

Article

Smooth 3D Path Planning by Means of Multiobjective Optimization for Fixed-Wing UAVs

Franklin Samaniego *, Javier Sanchis, Sergio Garcia-Nieto and Raul Simarro

Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, 46022 Valencia, Spain; jsanchis@isa.upv.es (J.S.); sgnieto@isa.upv.es (S.G.N.); rausifer@isa.upv.es (R.S.)

* Correspondence: frasarie@doctor.upv.es; Tel.: +593-99-130-5550

Received: 28 October 2019; Accepted: 25 December 2019; Published: 28 December 2019



Abstract: Demand for 3D planning and guidance algorithms is increasing due, in part, to the increase in unmanned vehicle-based applications. Traditionally, two-dimensional (2D) trajectory planning algorithms address the problem by using the approach of maintaining a constant altitude. Addressing the problem of path planning in a three-dimensional (3D) space implies more complex scenarios where maintaining altitude is not a valid approach. The work presented here implements an architecture for the generation of 3D flight paths for fixed-wing unmanned aerial vehicles (UAVs). The aim is to determine the feasible flight path by minimizing the turning effort, starting from a set of control points in 3D space, including the initial and final point. The trajectory generated takes into account the rotation and elevation constraints of the UAV. From the defined control points and the movement constraints of the UAV, a path is generated that combines the union of the control points by means of a set of rectilinear segments and spherical curves. However, this design methodology means that the problem does not have a single solution; in other words, there are infinite solutions for the generation of the final path. For this reason, a multiobjective optimization problem (MOP) is proposed with the aim of independently maximizing each of the turning radii of the path. Finally, to produce a complete results visualization of the MOP and the final 3D trajectory, the architecture was implemented in a simulation with Matlab/Simulink/flightGear.

Keywords: UAV; path planning; smooth path planning; multiobjective optimization

1. Introduction

The latest technological and scientific advances in the field of mobile robotics have enabled the area of autonomous vehicles (AV) to become a reality applied to the civil and military sectors [1,2]. Consequently, this technological branch presents a constant and vertiginous development in different fields, among them the development of new navigation and guidance techniques. The constant evolution in this field responds to new challenges in real applications [3–5].

Autonomous vehicles with non-holonomic characteristics [6–8] can be technologically adapted to different environments, and so, produce unmanned ground vehicles (UGVs) [9–11], unmanned underwater vehicles (UUVs) [12–14], and unmanned aerial vehicles (UAVs) [15–18]. Obviously, each of the above categories presents its own scientific and technological challenges, including path planning, navigation, and guidance.

It is important to note that the most common problem when determining a possible feasible path is the consideration of the AV intrinsic constraints. Therefore, the non-inclusion of kinematic and/or dynamic constraints of the AV when addressing the path planning problem may lead to non-feasible solutions that, for example, make it impossible for the AV to satisfactorily follow a path. However, including in the design all the constraints of the AV in the calculation phase of the path

planning can lead to very complex optimization problems without a single solution and with very high computational costs.

This paper focuses on the generation of smooth paths for fixed wing unmanned aerial vehicles (UAVs) that have high flight capabilities and extended flight-time missions, even in low power propulsion situations [19]. For these reasons, fixed-wing UAVs are suitable for use in terrain mapping applications for later action by the security forces, and in search and rescue tasks, both for the detection of people and the provision of first aid.

Due to the non-holonomic constraints of fixed-wing UAVs, the aim is to create a smooth three-dimensional curve from an initial point to a goal point through a complex 3D space with or without obstacles. To achieve this goal, it is essential to define a feasible path that minimizes flight turning effort and distance traveled. The ordered set of waypoints that will be used to generate the path to follow is defined as the control points set. In general, the problem of path planning is defined in a space region denoted as χ and split as the tripla $(\chi_{free}, \chi_{init}, \chi_{goal})$. The movement space is defined as $(\mathbb{W} = \mathbb{R}^N : N = \{2, 3\})$, where the obstacle region is denoted by χ_{obs} , so that χ/χ_{obs} is an open set denoting the collision-free space χ_{free} . The initial condition χ_{init} and the final condition χ_{goal} are elements of χ_{free} .

The set of control points that define the collision-free space is calculated using specific path planning methods based on continuous and discrete environment sampling. Some examples of these techniques are: the rapidly-exploring random tree (RRT) [20–23]; probabilistic road maps (PRM) [24–28]; heuristic planners (genetic algorithms—GA) [29,30]; swarm intelligence [31–34]; fuzzy logic [35,36]; Voronoi diagrams [37–39]; artificial potential [40–43]; and recursive rewarding modified adaptive cell decomposition (RR-MACD) [44].

All the techniques mentioned build piece-wise paths in 2D or 3D to address the standard problem of path planning. These methods may provide optimal or near-optimal paths; however, they cannot guarantee smoothness and continuity, which could make it difficult to guide the UAV through the paths generated. Moreover, these techniques do not directly incorporate the operational constraints of the UAV and the environment. Therefore, this paper proposes a methodology to define feasible and smooth UAV paths, including system operational kinematic constraints.

The ability of an UAV to fly from one position to another and the consequent definition of the mission to be performed remains a challenge that requires the application of increasingly sophisticated strategies. One of the fundamental constraints to be considered in mission planning is the ability to be positionable and sensorially oriented (on the UAV or the environment) throughout the duration of the mission. This sensory location enables the construction of maps of the environment, and also enables the UAV to estimate its own current position and complete its self-location on the map. Similarly, it is important to note that good positioning and sensory orientation can be achieved by making a path plan and tracking it across or beyond the sensory detection domain. It is important to mention that the definition of smooth paths is not a new subject; various approaches have been proposed for non-holonomic UAVs, such as Dubins [45,46] in which paths are defined by connecting lines and arc-circular segments. The disadvantage is the generation of discontinuities in the connection points between segments. Another useful methodology in the literature is Clothoid curves [47–49], the main advantage being increases in curvature as a function of the arc-length; meanwhile the disadvantage lies in the limitation of length. These approximate curves generate continuities [50] of the type C^1 (a continuous path C^1 preserves the continuity of the tangent vector, in addition to maintaining continuous speed) and have been used in applications on mobile vehicles and on UAVs with flight limitation at a constant altitude, and examples are mentioned in [51–55]. An intuitive approach that ensures C^2 continuity (a continuous trajectory C^2 preserves the second order differential values at each point of the trajectory, in addition to maintaining the continuity of the acceleration vector) in 3D planning focuses on curvature and torsion zero at the junction points, as proposed in [56]. This is an interesting proposal where the 3D curve is built from a 2D curve in a (x, y) plane; the resulting length of this curve is the main parameter to build the curve in the other (x, z) plane. Finally, the Bézier,

B-spline [57,58] are easy to implement polynomial curves; however, none are suitable for planning since they are sensitive to control parameters and weights [59], and do not take into account the constraints of the vehicles on which they are applied, so those curves need to be optimized. Finally, these types of parametric curves have no physical meaning and the relationship between design parameters and system variables is not defined. The above-mentioned approximation methodologies generate two different phenomena called interpolation (generation of a curve which must pass through the control points) and approximation (generation of a curve that approximates the control points, but may only go through the start and finish rather than all of them) detailed in [60,61]. The work presented here explores both phenomena in-depth to respond to the definition of feasible trajectories.

The starting point of this work is the set of 3D collision-free points generated by the various planners that guarantee that the path departs from the init point and ends at the goal point, and avoids static and dynamic obstacles. From these collision-free points, an ordered set of straight lines is built that define a first path that will later be smoothed to incorporate the feasibility and constraints of the UAV. The UAV constraints in this work are focused on its ability to turn horizontally and vertically. Therefore, for a UAV to complete a sequence of turns at a defined speed, it must determine its minimum turning radius R_p . If the turning radius is too small, the UAV will lose the trajectory; however, if the turning radius grows, the UAV can perform maneuvers with less effort.

The aim is to maintain 3D planning results, and at the same time, generate a finite set of possible 3D curves that optimize an approximate 3D curve, and simultaneously, the turns of the UAV—which is raised as a multiobjective optimization problem (MOP) [62]. This approach will result in a set of paths that meet UAV constraints expressed as dominant solutions on a Pareto n dimensional front [63]. Finally, selection criteria must be applied to determine the desired response from the point of view of curvature κ and torsion τ of the generated 3D curve. Thus, in order to verify the functionality of the proposed methodology, the results of the curves generated after the 3D curve optimization were compared with a known Bézier type approximation methodology [64].

This document is structured as follows. A brief summary of MOP concepts is given in Section 2.1. In Section 2.2, a brief description of smooth curves is given. Section 3 presents the formulation of the problem. Section 4 details the complete methodology for solving the problem. Section 5 details the experiments and results of 3D smooth path planning. Finally, conclusions and future work are presented in Section 6.

2. Background

2.1. MultiObjective Optimization

The optimization problem (OP) attempts to determine a solution that represents the optimal value (minimum or maximum) of a function, such as $f : X \rightarrow \mathbb{R}$, where X is a feasible decision vector, being $\min(f(x)) : x \in X$. However, for problems where simultaneous optimization of more than one objective is necessary, i.e., multiobjective optimization (MOP), the function is shaped $f : X \rightarrow \mathbb{R}^k$, where $k \geq 2$ is the number of objectives. Therefore, the value vector of the target function could be defined as $f : X \rightarrow \mathbb{R}^k, f(x) = (f_1(x), \dots, f_k(x))^T$. However, there is not usually a single X that generates an optimum that simultaneously satisfies each of the k objectives, due to the conflict between the objectives. The aim is to find a situation in which all objectives are satisfactorily within acceptable parameters. The MOP solution leads to points where any improvement in one target results in the degradation of any other target (one or more). Thus, these points are represented as a Pareto front [63], where all the points of the front are equally optimal.

Therefore, as expressed in [62] the MOP can be established as

$$\min \mathbf{J}(\theta) = \min_{\theta \in D} [J_1(\theta), J_2(\theta), \dots, J_m(\theta)] \quad (1)$$

subject to:

$$\begin{aligned} g(\theta) &\leq 0 \\ h(\theta) &= 0 \\ \theta_{il} &\leq \theta_i \leq \theta_{iu}, i = [1, \dots, n], \end{aligned} \tag{2}$$

where $\theta \in \mathbb{R}^n$ is the decision vector, D is the decision space; $\mathbf{J}(\theta) \in \mathbb{R}^m$ is the target vector; $g(\theta)$ and $h(\theta)$ are constraint vectors; and finally, θ_{il} is the upper boundary and θ_{iu} is the lower boundary of the decision space. Consequently, there is no single optimal model; in fact, there is a set of optimal solutions with different trade-offs between objectives, where none is better than the others. Using the definition of dominance, the Pareto set Θ_P is the set of each non-dominated solution.

In this way, the Pareto domination is defined in case a solution θ^1 dominates another solution θ^2 ; that is, $(\theta^1 \prec \theta^2)$, if

$$\forall i \in B, J_i(\theta^1) \leq J_i(\theta^2) \wedge \exists k \in B : J_k(\theta^1) < J_k(\theta^2), \tag{3}$$

where $J_i(\theta), i \in B := [1 \dots m]$ are the objectives to be optimized. Therefore, the optimal set of Pareto Θ_P is given by

$$\begin{aligned} \Theta_P &= \theta \in D | \nexists \tilde{\theta} \in D : \tilde{\theta} \prec \theta \\ J(\Theta_P) &= \{J(\theta) | \theta \in \Theta_P\}, \end{aligned} \tag{4}$$

where Θ_P and $J(\Theta_P)$ are MOP solutions. However, in most cases they are unreachable because Θ_P normally includes infinite solutions. Therefore, a finite set of Θ_P^* from Θ_P and another finite set of $J(\Theta_P^*)$ from $J(\Theta_P)$ represent satisfactory solutions. Starting from $J(\Theta_P^*)$, the decision maker selects a solution according to the established preferences. For example, a certain point in the Pareto front that is close to the ideal point (called utopia point) \mathbf{J}^{ideal} .

$$\mathbf{J}^{ideal} = \{J_1 \min(\theta), \dots, J_m \min(\theta)\}. \tag{5}$$

Hence, an appropriate methodology for characterizing MOP is known as the elitist multiobjective evolutionary algorithm (ϵ^λ -MOGA) [62], which makes a distributed approach to Pareto's front. The aim of ϵ^λ -MOGA is to find an intelligent distributed convergence towards a set of ϵ -Pareto; i.e., determine $\theta_{P\epsilon}^*$ along the Pareto front $\mathbf{J}(\Theta_P)$. The target space is split into a fixed number of boxes. Therefore, for each dimension $i \in B$, cells n_box_i wide are created ϵ_i , where

$$\begin{aligned} \epsilon_i &= (J_i^{max} - J_i^{min}) / n_box_i \\ J_i^{max} &= \max_{\theta \in \Theta_{P\epsilon}^*} J_i(\theta), J_i^{min} = \min_{\theta \in \Theta_{P\epsilon}^*} J_i(\theta). \end{aligned} \tag{6}$$

Each box can be occupied by a single solution; therefore, this grid produces an intelligent distribution and preserves the diversity of $\mathbf{J}(\Theta_{P\epsilon}^*)$. In addition, it is important to note that only the occupied boxes are verified, avoiding the need to use other clustering techniques to obtain adequate distributions. On the other hand, for a solution $\theta \in D$, $box_i(\theta)$ is defined as

$$box_i(\theta) = \left[\frac{J_i(\theta) - J_i^{min}}{J_i^{max} - J_i^{min}} \cdot n_box_i \right] \forall i \in B \tag{7}$$

Then, $\mathbf{box}(\theta) = \{box_1(\theta), \dots, box_m(\theta)\}$. A solution θ^1 with value $\mathbf{J}(\theta^1) \in \epsilon$ dominates the solution θ^2 with value $J(\theta^2)$, denoted by $\theta^1 \prec_\epsilon \theta^2$, only if

$$\mathbf{box}(\theta^1) \prec \mathbf{box}(\theta^2) \vee (\mathbf{box}(\theta^1) = \mathbf{box}(\theta^2) \text{ and } \theta^1 \prec \theta^2). \tag{8}$$

So, a set $\Theta_{P\epsilon}^* \subseteq \Theta_P$ is ϵ -stop only if

$$\forall \theta^1, \theta^2 \in \Theta_{P\epsilon}^*, \theta^1 \neq \theta^2, \mathbf{box}(\theta^1) \neq \mathbf{box}(\theta^2) \text{ and } \mathbf{box}(\theta^1) \not\prec_{\epsilon} \mathbf{box}(\theta^2). \tag{9}$$

Thus, a $\Theta_{P\epsilon}^*$ is reached with the greatest possible number of solutions that adequately characterize Pareto's front and whose number of possible solutions depend on n_box_i , and will not exceed the following level.

$$|\Theta_{P\epsilon}^*| \leq \frac{\prod_{i=1}^n n_box_i + 1}{n_box_{max} + 1}, n_box_{max} = \max_i n_box_i. \tag{10}$$

Hence, it is possible to control the maximum number of solutions to characterize the Pareto front. Finally, due to the definition of the box, the anchor points $J_i(\theta^{i*})$ are assigned a value of $box_i(\theta^{i*}) = 0$, whereby $J_i(\theta^{i*}) = J_i^{min}$. Therefore, no solution θ can ϵ -dominate them because, by applying the definition of the box, their $box_i(\theta) \geq 1$.

The above process delivers two defined sets of responses: (a) the Pareto front Θ_P^* deploys a finite set of minimum values as the optimal path response within the search space; (b) the corresponding optimal points $J(\Theta_P^*)$.

2.2. 3D Curves for UAVs

A non-holonomic UAV [65] can perform flights in 3D Euclidean space. Nevertheless, to complete each movement sequence (horizontal and vertical), a set of UAV flight constraints must overcome. An UAV attempts to perform 3D movements at a defined velocity, meaning that the UAV is moving continuously, attempting to maintain that velocity. However, an UAV has a maximum capacity of turn and elevation at a defined velocity. Therefore, the aim is to build a 3D smooth curve inside the UAV flight turning boundaries, in such a way to reach a complete 3D smooth curve tracking.

A smooth curve can be defined as the representation of a continuous function, which can be expressed as $\mathbf{C} : I \rightarrow X$, where I is the curve interval composed of real numbers, while X represents the topological space. If the topological space is three-dimensional $X = \mathbb{R}^3$, then $\mathbf{C} : [a, b] \rightarrow \mathbb{R}^3$, is a differentiable injectable and continuous function, whose arc-length s is independent of the parametrization \mathbf{C} . If a UAV is considered as a particle that travels along the envelope of the \mathbf{C} curve at a defined speed v , this particle suffers changes in its local coordinate system due to the set of rotations of the \mathbf{C} curve. Therefore, \mathbf{C} must not allow rotation changes outside of the intrinsic rotation capabilities of the UAV.

It is important to mention that the formulation known as Frenet–Serret [66,67] describes the kinematic properties of particles that move along three-dimensional Euclidean space \mathbb{R}^3 continuous and differentiable $\mathbf{C}(s)$, and parametrized by its arc-length s (the arc-length is an invariant Euclidean characteristic of the curve). However, if we assume a curve given by a series of points along the Euclidean space as $r(t)$, where the parameter t does not need the arc-length, then the tangent (T), normal (N) and bi-normal (B) derived vectors can be described, as all are mutually perpendicular (orthogonal base). Therefore, according to the theory of differential geometry of curves [68], the following equals can be defined as:

$$\begin{aligned} T(t) &= N(t) \times B(t) = \frac{r'(t)}{\|r'(t)\|} \\ B(t) &= T(t) \times N(t) = \frac{r'(t) \times r''(t)}{\|r'(t) \times r''(t)\|} \\ N(t) &= B(t) \times T(t) = \frac{[r'(t) \times r''(t)] \times r'(t)}{\|[r'(t) \times r''(t)] \times r'(t)\|} \end{aligned} \tag{11}$$

where $r'(t) = \frac{dr(t)}{dt}$, $r''(t) = \frac{d^2r(t)}{dt^2}$ and $r'''(t) = \frac{d^3r(t)}{dt^3}$ are the position vector derivatives $r(t)$. These three vectors configure a navigation reference system of the UAV. Similarly, it is important to mention that

the tangent vector $T(t)$ is parallel to velocity, while the normal vector $N(t)$ is represented by the change of direction per time unit of velocity.

The curvature terms κ (change of direction of the vector tangent $T(t)$ to the curve $r(t)$) and torsion τ (change of direction of the vector bi-normal $B(t)$) are defined as:

$$\kappa = \frac{\|r'(t) \times r''(t)\|}{\|r'(t)\|^3} \quad (12)$$

$$\tau = \frac{r'(t) \cdot [r''(t) \times r'''(t)]}{\|r'(t) \times r''(t)\|^2}. \quad (13)$$

κ indicates a direct correlation with the horizontal rotation capability of the UAV, while τ indicates the elevation capability of the UAV. Therefore, the triedro Frenet–Serret can be defined in matricial notation as a skew-symmetric matrix:

$$\begin{bmatrix} \dot{T} \\ \dot{N} \\ \dot{B} \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix}, \quad (14)$$

where the point over the variable indicates the derivative with respect to the parameter of arc-length s .

In summary, the space curve according to the formulation Frenet–Serret, defines a smooth curve that will be built from a spatial point and its tangent vector; this curve will be generated in the Euclidean 3D space depending on the pre-defined values of κ and τ , and finalize at a spatial 3D point after completing the arc-length s . However, our goal was to start from a defined point p_{init} , and build a curve that touches a target point p_{goal} ; therefore, the values of κ and τ have to fit so that when complete, the arc-length s will touch p_{goal} . That gives us an infinite number of possible values of κ and τ with which we could meet that goal. Even if the maximum and minimum values of κ and τ are bounded, the complexity of the problem is high. In [69] it is mentioned that the complexity in a 2D environment becomes NP-hard, and the need is demonstrated for path planning algorithms that generate short paths with bounded curvatures in complicated environments. The aim is to find for possible approximate values of κ and τ that do not exceed the turn capabilities of the UAV. In other words, starting from Figure 1, and assuming a configuration of the UAV as a triplet (ρ, κ, τ) , where $\rho = [P_1, \dots, P_5]$ is a dimensional vector that specifies n collision-free points, κ and τ are a set of curvatures and torsion along the path. The smooth curve starts from P_1 and reaches out to P_5 , and approaches the remaining ρ without affecting them, with values of κ and τ within the boundaries established by the maneuverability capabilities of the UAV. In summary, the selection of ρ_{goal} points which determine radii of the tangent curves to the ρ points, will be obtained by solving a multiobjective optimization problem (MOP). This MOP is stated in such a way that the values of all radii will be maximized simultaneously. Obviously, the optimizer handles these values, taking into account that they are in conflict (as radius of one of them is increased, consequently, the adjacent radius are reduced). Therefore, the MOP solver will try to find a trade-off solution that guarantees the best set of points ρ_{goal} for all tangent curves between control points ρ .

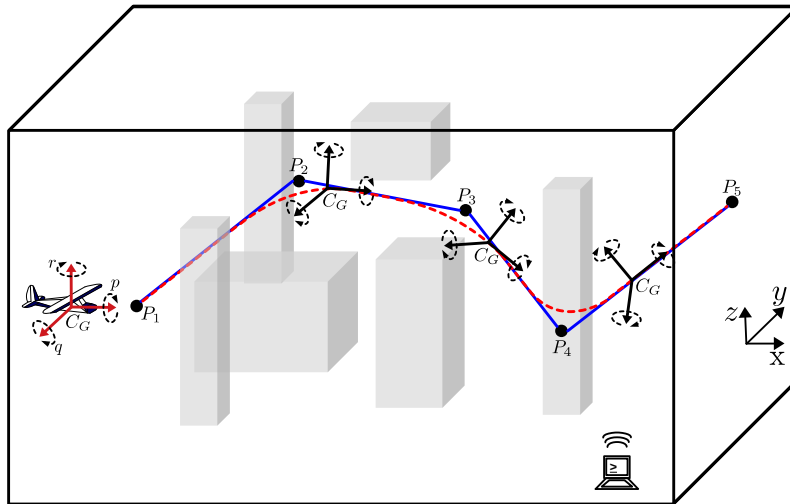


Figure 1. Perspective of the 3D flight problem for a fixed wing UAV, where C_G represents the position vector of the center of gravity of the UAV. The global coordinate system CS_g is placed at the origin, the orientation of the local body coordinate system CS_b expressed by Euler roll, pitch and yaw angles respectively, which have been defined by three unitary orthogonal vectors aligned with the three axes of the vehicle and centered at C_G . Finally, the angular velocities along the local axis of the UAV X , Y and Z are represented by p , q and r , respectively. The set of collision-free points P_i is represented by black dots; the blue line describes the discrete path built from 3D path planning; the red dotted line is the new smooth optimized path that the UAV could follow.

3. Problem Definition

Let us assume a workspace $\mathbb{W} = \mathbb{R}^3$, where it is possible to define a set of static or dynamic obstacles, such as ground or aerial boxes of different dimensions and locations (see Figure 1). The in-flight UAV receives data from its control station regarding environmental conditions and performs the necessary calculations to determine the best smooth 3D trajectory. Relevant data include the set of ordered 3D flight waypoints that are collision-free $\rho = [P_1, \dots, P_5]$ in the environment. The intrinsic maneuverability capabilities are determined by a R_p turning radius (which determines the vertical and horizontal turning limitations) defined by their flight speed. The aim is to start from ρ_{init} and reach ρ_{goal} in such a way that the UAV approaches the direct trajectory marked by the ordered sequence ρ . Therefore, $\rho = P_i(x_i, y_i, z_i)$ where $(i = 1, \dots, n)$ and n is the total set of collision-free spaces and can be expressed as a discrete interpolation sequence $\rho = f(t_i) \rightarrow \mathbb{R}$, where $f(t_i)$ is a set of nodes in 3D space. Therefore, it is possible to establish a set of sub-intervals $(n - 1)$ between $i = 1$ and $i = n$ partitioned in $[a, b]$, defined as:

$$[a, b] = [t_1, t_2] \cup [t_2, t_3] \cup \dots \cup [t_{n-2}, t_{n-1}] \cup [t_{n-1}, t_n] \tag{15}$$

$$a = t_1 \leq t_2 \leq \dots \leq t_{n-1} \leq t_n = b.$$

A linear union between pairs of points then results in $L : [a, b] \rightarrow (x, y, z)$. And can be expressed as a set of straight lines that mark a direct flight path $L(t)$ split into $(n - 1)$ piece-wise.

$$L(t) = \begin{cases} L_1(t) : t \in [t_1, t_2] \\ L_2(t) : t \in [t_2, t_3] \\ \vdots \\ L_n(t) : t \in [t_{n-1}, t_n] \end{cases} \tag{16}$$

$$L(t) = L_1(t) + L_2(t) + \dots + L_n(t).$$

Therefore, $L(t)$ is a linear interpolation function for the discrete sequence $\rho = f(t_i)$. In the same way, between the ρ points, there is a subset of $(n - 1)$ straight lines that join the init and the goal of the trajectory along the collision-free flight space.

However, a non-holonomic UAV cannot perform every type of maneuver defined by $L(t)$. In general, it is desirable to perform maneuvers with a high turning radius. Therefore, the approach presented in this work builds a smooth trajectory from ρ , that attempts to avoid inappropriate maneuvers using low values of κ and τ included within the boundaries of the UAV flight turn, while simultaneously closing in on the trajectory $L(t)$.

Let us assume, from Figure 1, that the blue line denoted as $L(t)$ is the direct path between the collision-free points of the environment, and the red dotted line is a smooth 3D spatial curve defined as $C(t)$. The construction of this 3D smooth curve $C(t)$ is done by joining a set of segments that can be of two types: spherical curves S (defined from a sphere of radius $R\rho$) or straight lines L . Thus, each S segment is defined by three continuous points of ρ ; this segment S has two points of tangency, one for each pair of adjacent straight lines $L(t)$ formed by the current set of three points ρ . Hence, each S segment may have infinite solutions, with each radius $R\rho$ resulting in different tangent points on the lines $L(t)$. Therefore, for each S segment, an infinite set of spheres can be defined, which will be linked through the relevant L segments or another S segment. Obviously, this approach to the problem suggests the existence of infinite combinations for the S and L segments. The way to address this issue has been through the approach of an MOP.

4. Methodology

This section describes the proposed methodology for the generation of 3D smooth trajectories. The proposed method is split into two parts, first detailing how the S segments were obtained, and then describing the union with the L segments.

4.1. Definition of Spherical Segment

Let us assume that from the result of a path planning, $\rho = [P_1, \dots, P_n]$ is the set of collision-free points of the environment described in Figure 2 (red points). This set of points is defined as $P_i(x_i, y_i, z_i) : i = \{1, \dots, n\}$, where $p_{init} = P_i(x_i, y_i, z_i) : i = \{1\}$ and $p_{goal} = P_i(x_i, y_i, z_i) : i = \{n\}$.

As indicated above, Figure 2b shows an osculating sphere (oS) [68] defined with a minimum turning radius value $R\rho$, located between the set of the first 3 P_i and tangential to the straight lines $L(t)$ formed between the same set of P_i . Therefore, taking into account the number of collision-free points ρ , the set of spheres is equal to $G_i : i = \{1, \dots, n - 2\}$, as shown in Figure 2c (orthogonal view).

Figure 2b shows the first $G_i : i = 1$ located among the three first P_i s. Therefore, it is possible to define a plane $\pi_i : i = 1$ between the same points P_i , which will have an angle in relation to the location of the current set of points P_i , as can be seen in the Figure 3a,b. The importance of the definition of this plane is given by the fact that within it is contained the center of G_i with radius $R\rho$. In this way, there is a self-contained curve S_i (as a series of points along the Euclidean space) on the surface of the sphere and tangent to $L(t)$ with t_2 and t_3 in plane π_i , as shown in Figure 3; hence, the S_i curve segment (black line) is defined as:

$$\begin{aligned}
 S_i(t) &= [S_x, S_y, S_z] \\
 \left. \begin{aligned}
 S_{i_x} &= x_0 + R\rho * \sin(\psi) * \cos(\varphi) \\
 S_{i_y} &= y_0 + R\rho * \sin(\psi) * \sin(\varphi) \\
 S_{i_z} &= z_0 + R\rho * \cos(\psi)
 \end{aligned} \right\} \begin{aligned}
 &\varphi_1 \geq \varphi \geq \varphi_2 \\
 &\quad \wedge \\
 &\psi_1 \leq \psi \leq \psi_2.
 \end{aligned}
 \end{aligned} \tag{17}$$

where, x_0, y_0 and z_0 together represent the center of G_i . The curve S_i performs a horizontal and vertical path due to the angular ranges of ψ and φ , which implies variations in the values of κ and τ (these have a direct connection to $R\rho$ and the arc-length of S_i). Consequently, if the value of $R\rho$ grows, S_i also grows, while κ and τ decrease.

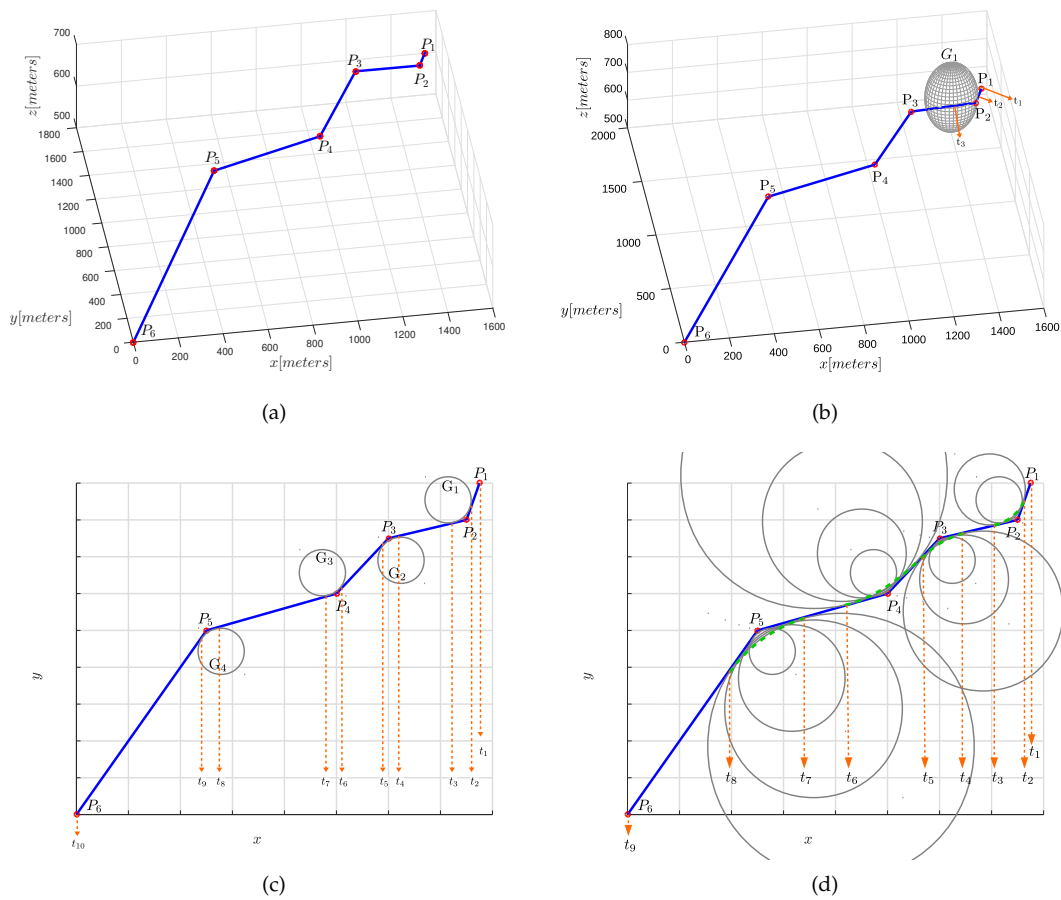


Figure 2. Smooth path planning problem. The red dots represent ρ and the blue line is the path made up of straight-lines $L(t)$. (a) Result of path planning with ρ collision-free points. (b) Definition of a sphere with a relation of the $R\rho$ minimum. (c) Set of G_i over ρ . (d) Example of optimal smooth trajectory, with optimized κ and τ represented by dotted green lines in orthogonal view.

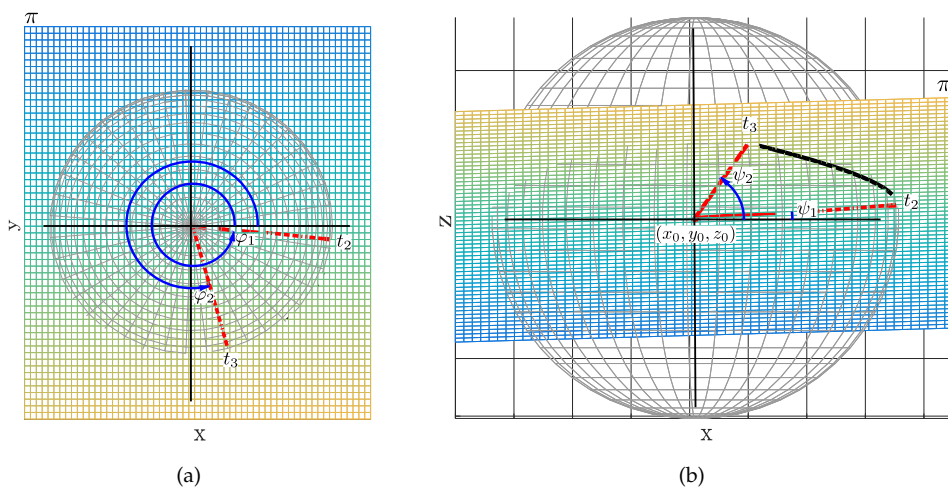


Figure 3. Spherical smooth path, where the black line shows the curve in a segment S_i along the plane π_i . The red lines show the union from the center of coordinates of the sphere to the points of intersection between the sphere and the π_i plane resulting in the spherical semicurve. (a) View perpendicular to the horizontal plane (x, y) . (b) Perpendicular view to the vertical plane (x, z) .

Remark 1. If the plane π_i is parallel to the horizontal plane (x, y) of the environment, then $\tau = 0$ because the movements of the UAV will be horizontal. In the same way, if π_i is parallel to the vertical plane (x, z) of the environment, then $\kappa = 0$.

However, before applying Equation (17), it is necessary to determine the location of the points (x_0, y_0, z_0) so that G_i is tangent at a point on its surface with $L(t)$, as shown in Figure 2b on the points $(t_2$ and $t_3)$. Nevertheless, it should be noted that there is an angle between each pair of $L(t)$, and this leads to G_i approaching or moving away from the lines and their tangent points. Therefore, the geometric analysis applied to arrive at an optimal solution is detailed below.

First, a vector direction in space can be defined as $\vec{v} = p - q : p \wedge q \in \mathbb{R}^3$. Therefore, starting from the known data $\rho = [P_1, \dots, P_n]$, taking Figure 2 as an example, where it is assumed that the collision-free initial points are $(P_i : \{i = 1, \dots, 3\})$, a first set of two vectors is defined as:

$$\left. \begin{aligned} \vec{u}_i &= p - q : p = P_{(i+1)}, q = P_{(i)} \\ \vec{v}_i &= p - q : p = P_{(i+1)}, q = P_{(i+2)} \end{aligned} \right\}, i = 1. \tag{18}$$

Just like a perpendicular vector from \vec{u}_i to \vec{v}_i , denoted as $\vec{\eta}$, the normal vector is defined as:

$$\vec{\eta} = \vec{u}_i \times \vec{v}_i. \tag{19}$$

Consequently, the parametric equation of the π_i plane containing three points is defined as:

$$\pi_i = \left\{ \begin{aligned} (x - p_x) \\ (y - p_y) \\ (z - p_z) \end{aligned} \right\} * [\vec{\eta}] : p_x = P_{(i+1)_x}, p_y = P_{(i+1)_y}, p_z = P_{(i+1)_z}. \tag{20}$$

In the same way, the Euclidean distance defined between two points p and q is given by

$$d(p, q) = \sqrt{\sum (p - q)^2}. \tag{21}$$

Therefore, two distances can be defined as $du_i : p = P_{(i+1)}, q = P_{(i)}$ and $dv_j : p = P_{(i+1)}, q = P_{(i+2)}$. Finally, the angle between \vec{v}_i and \vec{u}_i is defined by:

$$\angle(\vec{u}_i, \vec{v}_i) = \phi_i = \tan^{-1} \frac{\|\vec{u}_i \times \vec{v}_i\|}{\vec{u}_i \cdot \vec{v}_i}. \tag{22}$$

Therefore, with Equation (22) and $R\rho$ known, the tangential points at the lines $L(t)$ can be located at a distance defined as:

$$\sigma_i = \frac{R\rho}{\phi_i/2}. \tag{23}$$

Thereby, two spatial points defined as pUi_i and pUg_i located in the direction of the vector \vec{u}_i provide that $Pi = P_{i+1}, Pg = P_i$ and $d(p, q) = du_i$; thus

$$\begin{aligned} \gamma &= \sigma_i/d(p, q) \\ pUi_j &= (Pi - Pg) * \gamma + Pi \\ pUg_j &= -(Pi - Pg) * \gamma + Pi. \end{aligned} \tag{24}$$

In the same way, two points pVi_i and pVg_i can be defined in the vector direction \vec{v}_i , so long as $Pi = P_{i+1}, Pg = P_{i+2}$ and $d(p, q) = dv_i$, according to Equation (24). Hence, the perpendicular bisector of pUi_i and pVi_i on the plane π_i determines the center of the sphere (x_0, y_0, z_0) . Figure 4a shows the application of the Equations (18)–(24), which can be repeated throughout the successive collision-free points ρ (this first Algorithm 1 is summarized in pseudocode). Between the centers of the spheres

(x_0, y_0, z_0) and the points of intersection with $L(t)$ are the displacement angles φ and ψ of the segment S_i , as can be seen in the Figure 3, where $pUi_i = t_2$ and $pVi_i = t_3$.

Remark 2. Regardless of the angle condition produced by the pair of straight lines $L(t)$ denoted in the Equation (22), the angle formed between the points of intersection on the sphere G_i , seen from its center towards the vertical or horizontal component, does not exceed 90° in any case; that is, $(0^\circ < \varphi < 90^\circ)$ and $(0^\circ < \psi < 90^\circ)$.

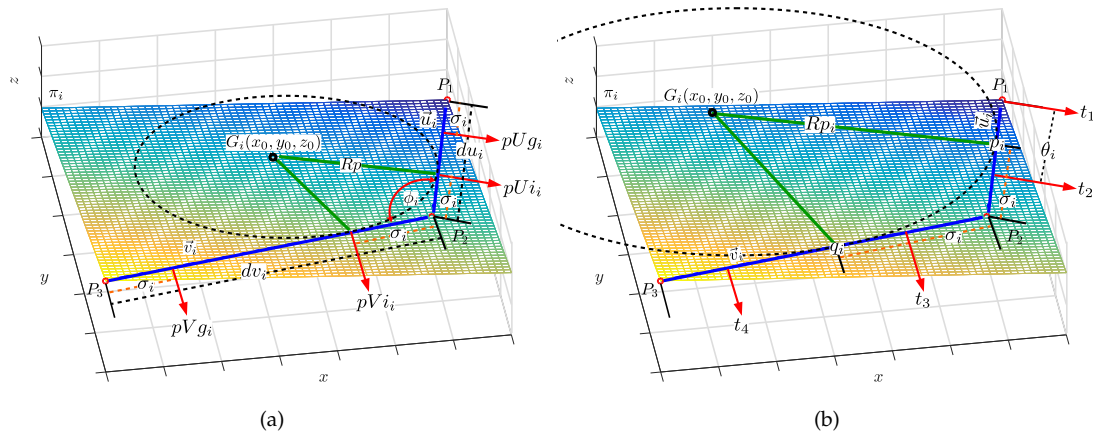


Figure 4. Description of the methodological basis. The circle of black dots shows the osculating sphere (oS) ; the green line is the radius of turn Rp_i . (a) Sphere location G_i with minimum turning radius. (b) Sphere location G_i with upper turning radius displaced within the bounds given by $[t_1, t_2]$ and defined by the value of θ_i .

Algorithm 1 First set of G_i

```

1:  $\rho = [P_1, \dots, P_n] \rightarrow$  Path planning problem result.
2:  $Rp =$  minimum sphere radius.
3:  $k = 1$  counter for the set of intervals  $t$ .
4: for  $i = 1 : n - 2$  do
5:    $(\vec{u}_i, \vec{v}_i) =$  vectors .directions;
6:    $\phi_i =$  angle.betweenLines( $\vec{u}_i, \vec{v}_i$ )
7:    $\sigma_i =$  distance.intersectionPoint( $Rp, \phi_i$ );
8:    $(pUi_i, pUg_i) =$  intersectPoint( $P_{i+1}, P_i, \sigma_i$ );
9:    $(pVi_i, pVg_i) =$  intersectPoint( $P_{i+1}, P_{i+2}, \sigma_i$ );
10:   $G_i(x_0, y_0, z_0) =$  bisectPoint( $pUi_i, pVi_i, \pi_i$ );
11:  if  $i == 1$  then
12:     $t_k = P_1$ ;
13:     $t_{k+1} = pUi_i$ ;
14:  else
15:     $t_k = pUg_i$ ;
16:     $t_{k+1} = pUi_i$ ;
17:  end if
18:  if  $i == n - 2$  then
19:     $t_{k+2} = pVi_i$ ;
20:     $t_{k+3} = P_n$ ;
21:  end if
22:   $k = k + 2$ ;
23: end for

```

Algorithm 1 summarizes the geometric procedure followed. In line 4, a loop is started that performs through all the collision-free points marked by ρ . From line 5 to line 9, the necessary

computations are made to determine the center of G_i (line 10). The definitions of the intervals containing the S and L segments are in lines 12, 13, 15, 16, 19 and 20.

The described process shows the geometric analysis for the location of the set of spheres G_i defined with constant radius Rp , as can be seen in Figure 2c. In addition, there is a set of four segments S and another set of five segments L , being segments S —those comprised by the intervals $[t_2, t_3]$, $[t_4, t_5]$, $[t_6, t_7]$ and $[t_8, t_9]$, and the segment L included by the intervals $[t_1, t_2]$, $[t_3, t_4]$, $[t_5, t_6]$, $[t_7, t_8]$ and $[t_9, t_{10}]$. The goal now is to increase the radius Rp in each segment, so that the values of κ and τ along the curve are minimized, and the solution is to increase the radius Rp_i in each G_i .

The solution adopted in this work is to move the intersection point of each sphere G_i in the direction of the adjacent segment $L(t)$. Consequently, $G_i : i = 1$ approximates symmetrically to the intervals t_1 and t_4 , $G_i : i = 2$ makes the corresponding approximation to the intervals t_3 and t_6 , etc. Therefore, in Figure 4b, the segments $L(t)$ can be seen adjacently to $G_i : i = 1$, denoted as $[t_1 \equiv P_1, t_2 \equiv pU_i, t_3 \equiv pV_i, t_4 \equiv pVg_j]$. Thus, between the adjacent intervals $[t_1, t_2]$ a vector is defined $\vec{u}_i = t_2 - t_1$, and associated with this vector is a spatial point p_i defined by the parametric equation:

$$\left. \begin{aligned} p_{i_x} &= t_{2_x} + \theta_i * \vec{u}_{i_x} \\ p_{i_y} &= t_{2_y} + \theta_i * \vec{u}_{i_y} \\ p_{i_z} &= t_{2_z} + \theta_i * \vec{u}_{i_z} \end{aligned} \right\}, 0 \leq \theta_i \leq 1, \tag{25}$$

where θ_i defines the space point p_i along \vec{u}_i and within the intervals $[t_1, t_2]$. Therefore, the value of the distance σ_i from P_{i+1} to p_i is defined according to the equation on (21), being $p = P_{i+1}$ and $q = p_{i+2}$. A symbolic space point is defined q_i between the intervals $[t_3, t_4]$ with direction $\vec{v}_i = t_3 - t_4$ at the same distance σ_i . Then σ_i also has the angle ϕ_i , and according to the Equation (22) it is possible to define a new Rp_i according to Equation (23), which would have a higher radius value. Finally, the perpendicular bisector of p_i and q_i on the plane π_i determines the center of $G_i(x_0, y_0, z_0)$ (see Figure 4b). Therefore, as the center of G_i is defined, the Equation (17) defines the segment S_i —and over this segment we find lower values of κ and τ according to the Equations (12) and (13).

Multiobjective Problem Definition (MOP)

Given Equation (25), it is important to note that any value of θ_i between 0 and 1, defines a space point between the interval $[t_1, t_2]$. In the same way, it is important to mention that within the boundaries of θ_i , there is an infinite number of space points with an infinite number of radius Rp_i and its corresponding infinite number of G_i , with which its corresponding segments S_i , can be built.

Therefore, to obtain an optimal solution, a multiobjective problem (MOP) is solved using evolutionary algorithms [70] based on the concept of ϵ -dominance [71]. To do this, it is necessary to define the decision variables, the initial conditions of the process, the constraints of the MOP and the index vector to be optimized to represent the Pareto front. If it is assumed that the number of spheres G_i is equal to m , and the number of objectives for each G_i is equal to two, then $J^{ideal}(\theta) = [J_1(\theta), J_2(\theta), \dots, J_{2*m}(\theta)]$ is the objectives vector, where J_i denotes the i^{th} objective. Consequently, $J_i^A = \min(\kappa(\theta_i))$, $J_i^B = \min(\tau(\theta_i)) \in G_i : [i = 1, \dots, m]$, where J_i^A and J_i^B depend on the decision variables vector θ . Assume D as a decision space within a subset \mathbb{R}^D , where θ is the decision variable vector composed of a set of θ_i for all $i \in 1 < i < m$, where θ_i is $[0, 1]^D$. Consequently, the MOP problem can be stated as:

$$\min_{\theta \in D} [J_i^A(\theta), J_i^B(\theta)]_{1 \times (2*m)}, \forall i \in 1 \leq i \leq m. \tag{26}$$

where

$$J_i^A = \frac{\|S'_i(t) \times S''_i(t)\|}{\|S'_i(t)\|^3},$$

from Equation (12)

$$J_i^B = \frac{S_i'(t) \cdot [S_i''(t) \times S_i'''(t)]}{\|S_i'(t) \times S_i''(t)\|^2},$$

from Equation (13)

$$\theta = [\theta_i]_{1 \times m}, \forall i \in 1 \leq i \leq m$$

subject to:

$$S_i(t) = \begin{cases} S_{i_x} = x_0 + Rp_i * \sin(\psi) * \cos(\varphi) \\ S_{i_y} = y_0 + Rp_i * \sin(\psi) * \sin(\varphi) \\ S_{i_z} = z_0 + Rp_i * \cos(\psi) \end{cases}$$

from ec. (17)

$$Rp_i = \sigma_i * (\phi_i/2)$$

from Equation (23)

$$\sigma_i = \sqrt{\sum (p_i - P_{i+1})^2}$$

from Equation (21)

$$p_i = \begin{cases} p_{i_x} = t_{2_x} + \theta_i * \vec{u}_{i_x} \\ p_{i_y} = t_{2_y} + \theta_i * \vec{u}_{i_y} \\ p_{i_z} = t_{2_z} + \theta_i * \vec{u}_{i_z} \end{cases}$$

from Equation (25)

$$\theta_i \in [0, 1]^D.$$

In summary, the aim is to find an optimal 3D smooth curve that minimizes κ and τ in each of the possible S_i . It is important to mention that the adjacent spheres G_i can grow into each other, until a maximum of $q_i \in \vec{v}_i \equiv p_{i+1} \in \vec{u}_i$, which implies a decrease in the total set of segments, as described Figure 2d, where the green dotted line shows the set of S_i segments belonging to G_i .

An example of reconstruction according to the response Θ_p^* can be seen in Figure 2d, where the S reconstruction is made in four segments, defined by the boundaries $[t_2, t_3]$, $[t_4, t_5]$, $[t_5, t_6]$ and $[t_7, t_8]$; the S segments belonging to $C(t)$ are defined according to Equation (17).

In contrast, and with reference to Figure 2d, the L segments are defined by the rest of the boundaries, those boundaries being $[t_1, t_2]$, $[t_3, t_4]$, $[t_6, t_7]$ and $[t_8, t_9]$.

4.2. Definition of Straight-Line Segment

A segment path of L in a straight-line can be described by two points in the Euclidean space. Figure 2d shows an example of an L segment defined by the $[t_1, t_2]$ points, where the direction of the line is given by the flight path of the UAV. Therefore, \vec{u} (Figure 5) is a unit vector that points in the direction of the desired orientation, and with d defined as the distance between t_1 and t_2 according to Equation (21). Therefore, the L segments will be described, in general, as:

$$L(t) = \{r \in \mathbb{R}^3 : r = (t_1 - t_2) * \gamma - t_1\} \rightarrow 0 \leq \gamma \leq d. \tag{27}$$

Finally, the interpolation of S and L build a final 3D smooth curve on the plane (x, y, z) .

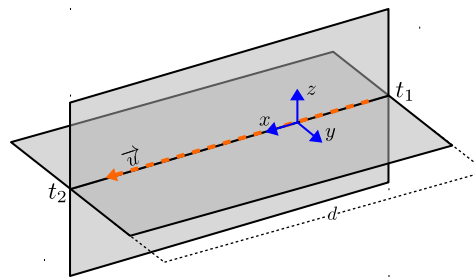


Figure 5. Straight-line segment.

5. Experiments and Results

This section presents the results of the computer simulation using Matlab/Simulink software and flightGear flight simulator for graphical visualization.

In this section we analyze five scenarios in 3D space, similar to the methodology proposed in [44]. Recursive rewarding modified adaptive cell decomposition (RR-MACD) splits the 3D environment like a discrete mesh of collision-free voxels. In particular, it places the UAV within an initial voxel and determines which of the adjacent voxels is the most suitable to make a displacement. To determine the best choice of displacement, the set of adjacent voxels have a set of associated constraints (conditions such as distance, vertical, and horizontal movement angles, battery consumption, etc.) to be satisfied before performing a discrete displacement. The voxel that minimizes the total effort and satisfies the constraints will be the next collision-free point. The RR-MACD methodology gives two sets of results based on the defined constraints. The results presented in [44] are shown in summary form in Table 1, where the first column shows the scenario number. The second column shows RR-MACD with four constraints and the RR-MACD with 10 constraints in the third column shows the conditions to solve the path planning problem. The 3D control points reflected in Table 1, $\rho_x(F) \approx \rho$, are the starting points for analyzing the method described in this paper for generating 3D smooth curves. Finally, it is important to note that the algorithms have been run on an Intel(R) Core(TM) i7-4790 3.60 GHz CPU (Manufacturer: Gigabyte Technology Co., Ltd., Model: B85M-D3H) with 8Gb RAM and S.O. Ubuntu Linux 16.04 LTS. The algorithms were programmed in MATLAB version 9.4.0.813654 (R2018a).

Table 1. 3D path planning results. The number of free-collision voxels within a defined environment is indicated as S_{free} , while the number of discrete 3D nodes built by [44] is denoted by $\rho_x(F)$ (3D final path).

Env.	RR-MACD 4 Constraints		RR-MACD 10 Constraints	
	#	#	#	#
	S_{free}	$\rho_x(F)$	S_{free}	$\rho_x(F)$
# 1	115	18	202	27
# 2	27	8	35	10
# 3	19	6	16	7
# 4	11	6	51	10
# 5	19	7	35	10

It is important to mention that the characteristics of the UAV assumed in the experiments have been taken from [65], whose study has been carried out on a fixed wing UAV model *kadett 2400*, represented by six states $(x, y, z, \phi, \theta, \psi)$, where the first three states define the position vector of the UAV's global coordinate system, located at the origin of its center of gravity. The last three are the Euler angles of roll, pitch and yaw respectively, which define the orientation of the UAV.

Finally, the aim of the simulations is for the UAV to maintain its continuous flight at a defined constant speed of 18 m/s, within an established minimum radius of curvature $R_p = 33$ m to achieve smooth behavior and without maneuvers that could endanger the integrity of the aircraft.

It is important to remember that because the number of $\rho_x(F) = [P_1, \dots, P_n]$ collision-free points is greater than five, a proper visualization method is essential to the decision making process for the final solution. Thus, the graphical representation method called level diagram [72] has been used, which consists of representing each objective and each design parameter in separate diagrams, all of which are synchronized with their y axis. Synchronization is made with the normalized distance of each point from the Pareto front to the ideal point. Therefore, with a brief training this representation offers a good visual understanding of the compromise reached on the Pareto front.

5.1. Bézier

To compare the results, an approximation of curves has been applied using Bézier [68], which enables the generation of trajectories for non-holonomic systems through a set of discrete control points, where the curve in a multidimensional environment can be described as:

$$B(t) = \sum_{i=0}^n P_i b_{i,n}(t), \quad t \in [0, 1] \quad (28)$$

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-1}, \quad i = 0, \dots, n$$

Bernstein-Bézier generates a finite vector of points belonging to the curve and guarantees to go through the first and last control points (translated as $\rho = [P_1, \dots, P_n]$) and remaining inside the convex envelope.

5.2. Application Example

To represent visual and numerical results from Table 1, the results for the #3 environment with RR-MACD with four constraints are detailed below. As in this example, the total number of P_i equals 6, and this means that the decision criteria of Equation (26) $\rightarrow m = 4$. Therefore, there are four values of κ and four values of τ ; i.e., $\Theta_p^* = (J_1(\theta_1) = \kappa_1, J_3(\theta_2) = \kappa_2, J_5(\theta_3) = \kappa_3, J_7(\theta_4) = \kappa_4, J_2(\theta_1) = \tau_1, J_4(\theta_2) = \tau_2, J_6(\theta_3) = \tau_3 \text{ and } J_8(\theta_4) = \tau_4)$, as can be seen in Figure 6.

It is important to mention that the interpolation of segments S and L builds a set of 3D smooth curves, all of which are possible solutions. It is, therefore, necessary to address a decision stage (decision maker (DM)) that selects one of them; i.e., a point on the Pareto front. In this work, the selection criteria based on the shortest distance to the ideal point have been used. Figures 6 and 7 show the selected point in red from $J(\Theta_p^*)$ and Θ_p^* , which have been selected using the ∞ -norm standard.

Figure 8a shows the build of the 3D smooth curve $C(t)$, while Figure 8b shows the best optimization in terms of curvature κ and total torsion τ , according to Equations (12) and (13). It is important to note that the mathematical mean of the values of the geometric variables κ and τ tends to be low. However, in some particular cases an increase is detected due to the change in direction of the flight; i.e., when the UAV is terminating an S segment in one direction and another S segment begins in the opposite direction. The curve generated by Bézier $B(t)$ is shown as a yellow line, a more direct route can be seen between the init and goal point. However, this curve is very close to the bottom obstacle. To solve this, different authors propose modifying the control points ρ , or adding new points within the points established initially, as remarked in Section 1.

Figure 9 shows the set of four additional examples from Table 1, in order to represent the functionality of the algorithm. Similarly, it is important to note that the number of ρ was different in each experiment, as were the altitudes, which guaranteed movement and 3D planning. It is also important to mention that the first environment shown in Figure 9a has smaller flight dimensional

characteristics, so the turning radius in this example was set at $R_p = 3$ m with an average flight speed of 1.7 m/s.

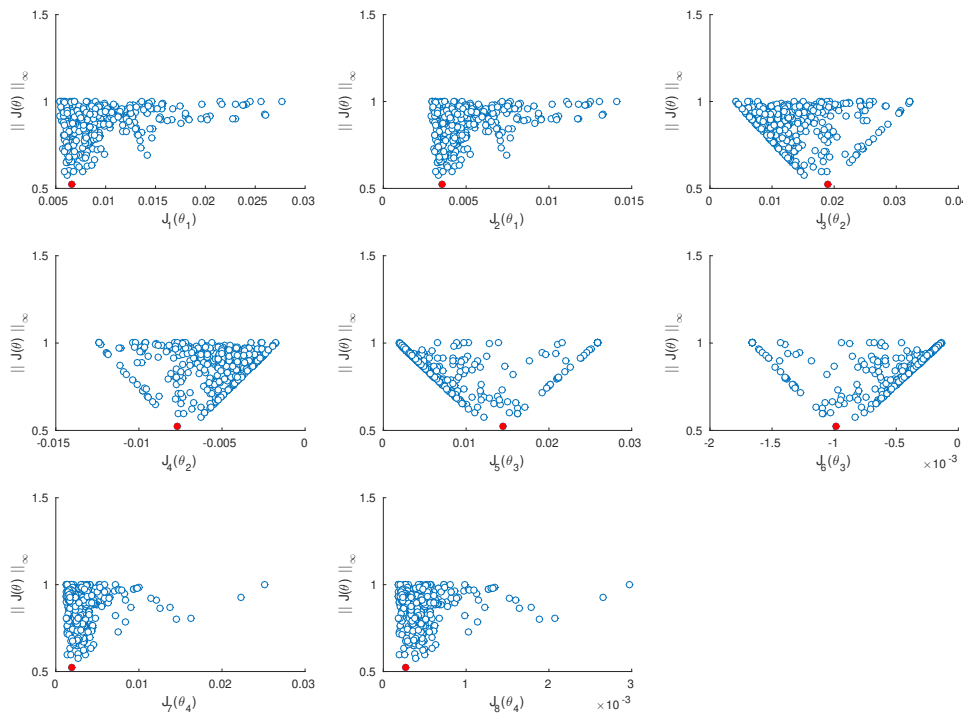


Figure 6. Representation of Pareto front using ∞ – norm. The J pair sub-indices represent the κ values in each S_i segment, while the J odd sub-indices represent the τ values in the same S_i segments. Targets closest to J^{ideal} are shaded in red circles.

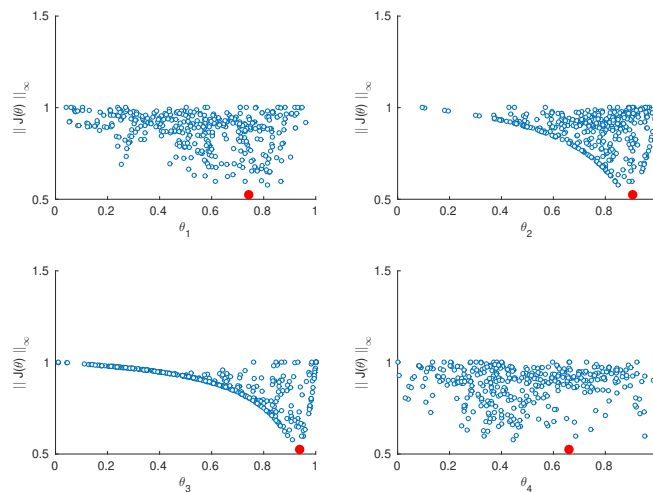


Figure 7. Representation of Pareto’s optimal parameters. Targets closest to J^{ideal} are shaded in red circles.

To describe the different groups of trajectories generated by $L(t)$, $C(t)$ or $B(t)$ displayed in Figure 9, Table 2 shows the flight results from the init to goal point in terms of distances; according to one of the results of each environment set by Table 1. It can be seen that the set of greater distances corresponding to the path in the form of the straight line marked by $L(t)$, $C(t)$ reduces the distance by $L(t)$. As $B(t)$ makes an approximation (as a mathematical expression) between the set of ρ of each environment, its route is the shortest. The column “EAA Error (meters)” shows the approximate absolute error $EAA = \frac{1}{n} \sum_{i=1}^n |A - B|$, where $A = L(t)$ and $B = C(t) \wedge B = B(t)$. Therefore, the results

of the column *EAA Error (meters)* show a greater approximation by $C(t)$ and which results in a better avoidance of obstacles.

Table 2. Flight distance of the curves. The column “Flight distance (meters)” shows the distance in meters at the initial and final free collision points marked by ρ . The column “EAA Error (meters)” shows the average error in meters along the trajectories.

Env.	Flight Distance [Meters]			EAA Error [Meters]	
	$L(t)$	$C(t)$	$B(t)$	$L(t)$ vs $C(t)$	$L(t)$ vs $B(t)$
#1	182.929355	174.002834	148.911388	0.622684	3.248545
#2	1728.757868	1610.781941	1453.060601	17.234613	41.453691
#3	1863.391222	1721.505017	1526.055284	14.600159	56.678212
#4	1936.078758	1860.263202	1772.944453	9.871725	36.617234
#5	1873.814514	1839.965587	1743.723244	9.891240	36.614752

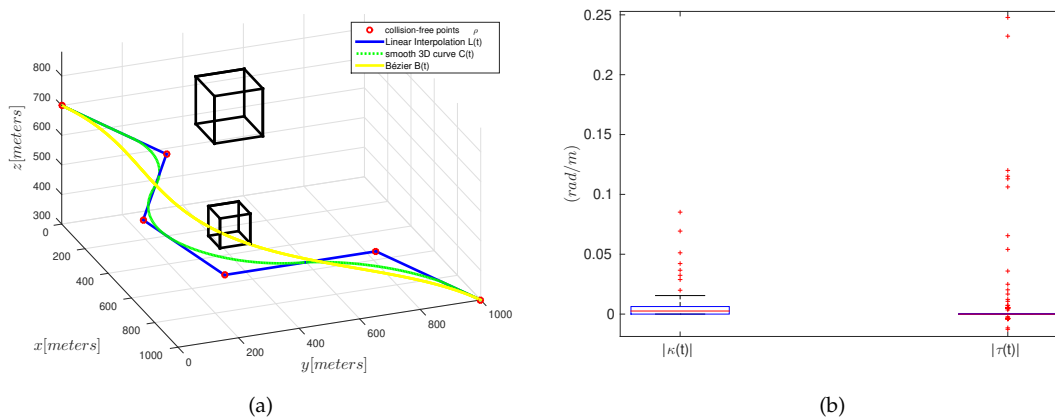


Figure 8. Example of a 3D environment cluttered of obstacles. (a) Reconstruction of 3D trajectories, where the obstacles are the black boxes; the red circles show the collision-free points; and the green line shows the final 3D trajectory $C(t)$. The yellow line represents the Bézier $B(t)$ curve. (b) Geometric averages of the variables κ and τ of the final path.

Similarly, Table 3 shows a set of results referring to the five environments analyzed. The averages of κ and τ generated along each smooth curve shows that $B(t)$ exceeds $C(t)$. However, in the first environment there is a collision caused by the $B(t)$ curve.

Table 3. Average results of κ and τ along the curves $C(t)$ and $B(t)$. The column “Collision” indicates collision (x) or no collision (o) of a curve against an obstacle.

Env.	Curve	κ	τ	Collision
#1	$C(t)$	0.157961	0.185973	o
	$B(t)$	0.019513	0.092539	x
#2	$C(t)$	0.007138	0.159732	o
	$B(t)$	0.001082	0.006652	o
#3	$C(t)$	0.004556	0.185806	o
	$B(t)$	0.001068	0.004442	o
#4	$C(t)$	0.003445	0.574121	o
	$B(t)$	0.000812	0.003332	o
#5	$C(t)$	0.004515	0.135183	o
	$B(t)$	0.000643	0.004253	o

Finally, the results produced by the simulation of the UAV kaddet 2400 matlab/simulink/flightGear over the environment #3 are shown in Figure 10. The geodesic coordinates of Figure 10a are

expressed in decimal degrees; in this example the flight starts with an altitude of 500.4 m, and after the maneuvers performed by the UAV, it reaches a new altitude of 603.1 m. Figure 10b shows the UAV model in flight using FlightGear Simulator as visualization platform.

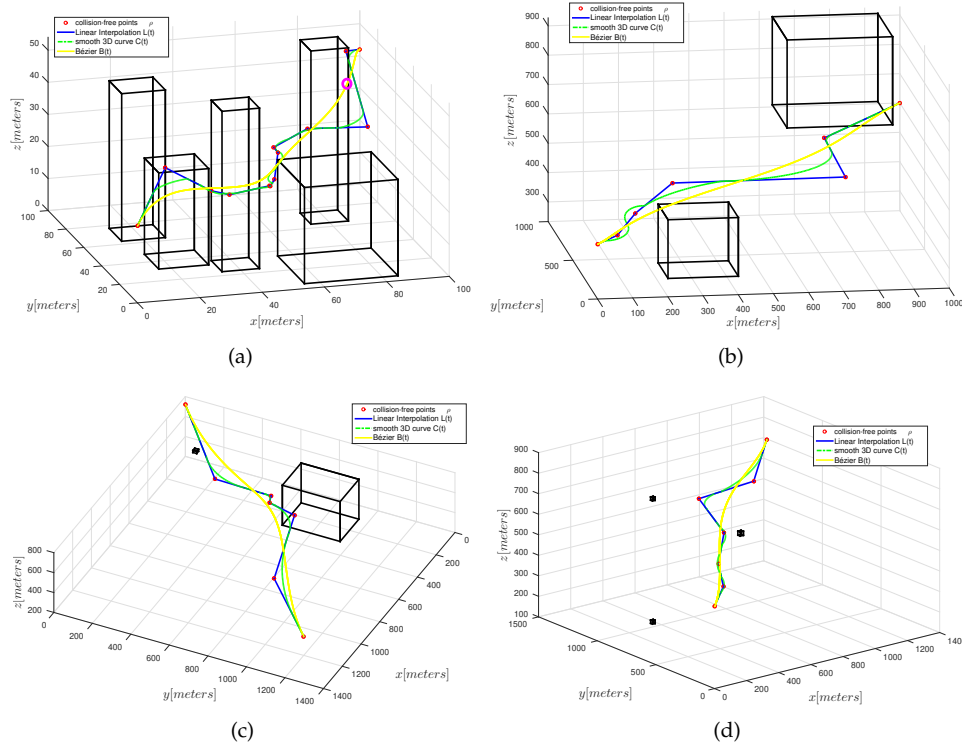


Figure 9. Additional 3D environment experiments in which the obstacles are the black boxes and the green line shows the final 3D trajectory $C(t)$; the yellow line represents the Bézier $B(t)$ curve. (a) (Table 1—Environment #1.) It represents an unstructured environment with different buildings, where a collision between $B(t)$ and a building (collision marked as a circumference of magenta color) can be seen. (b) (Table 1—Environment #2.) 3D environment with two obstacles of different sizes. (c) (Table 1—Environment #4.) 3D environment with two obstacles of different sizes. (d) (Table 1—Environment #5.) 3D environment with three small aerial obstacles.

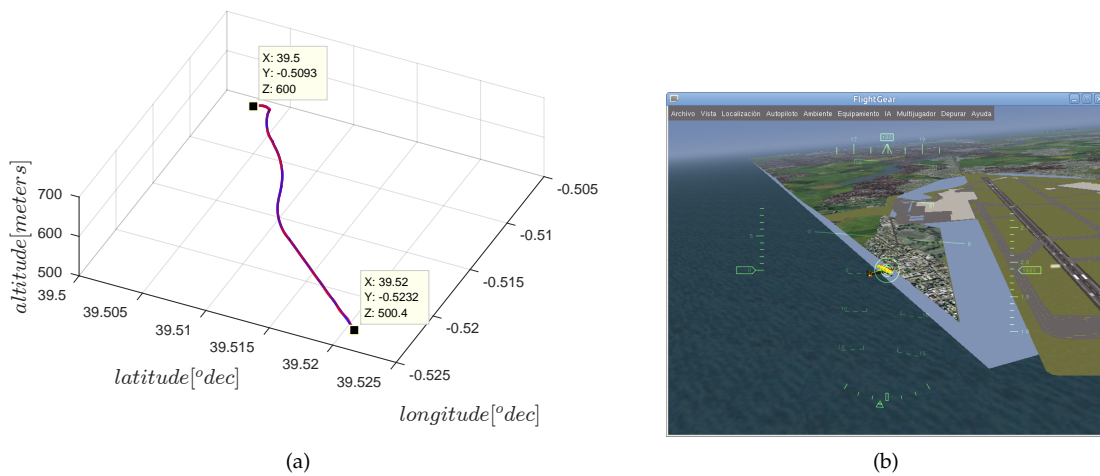


Figure 10. Simulation of the UAV flight simulink, where the starting point is $[39.52, -0.5232, 500]$ and the target point is $[39.50, -0.5093, 600]$. (a) Flight of the UAV, where the blue line is the path calculated from the process described and the red line is the actual path of the UAV. (b) View from a model perspective.

6. Conclusions and Future Works

This paper describes an approach to the generation of a continuous 3D smooth path that enables consideration of the operational constraints of fixed-wing UAVs.

Firstly, the document describes the formulation of the problem by defining two types of segments within the trajectory: the *S* segments as a set of sphere segments that ensure a continuous and minimum curvature profile, and then the definition of *L* segments that generally connect *S*.

Next, we proposed the resolution of an MOP problem to obtain the numerical values of the parameters of the trajectory, given that the problem has infinite feasible solutions. When solving an MOP problem, the DM stage is essential to finally select the desired point from the Pareto set of optimal solutions.

It is important to remember that with methods such as classic Bézier or B-splines curves, you can define the number of samples along the path. However, the distance measured between one point and the next is not the same or even close (the difference can be large). These types of curves are useful in relatively simple environments (few obstacles); however, as the number of obstacles grows, the control points increase due to trajectory planning. Consequently, the construction of the curve can cause collisions. This work enables a constant approach distance between pairs of contiguous points.

The kinematic constraints of UAVs have been considered in this work, in the same way that dynamic constraints could be calculated mathematically. However, an important consideration that can enhance the generation of new trajectories in a new job is to increase the optimization criterion by improving variables such as energy consumption or incomplete data in dynamic environments.

Connected with the results shown in this paper, several future works arise. For example, integration of dynamic obstacles (UAVs swarms or other aircraft systems) into the flying area. From the optimization point of view, the proposal can be improved by taking into account dynamic constraints (i.e., inertia, wind disturbances, torque forces, etc.) into the MOP problem. Similarly, new cost indexes as flight time or/and spent energy could be added to the optimization problem. Finally, implementation of the proposed algorithm under real conditions (UAV in an outdoor environment) and its application to different uses (such as satellite trajectory generation [73]) will be investigated.

Author Contributions: Conceptualization, F.S.; formal analysis, F.S., J.S., S.G.-N. and R.S.; methodology, F.S.; supervision, J.S., S.G.-N. and R.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the Spanish Ministerio de Ciencia, Innovación y Universidades for providing funding through the project RTI2018-096904-B-I00 and the local administration Generalitat Valenciana through projects GV/2017/029 and AICO/2019/055. Franklin Samaniego thanks IFTH (Instituto de Fomento al Talento Humano) Ecuador (2015-AR2Q9209), for its sponsorship of this work.

Acknowledgments: The authors would like to thank the editors and the reviewers for their valuable time and constructive comments.

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

References

1. Fagnant, D.J.; Kockelman, K. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transp. Res. Part A Policy Pract.* **2015**, *77*, 167–181. [[CrossRef](#)]
2. Kyriakidis, M.; Happee, R.; de Winter, J.C. Public opinion on automated driving: Results of an international questionnaire among 5000 respondents. *Transp. Res. Part F Traffic Psychol. Behav.* **2015**, *32*, 127–140. [[CrossRef](#)]
3. Münzer, S.; Zimmer, H.D.; Schwalm, M.; Baus, J.; Aslan, I. Computer-assisted navigation and the acquisition of route and survey knowledge. *J. Environ. Psychol.* **2006**, *26*, 300–308. [[CrossRef](#)]
4. Morales, Y.; Kallakuri, N.; Shinozawa, K.; Miyashita, T.; Hagita, N. Human-comfortable navigation for an autonomous robotic wheelchair. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2737–2743. [[CrossRef](#)]
5. Krotkov, E.; Hebert, M. Mapping and positioning for a prototype lunar rover. In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 3, pp. 2913–2919. [[CrossRef](#)]

6. Rodríguez-Seda, E.J. Decentralized trajectory tracking with collision avoidance control for teams of unmanned vehicles with constant speed. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 1216–1223. [\[CrossRef\]](#)
7. Ren, X.; Cai, Z. Kinematics model of unmanned driving vehicle. In Proceedings of the 2010 8th World Congress on Intelligent Control and Automation, Jinan, China, 7–9 July 2010; pp. 5910–5914. [\[CrossRef\]](#)
8. Dixon, C.; Frew, E.W. Decentralized extremum-seeking control of nonholonomic vehicles to form a communication chain. In *Advances in Cooperative Control and Optimization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 311–322. [\[CrossRef\]](#)
9. Jun, J.Y.; Saut, J.P.; Benamar, F. Pose estimation-based path planning for a tracked mobile robot traversing uneven terrains. *Robot. Auton. Syst.* **2016**, *75*, 325–339. [\[CrossRef\]](#)
10. Li, Y.; Ding, L.; Liu, G. Attitude-based dynamic and kinematic models for wheels of mobile robot on deformable slope. *Robot. Auton. Syst.* **2016**, *75*, 161–175. [\[CrossRef\]](#)
11. Mekonnen, G.; Kumar, S.; Pathak, P. Wireless hybrid visual servoing of omnidirectional wheeled mobile robots. *Robot. Auton. Syst.* **2016**, *75*, 450–462. [\[CrossRef\]](#)
12. Xu, J.; Wang, M.; Qiao, L. Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. *Ocean Eng.* **2015**, *105*, 54–63. [\[CrossRef\]](#)
13. Gafurov, S.A.; Klochkov, E.V. Autonomous unmanned underwater vehicles development tendencies. *Procedia Eng.* **2015**, *106*, 141–148. [\[CrossRef\]](#)
14. Qi, X.; Cai, Z.J. Three-dimensional formation control based on nonlinear small gain method for multiple underactuated underwater vehicles. *Ocean Eng.* **2018**, *151*, 105–114. [\[CrossRef\]](#)
15. Ramasamy, S.; Sabatini, R.; Gardi, A.; Liu, J. LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. *Aerosp. Sci. Technol.* **2016**, *55*, 344–358. [\[CrossRef\]](#)
16. Zhu, L.; Cheng, X.; Yuan, F.G. A 3D collision avoidance strategy for UAV with physical constraints. *Measurement* **2016**, *77*, 40–49. [\[CrossRef\]](#)
17. Chee, K.; Zhong, Z. Control, navigation and collision avoidance for an unmanned aerial vehicle. *Sens. Actuators A Phys.* **2013**, *190*, 66–76. [\[CrossRef\]](#)
18. Courbon, J.; Mezouar, Y.; Guénard, N.; Martinet, P. Vision-based navigation of unmanned aerial vehicles. *Control Eng. Pract.* **2010**, *18*, 789–799. [\[CrossRef\]](#)
19. Hull, D.G. *Fundamentals of Airplane Flight Mechanics*; Springer: Berlin/Heidelberg, Germany, 2007. Available online: https://scholar.google.co.uk/scholar?q=fundamental+of+airplane+flight+mechanics+and+David+Hull&btnG=&hl=en&as_sdt=0,5#0 (accessed on 1 September 2019).
20. Abbadì, A.; Matousek, R.; Jancik, S.; Roupec, J. Rapidly-exploring random trees: 3D planning. In Proceedings of the 18th International Conference on Soft Computing MENDEL, Brno, Czech Republic, 27–29 June 2012; pp. 594–599.
21. Aguilar, W.; Morales, S. 3D environment mapping using the Kinect V2 and path planning based on RRT algorithms. *Electronics* **2016**, *5*, 70. [\[CrossRef\]](#)
22. Aguilar, W.G.; Morales, S.; Ruiz, H.; Abad, V. RRT* GL based optimal path planning for real-time navigation of UAVs. In Proceedings of the International Work-Conference on Artificial Neural Networks, Cadiz, Spain, 14–16 June 2017; pp. 585–595.
23. Yao, P.; Wang, H.; Su, Z. Hybrid UAV path planning based on interfered fluid dynamical system and improved RRT. In Proceedings of the IECON 2015—41st Annual Conference of the IEEE Industrial Electronics, Yokohama, Japan, 9–12 November 2015; pp. 000829–000834.
24. Yan, F.; Liu, Y.S.; Xiao, J.Z. Path planning in complex 3D environments using a probabilistic roadmap method. *Int. J. Autom. Comput.* **2013**, *10*, 525–533. [\[CrossRef\]](#)
25. Yeh, H.Y.; Thomas, S.; Eppstein, D.; Amato, N.M. UOBPRM: A uniformly distributed obstacle-based PRM. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 2655–2662. [\[CrossRef\]](#)
26. Denny, J.; Amato, N.M. Toggle PRM: A coordinated mapping of C-free and C-obstacle in arbitrary dimension. In *Algorithmic Foundations of Robotics X*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 297–312. [\[CrossRef\]](#)
27. Li, Q.; Wei, C.; Wu, J.; Zhu, X. Improved PRM method of low altitude penetration trajectory planning for UAVs. In Proceedings of the IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014; pp. 2651–2656.

28. Ortiz-Arroyo, D. A hybrid 3D path planning method for UAVs. In Proceedings of the 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Cancun, Mexico, 23–25 November 2015; pp. 123–132.
29. Cheng, X.; Yang, C. Robot path planning based on adaptive isolation niche genetic algorithm. In Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application, Shanghai, China, 20–22 December 2008; Volume 2, pp. 151–154. [\[CrossRef\]](#)
30. Liang, Y.; Xu, L. Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm. In Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China, 12–14 June 2009; pp. 303–308. [\[CrossRef\]](#)
31. Liu, J.; Yang, J.; Liu, H.; Tian, X.; Gao, M. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [\[CrossRef\]](#)
32. Cao, H.; Sun, S.; Zhang, K.; Tang, Z. Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm. *Optik* **2016**, *127*, 9974–9983. [\[CrossRef\]](#)
33. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [\[CrossRef\]](#)
34. Alhanjouri, M.A.; Alfarra, B. Ant colony versus genetic algorithm based on travelling salesman problem. *Int. J. Comput. Technol. Appl.* **2011**, *2*, 570–578. [\[CrossRef\]](#)
35. Bakdi, A.; Hentout, A.; Boutami, H.; Maoudj, A.; Hachour, O.; Bouzouia, B. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robot. Auton. Syst.* **2017**, *89*, 95–109. [\[CrossRef\]](#)
36. Pandey, A.; Parhi, D.R. Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm. *Def. Technol.* **2017**, *13*, 47–58. [\[CrossRef\]](#)
37. Thanou, M.; Tzes, A. Distributed visibility-based coverage using a swarm of UAVs in known 3D-terrains. In Proceedings of the 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), Athens, Greece, 21–23 May 2014; pp. 425–428.
38. Qu, Y.; Zhang, Y.; Zhang, Y. Optimal flight path planning for UAVs in 3-D threat environment. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 149–155.
39. Fang, Z.; Luan, C.; Sun, Z. A 2d voronoi-based random tree for path planning in complicated 3d environments. In Proceedings of the International Conference on Intelligent Autonomous Systems, Shanghai, China, 3–7 July 2016; pp. 433–445.
40. Khuswendi, T.; Hindersah, H.; Adiprawita, W. Uav path planning using potential field and modified receding horizon a* 3d algorithm. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17–19 July 2011; pp. 1–6.
41. Chen, X.; Zhang, J. The three-dimension path planning of UAV based on improved artificial potential field in dynamic environment. In Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2013; Volume 2, pp. 144–147.
42. Rivera, D.M.; Prieto, F.A.; Ramírez, R. Trajectory planning for uavs in 3d environments using a moving band in potential sigmoid fields. In Proceedings of the 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, Fortaleza, Brazil, 16–19 October 2012; pp. 115–119.
43. Lifen, L.; Ruoxin, S.; Shuandao, L.; Jiang, W. Path planning for UAVS based on improved artificial potential field method through changing the repulsive potential function. In Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Nanjing, China, 12–14 August 2016; pp. 2011–2015.
44. Samaniego, F.; Sanchis, J.; García-Nieto, S.; Simarro, R. Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs. *Electronics* **2019**, *8*, 306. [\[CrossRef\]](#)
45. 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. [\[CrossRef\]](#)
46. Beard, R.W.; McLain, T.W. *Small Unmanned Aircraft: Theory and Practice*; Princeton University Press: Princeton, NJ, USA, 2012.
47. Fleury, S.; Soueres, P.; Laumond, J.P.; Chatila, R. Primitives for smoothing mobile robot trajectories. *IEEE Trans. Robot. Autom.* **1995**, *11*, 441–448. [\[CrossRef\]](#)

48. Vanegas, G.; Samaniego, F.; Girbes, V.; Armesto, L.; Garcia-Nieto, S. Smooth 3D path planning for non-holonomic UAVs. In Proceedings of the 2018 7th International Conference on Systems and Control (ICSC), Valencia, Spain, 24–26 October 2018; pp. 1–6. [[CrossRef](#)]
49. Brezak, M.; Petrović, I. Real-time approximation of clothoids with bounded error for path planning applications. *IEEE Trans. Robot.* **2013**, *30*, 507–515. [[CrossRef](#)]
50. Barsky, B.; DeRose, T. Geometric continuity of parametric curves: Three equivalent characterizations. *IEEE Comput. Graph. Appl.* **1989**, *9*, 60–69. [[CrossRef](#)]
51. Kim, H.; Kim, D.; Shin, J.U.; Kim, H.; Myung, H. Angular rate-constrained path planning algorithm for unmanned surface vehicles. *Ocean Eng.* **2014**, *84*, 37–44. [[CrossRef](#)]
52. Isaacs, J.; Hespanha, J. Dubins Traveling Salesman Problem with Neighborhoods: A Graph-Based Approach. *Algorithms* **2013**, *6*, 84–99. [[CrossRef](#)]
53. Masehian, E.; Kakahaji, H. NRR: A nonholonomic random replanner for navigation of car-like robots in unknown environments. *Robotica* **2014**, *32*, 1101–1123. [[CrossRef](#)]
54. Fraichard, T.; Scheuer, A. From Reeds and Shepp's to continuous-curvature paths. *IEEE Trans. Robot.* **2004**, *20*, 1025–1035. [[CrossRef](#)]
55. Pepy, R.; Lambert, A.; Mounier, H. Path Planning using a Dynamic Vehicle Model. In Proceedings of the 2006 2nd International Conference on Information & Communication Technologies, Damascus, Syria, 24–28 April 2006; Volume 1, pp. 781–786. [[CrossRef](#)]
56. Girbés, V.; Vanegas, G.; Armesto, L. Clothoid-Based Three-Dimensional Curve for Attitude Planning. *J. Guid. Control Dyn.* **2019**, *42*, 1–13. [[CrossRef](#)]
57. De Lorenzis, L.; Wriggers, P.; Hughes, T.J. Isogeometric contact: A review. *GAMM-Mitteilungen* **2014**, *37*, 85–123. [[CrossRef](#)]
58. Pigounakis, K.G.; Sapidis, N.S.; D, K.P. Fairing spatial B-spline curves. *J. Ship Res.* **1996**, *40*, 35.
59. Pérez, L.H.; Aguilar, M.C.M.; Sánchez, N.M.; Montesinos, A.F. Path Planning Based on Parametric Curves. In *Advanced Path Planning for Mobile Entities*; InTech: London, UK, 2018. [[CrossRef](#)]
60. Huh, U.Y.; Chang, S.R. A G^2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 25. [[CrossRef](#)]
61. Chang, S.R.; Huh, U.Y. A Collision-Free G^2 Continuous Path-Smoothing Algorithm Using Quadratic Polynomial Interpolation. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 194. [[CrossRef](#)]
62. Herrero, J.M.; Martínez, M.; Sanchis, J.; Blasco, X. Well-Distributed Pareto Front by Using the ϵ^{λ} -MOGA Evolutionary Algorithm. In *Computational and Ambient Intelligence*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 292–299. [[CrossRef](#)]
63. Jin, Y.; Sendhoff, B. Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2008**, *38*, 397–415. [[CrossRef](#)]
64. Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*; Elsevier: Amsterdam, The Netherlands, 2014.
65. Velasco-Carrau, J.; García-Nieto, S.; Salcedo, J.V.; Bishop, R.H. Multi-Objective Optimization for Wind Estimation and Aircraft Model Identification. *J. Guid. Control. Dyn.* **2016**, *39*, 372–389. [[CrossRef](#)]
66. Honig, E.; Schucking, E.L.; Vishveshwara, C.V. Motion of charged particles in homogeneous electromagnetic fields. *J. Math. Phys.* **1974**, *15*, 774–781. [[CrossRef](#)]
67. Iyer, B.R.; Vishveshwara, C.V. The Frenet-Serret formalism and black holes in higher dimensions. *Class. Quantum Gravity* **1988**, *5*, 961–970. [[CrossRef](#)]
68. Abbena, E.; Salamon, S.; Gray, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2017.
69. Lazard, S.; Reif, J.; Wang, H. The Complexity of the Two Dimensional Curvature-Constrained Shortest-Path Problem. In Proceedings of the Third International Workshop on the Algorithmic Foundations of Robotics, Houston, TX, USA, 5–7 March 1998; pp. 49–57.
70. Coello Coello, C.A.; Pulido, G.T.; Montes, E.M. Current and Future Research Trends in Evolutionary Multiobjective Optimization. In *Information Processing with Evolutionary Algorithms*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 213–231. [[CrossRef](#)]
71. Laumanns, M.; Thiele, L.; Deb, K.; Zitzler, E. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evol. Comput.* **2002**, *10*, 263–282. [[CrossRef](#)]

72. Blasco, X.; Herrero, J.; Sanchis, J.; Martínez, M. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Inf. Sci.* **2008**, *178*, 3908–3924. [[CrossRef](#)]
73. McInnes, C. Satellite attitude slew manoeuvres using inverse control. *Aeronaut. J.* **1998**, *102*, 259–266.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).