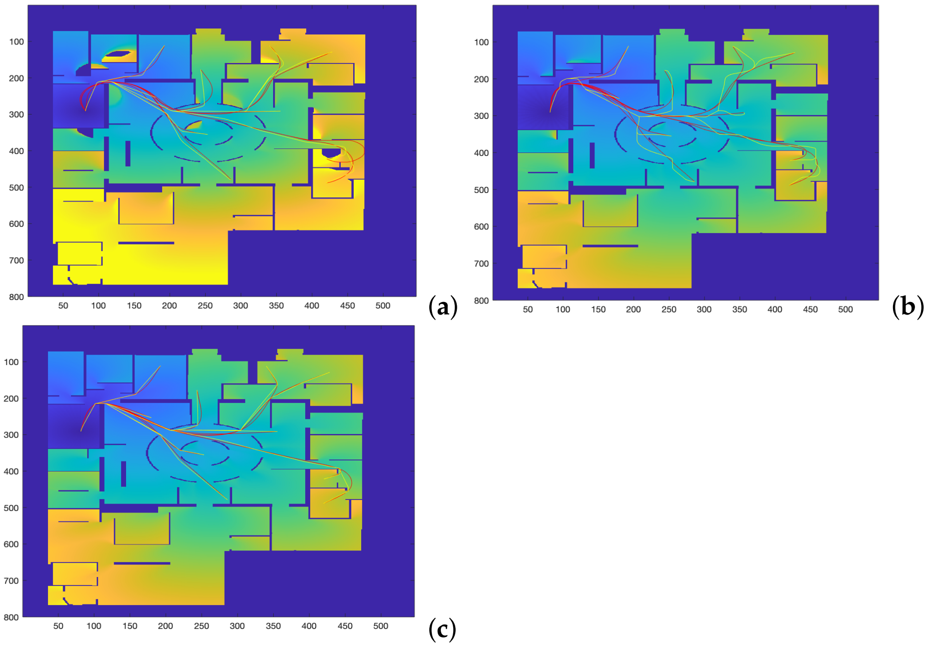


Figure 9. Curves corresponding to the best α and β values for the FM^2 method in yellow in comparison with the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp curves in red. The Fréchet distance between two curves is used as cost function.

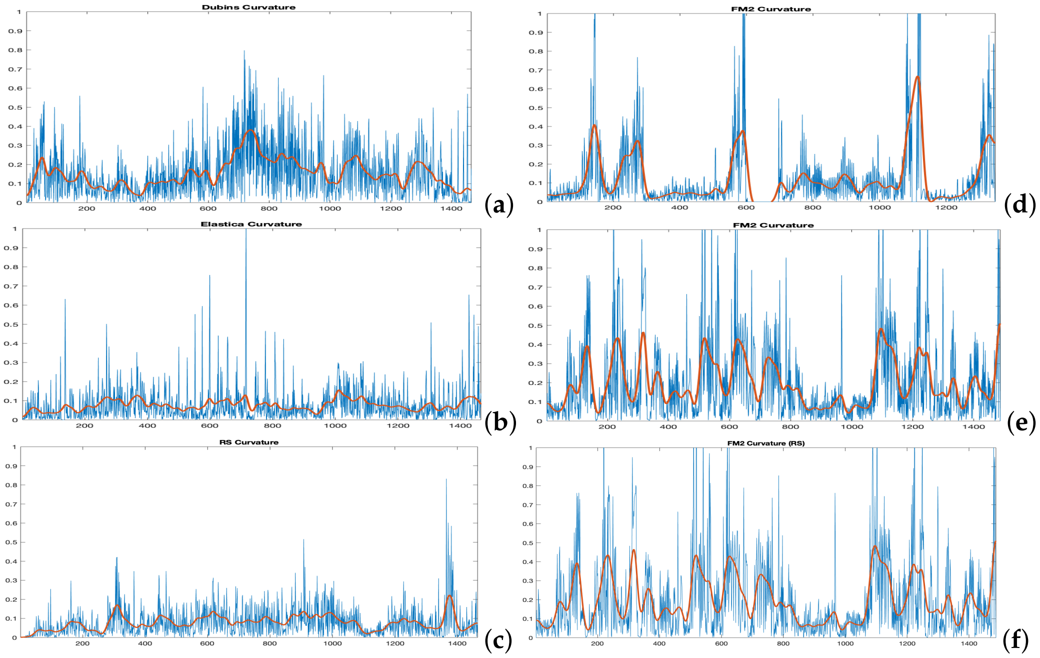


Figure 10. Curvatures corresponding to the (**a**) Dubins, the (**b**) Euler–Mumford Elastica, and the (**c**) Reeds–Shepp curves (left, from top to bottom, respectively) and their corresponding best approximations (**d**–**f**) using FM^2 method (right). The Fréchet cost function is used for comparison. The Fréchet distance between two curves is used as cost function. The curvatures in red are obtained after applying a smoothing process to the raw curvature data.

Table 2. Best values of parameters α and β and the Fréchet cost functions when FM^2 is used to approximate the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp curves.

	Dubins	Euler–Mumford Elastica	Reeds–Shepp
α	0.1	0.3	0.038
β	0.98	0.96	0.92
Fréchet cost function	8.1	11.1	9.2

3.3. Minimization of the Area between Curves

The area between two curves can also be used as a criterion to measure the distance between them, as shown in the example of Figure 11.

As the curves are not parameterized in X or Y and can take increasing or decreasing values in different sections, they are difficult to calculated by methods based on integration. Since the curves are numerically given by matrices with two rows (X and Y coordinates) and many columns (the number of points), they can be considered polygons. As the two discrete curves have the same first point and the same last point, if the order of the second curve is reversed and they are concatenated, a closed polygon with many sides is obtained whose area can be calculated by triangulation.

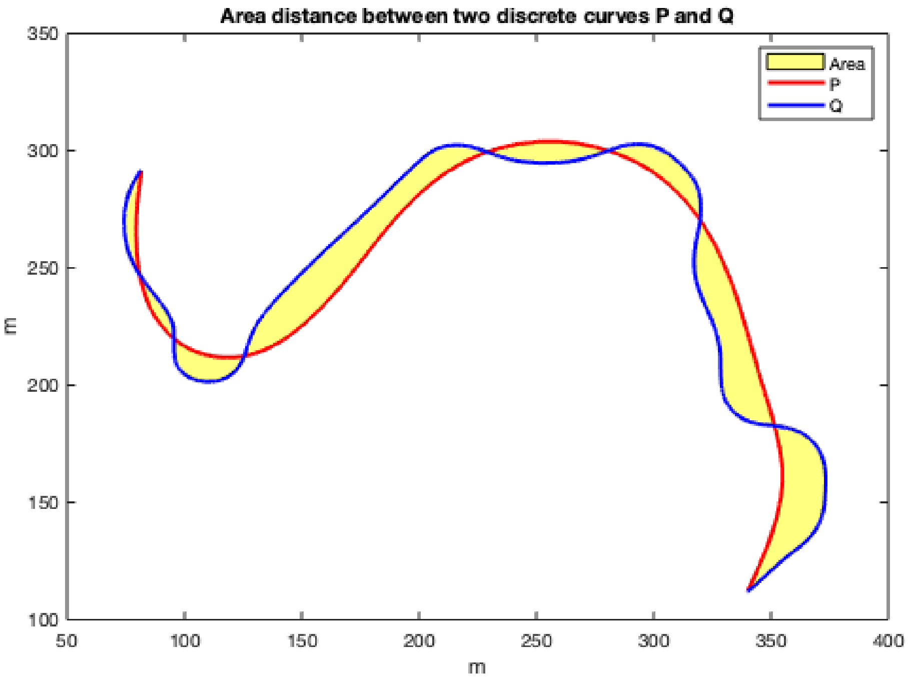


Figure 11. Area between two curves as a metric to measure the distance between the curves.

Applying this area concept, now the distance matrices between the FM^2 curves and the curves obtained by each of the other three planning methods are shown in Figure 12. Using these matrices, the value of α and β that make this distance minimum can be obtained. As can be seen in the three sub-figures, parameters α and β are coupled and a result similar to the Pareto fronts is obtained.

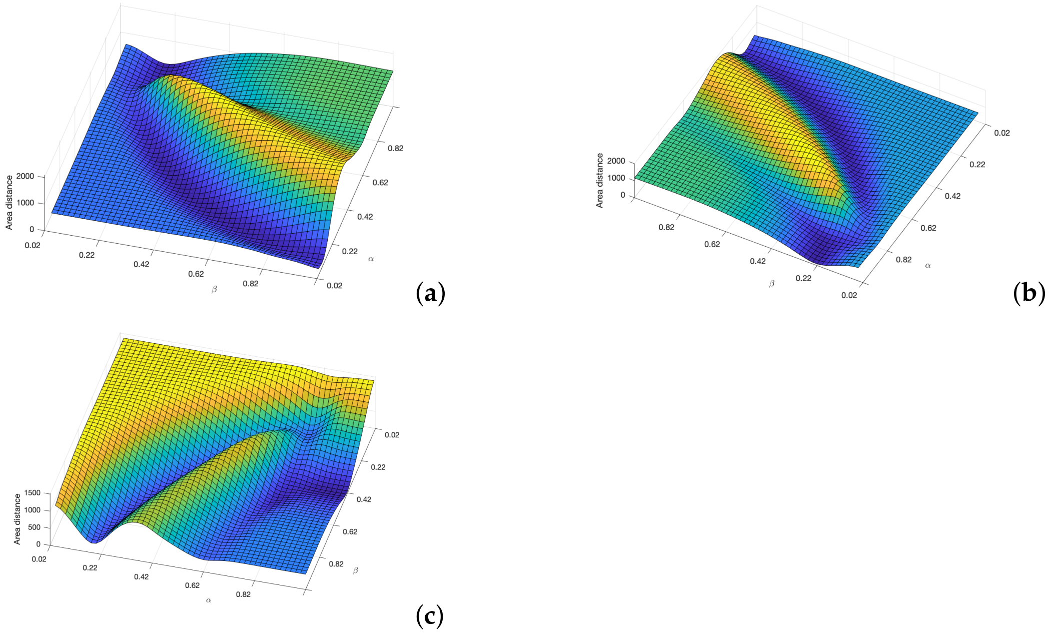


Figure 12. Cost functions (area between two curves) obtained by varying α and β parameters to study which set of parameters makes the FM^2 trajectories more similar to those obtain by the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp methods.

Figure 13 shows the curves corresponding to the best α and β values for the FM^2 method in yellow in comparison with the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp curves in red.

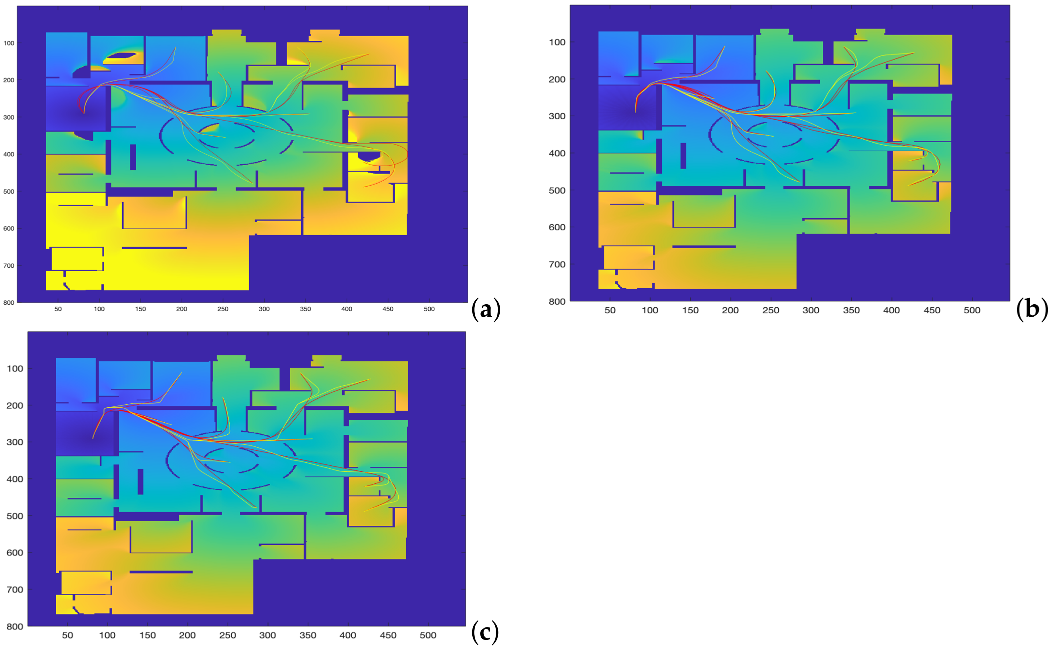


Figure 13. Curves corresponding to the best α and β values for the FM^2 method in yellow in comparison with the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp curves in red. The area between the two curves is used as cost function.

Figure 14 shows the comparison of the curvatures of the curves corresponding to the best α and β values for the FM^2 method (right) and those obtained by the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp methods (left, from top to bottom, respectively). The curvatures in blue correspond to the point-to-point representations, while the red ones are obtained after applying a smoothing process to the original ones, as discussed for the case of the Fréchet distance.

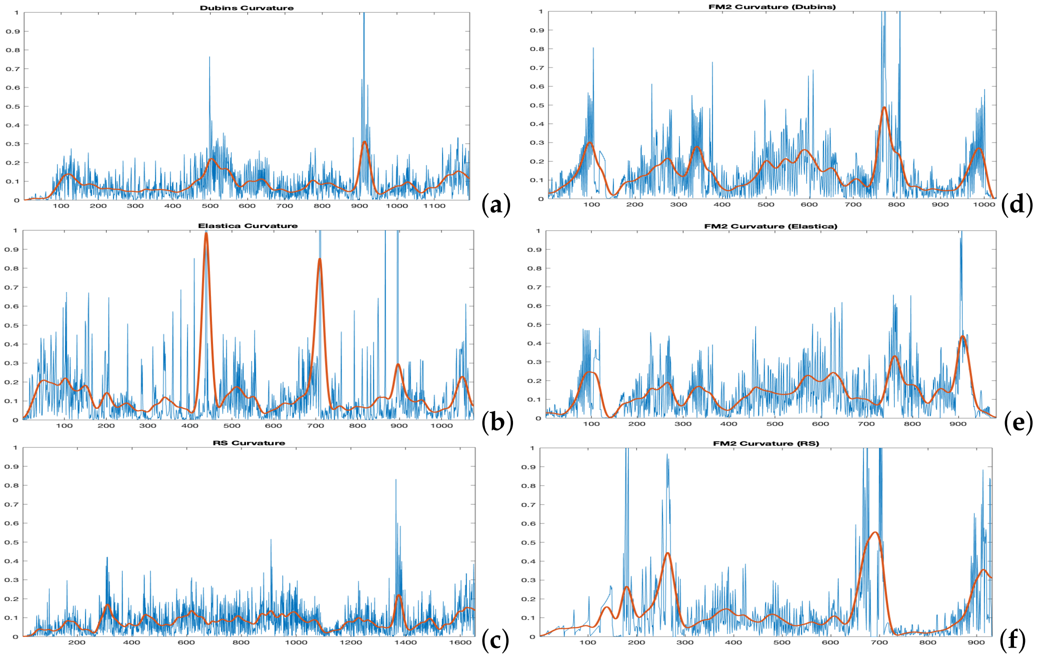


Figure 14. Curvatures corresponding to the (**a**) Dubins, (**b**) Euler–Mumford Elastica, and (**c**) Reeds–Shepp curves (left, from top to bottom, respectively) and their corresponding best approximations (**d**–**f**) using FM^2 method (right). The area between the curves is used as cost function for comparison. The curvatures in blue correspond to the point to point representations. The curvatures in red are obtained after applying a smoothing process.

Finally, Table 3 shows the best values of parameters α and β and the area cost functions when FM^2 is used to approximate the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp curves. It can be seen that taking the area between two curves as the cost funtion for the minimization provides better results in comparison to the Fréchet distance.

Table 3. Best values of parameters α and β and the area cost functions when FM^2 is used to approximate the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp curves.

	Dubins	**Euler–Mumford Elastica**	**Reeds–Shepp**
α	0.90	0.84	0.40
β	0.18	0.20	0.50
Area cost function	4.50	2.19	0.3352

3.4. Distance from Paths to Obstacles

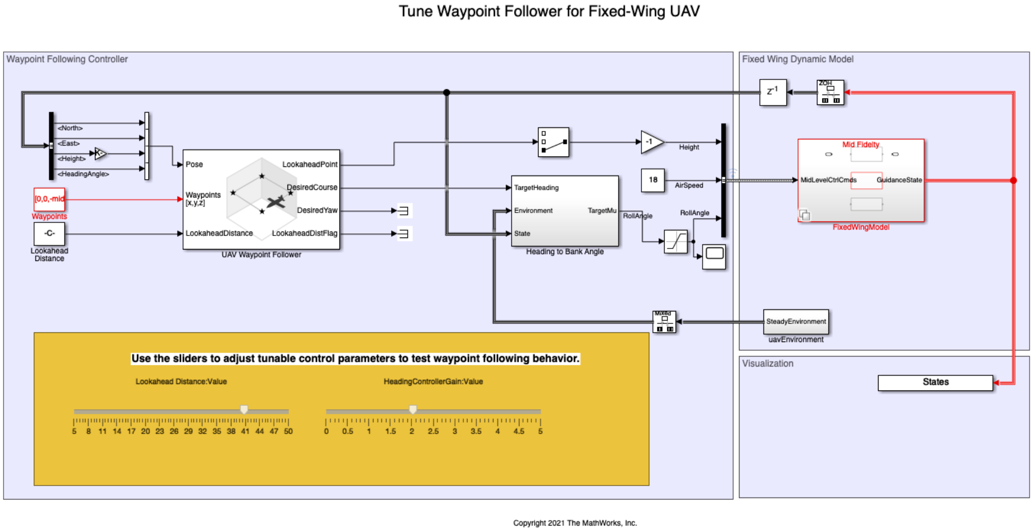
In addition to the curvature of the curves, another important measure of quality of the trajectories is how close they are to obstacles, that is, the minimum distance to obstacles. Table 4 shows the distances to the obstacles of the trajectories obtained by the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp methods in comparison with the distances obtained using their best FM^2 approximations. The area cost function is used for the comparison. It can be seen that the FM^2 method provides safer curves since their distances to obstacles is significantly bigger than the others, specially for the Dubins and Reeds–Shepp cases.

Table 4. Distances to the obstacles of the trajectories obtained by the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp methods in comparison with the distances obtained using their best FM^2 approximations. The area cost function is used for comparison.

	Dubins	**Euler–Mumford Elastica**	**Reeds–Shepp**
α	0.90	0.84	0.40
β	0.18	0.20	0.50
Original distance	0.0093	0.0169	0.0071
Distance of the FM^2 approximation	0.0276	0.0191	0.0214

3.5. Simulation Results Using a Quadcopter Model and a Fixed-Wing UAV

So far the study has focused on the quality of the trajectories based on their curvatures and their distances to the obstacles. However, it is also important to consider the vehicles dynamics and check if they are able to execute the paths, specially for the case of fixed-wing UAVs. In order to do this analysis, a vehicle dynamic model has been considered using the Matlab Simulink: file shared_uav_aeroblks/UAVFidelityExample [29] to simulate the tracking of the trajectories, as shown in Figure 15.



Tune Waypoint Follower for Fixed-Wing UAV

Use the sliders to adjust tunable control parameters to test waypoint following behavior.

Lookahead Distance:Value

HeadingControllerGain:Value

Copyright 2021 The MathWorks, Inc.

Figure 15. Matlab Simulink model to simulate the influence of the vehicle dynamics when executing the paths.

In the Simulink model, the UAV Waypoint Follower block stands out, which considers the dynamics of the UAV and allows the selection of a fixed-wing drone and a quadcopter. This block is based on a Pure Pursuit algorithm that follows a point on the desired trajectory. There are two parameters to select: the lookahead distance and the heading controller gain. The results are shown in Figure 16 for a lookahead distance of 40 m and a heading controller gain of 2, considering the FM^2 path that better approximate to the Dubins path (as calculated in the previous section). The figure shows that the difference between the reference path (FM^2 trajectory) and the executed path (UAV trajectory) is negligible enough so as to consider that the path is feasible to be used in practice.

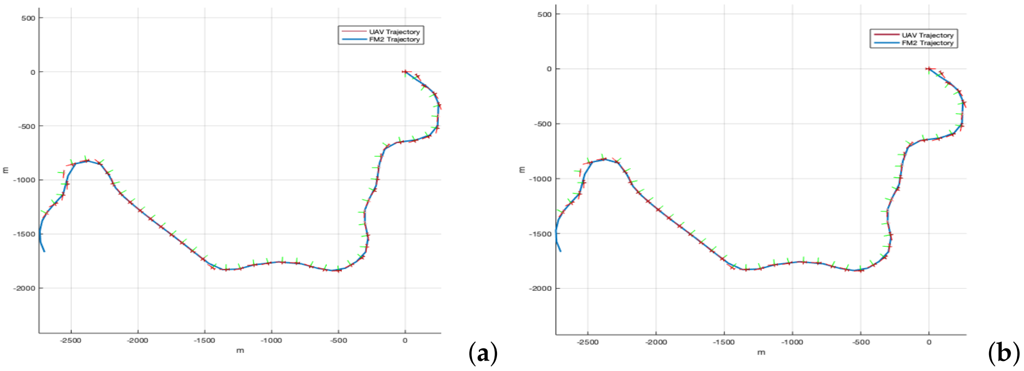


Figure 16. Original paths (approximation of FM^2 to Dubins method) versus executed paths considering (**a**) a fixed-wing UAV and (**b**) a quadcopter.

4. Conclusions

From the results presented in this work, some important conclusions can be reached. First of all, the FM^2 method has demonstrated to provide paths whose smoothness, curvature and distance to the obstacles can be easily adjusted thanks to the use of parameters α and β. Through the minimization of functions such as the Fréchet distance between two curves or the area between them, the tuning of these two parameters can be performed so that the resulting paths are competitive in comparison to those provided by other planning approaches such as the Dubins, the Euler–Mumford Elastica, and the Reeds–Shepp methods.

The results show that selecting the area between two curves as the cost function for the minimization provides better results than the Fréchet distance when it comes to approximating the FM^2 trajectories to those of the other three methods. In any case, other cost functions could be studied, and the alternative of selecting the curvature instead of the curve itself for the minimization could be of interest. These issues will be addressed in future works.

On the other hand, simulation tests have shown that the resulting paths are feasible and can be executed by both quadcopters and fixed-wing UAVs (more demanding dynamics), which allows the FM^2 method to be used for real drones applications.

Moreover, the comparison study has shown that the FM^2 method significantly overperforms the others in respect to computation time, computing the trajectories around 220 times faster. The nature of the algorithm and the fact that the vehicle's kinematic constraints do not need to be introduced in the computation process allow this fact. It also opens the possibility to use this method in real time for dynamic path-planning applications where the environment is changing during the planning activity (moving obstacles or other UAVs flying in the map).

Author Contributions: Investigation, S.G. and B.L.; Methodology, S.G.; Software, J.M. and F.Q.; Supervision, L.M.; Writing—original draft, S.G.; Writing—review & editing, C.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the EUROPEAN COMMISSION: Innovation and Networks Executive Agency (INEA), grant number 861696-LABYRINTH.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LaValle, S. *Planning Algorithms Cambridge*; Cambridge University Press: Cambridge, UK, 2006.
2. Dijkstra, E. A Note on Two Problems in Connection with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]

3. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
4. Kavraki, L.; Svestka, P.; Latombe, J.; Overmars, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
5. Yan, Z.; Zhao, Y.; Chen, T.; Deng, C. 3D path planning for AUV based on circle searching. In Proceedings of the OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean, Hampton Roads, VA, USA, 14–19 October 2012 .
6. Pivtoraiko, M.; Kelly, A. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, Canada, 2–6 August 2005; pp. 594–600.
7. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Practical search techniques in path planning for autonomous driving. *Int. Symp. Comb. Search SoCS* **2008**, *1001*, 18–80.
8. LaValle, S.; Kuffner, J. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [CrossRef]
9. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
10. Hernández, J.; Vidal, E.; Moll, M.; Palomeras, N.; Carreras, M.; Kavraki, L. Online motion planning for unexplored underwater environments using autonomous underwater vehicles. *J. Field Robot.* **2019**, *36*, 370–396. [CrossRef]
11. Pairet, E.; Hernández, J.D.; Lahijanian, M.; Carreras, M. Uncertainty-based Online Mapping and Motion Planning for Marine Robotics Guidance. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 2367–2374.
12. Gundlach I, Konigorski U, Hoedt J. Zeitoptimale Trajektorienplanung für automatisiertesFahren im fahrdynamischen Grenzbereich. *VDI Ber.* **2017**, *2292*, 223–234.
13. Perantoni G.; Limebeer DJ. Optimal control for a formula one car with variable parameters. *Veh. Syst. Dyn.* **2014**, *52*, 653–678. [CrossRef]
14. Heilmeier, A.; Wischnewski, A.; Hermansdorfer, L.; Betz, J.; Lienkamp, M.; Lohmann, B. Minimum curvature trajectory planning and control for an autonomous race car. *J. Veh. Syst. Dynam.* **2020**, *58*, 1497–1527. [CrossRef]
15. López, B. Path Planning and Collision Risk Management Strategy for Multi-UAV Systems in 3D Environments. *Sensors* **2021**, *21*, 4414. https://10.3390/s21134414. [CrossRef] [PubMed]
16. Levien, R. The Elastica: A Mathematical History. In *Technical Report UCB/EECS-2008-103*; EECS Department, University of California: Berkeley, CA, USA, 2008.
17. Oldfather, W.A.; Ellis, C.A.; Brown, D.M. Leonhard Euler's Elastic Curves. *Isis* **1933**, *20*, 72–160. [CrossRef]
18. Reeds, J.; Shepp, L. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393 . [CrossRef]
19. Alt, H.; Godau, M. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* **1995**, *5*, 75–91. [CrossRef]
20. Sethian, J.A. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA* **1996**, *93*, 1591–1595. https://10.1073/pnas.93.4.1591. [CrossRef] [PubMed]
21. Sethian, J. Level Set Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. In *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press: Cambridge, UK, 1996.
22. Gómez, J.V.; Álvarez, D.; Garrido, S.; Moreno, L. Fast Methods for Eikonal Equations: An Experimental Survey. *IEEE Access* **2019**, *7*, 39005–39029. https://10.1109/ACCESS.2019.2906782. [CrossRef]
23. Garrido, S.; Moreno, L.; Blanco, D. Voronoi Diagram and Fast Marching applied to Path Planning. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 3049–3054.
24. Garrido, S.; Moreno, L.; Abderrahim, M.; Blanco, D. FM2: A real-time sensor-based feedback controller for mobile robots. *Int. J. Robot. Autom.* **2009**, *24*, 48–65. https://10.2316/Journal.206.2009.1.206-3169.
25. Valero-Gomez, A.; Gomez, J.V.; Garrido, S.; Moreno, L. The Path to Efficiency: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories. *IEEE Robot. Autom. Mag.* **2013**, *20*, 111–120. https://10.1109/MRA.2013.2248309. [CrossRef]
26. Mirebeau, J.M.; Portegies, J. Hamiltonian Fast Marching: A Numerical Solver for Anisotropic and Non-Holonomic Eikonal PDEs. *Image Process. Line* **2019**, *9*, 47–93. [CrossRef]
27. Mirebeau, J.M. Fast Marching methods for Curvature Penalized Shortest Paths. *J. Math. Imag. Vision.* **2018**, *60*, 784–815. [CrossRef]
28. García, D. A fast all-in-one method for automated post-processing of PIV data. *Exp. Fluids* **2011**, *50*, 1247–1259. [CrossRef] [PubMed]
29. Matlab. Transition from Low to High-Fidelity UAV Models in Three Stages. 2022. Available online: https://es.mathworks.com/help/uav/ug/transition-from-low-to-high-fidelity-uav-models.html (accessed on 1 January 2022).