

Feasible path planning for fixed-wing UAVs using seventh order Bézier curves

Armando Alves Neto · Douglas G. Macharet ·
Mario F. M. Campos

Received: 15 May 2012 / Accepted: 16 October 2012 / Published online: 31 October 2012
© The Brazilian Computer Society 2012

Abstract This study presents a novel methodology for generating smooth feasible paths for autonomous aerial vehicles in the three-dimensional space based on a variation of the Spatial Quintic Pythagorean Hodographs curves. Generated paths must satisfy three main constraints: (i) maximum curvature, (ii) maximum torsion and (iii) maximum climb (or dive) angle. A given path is considered to be feasible if the main kinematic constraints of the vehicle are not violated, which is accomplished in our approach by connecting different waypoints with seventh order Bézier curves. This also indirectly insures the smoothness of the vehicle's acceleration profile between two consecutive points of the curve and of the entire path by controlling the curvature values at the extreme points of each composing Bézier curve segment. The computation of the Pythagorean Hodograph is cast as an optimization problem, for which we provide an algorithm with fast convergence to the final result. The proposed methodology is applicable to vehicles in three-dimensional environments, which can be modeled presuming the imposed constraints. Our methodology is validated in simulation with real parameters and simulated flight data of a small autonomous aerial vehicle.

Keywords Path planning · Continuous curvature profile · Fixed-wing Unmanned Aerial Vehicle System

1 Introduction

Path planning is a fundamental task for any kind of autonomous mobile robot. Even though it might be possible for such robots to traverse their environments solely in a reactive way, the competence of planning and computing paths is an important feature for a large number of vehicles and a great variety of tasks.

In spite of its relevance, there are not that many works dealing with vehicles that move in three-dimensional space. Furthermore, and less obvious, the increased freedom provided by a less restrictive environment poses many new challenges. Current problems, such as path planning for multiple Unmanned Aerial Vehicles (UAVs) and Autonomous Underwater Vehicles (AUVs), still call for better solutions.

The interest and research in UAVs have been increasingly growing, specially due to the decrease in cost, weight, size and performance of actuators, sensors and processors. As far as the capacity of covering a broad set of relevant applications is concerned, UAVs clearly have their niche of applications, which cannot be fulfilled by other types of mobile robots. One of their main advantages is in several types of monitoring and surveillance tasks, where they are able to navigate over large areas obviously faster than land vehicles, with a privileged view from above.

As their availability increases, so does the possibility of having multiple of such vehicles traversing a given volume of the air space. Therefore, there is a growing need to study and develop techniques for the generation of safe and feasible paths considering specific constraints of the different types of aerial robots. Robotics literature describes several such algorithms. However, one fundamental feature of a path planning algorithm is to insure that the vehicle will be able to execute the generated path, which means that limitations on vehicle movements must be obeyed (i.e., nonholonomic

A. A. Neto (✉) · D. G. Macharet · M. F. M. Campos
Departamento de Ciência da Computação,
Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
e-mail: aaneto@dcc.ufmg.br

D. G. Macharet
e-mail: doug@dcc.ufmg.br

M. F. M. Campos
e-mail: mario@dcc.ufmg.br

constraints). For example, curvature radius is one such restriction imposed on paths generated for typical Ackerman steering vehicles, since the sliding constraint of wheels dictates that the component of the wheel's motion orthogonal to the wheel plane must be zero.

1.1 Unmanned Aerial Vehicles

UAVs can be divided into at least three categories: rotary-wing aircrafts (e.g., helicopters and quadrotors), aerostatic aircrafts (such as airships and hot air balloons) and fixed-wing aircrafts (airplanes). The technique described in this text will have its focus mostly in a fixed-wing UAV, however, without the loss of generality, it can be applied to other types of vehicles (even ground and underwater robots).

Mobile robots frequently present some types of motion constraints that must be solved by path planning algorithms. Fixed-wing aerial vehicles, for example, present dynamic behaviors where position and orientation variables are completely interdependent, which in turn imposes several non-holonomic constraints to the system. These constraints are embedded, for example, in the maximum values of lateral acceleration that can be imposed by those vehicles, which can be translated by the smallest curvature radius that the vehicle can perform in space. Fixed-wing aircrafts present also other limitations on their mobility such as maximum climb or dive angles and minimum speed.

One of the fundamental motion constraints of a vehicle moving in three-dimensional space is the climb (or dive) angle. This basically refers to the rate of change in altitude, which may be severely restricted for some types of aerial vehicles. There are aircrafts, for example, that have a very small angle of attack¹ which is often limited by the control action of the guidance system. Other examples are underwater vehicles for which the climb angle is confined to short boundaries subtended by the configuration of their actuators and control surfaces.

In this study, a novel path planning methodology based on a variation of the Spatial Quintic Pythagorean Hodograph curves is proposed. The method takes into account three major motion constraints of a fixed-wing UAV in the three-dimensional space: maximum curvature, maximum torsion and maximum climb angle, but with special emphasis on the cases of limited climb (or dive) rates.

For the special case of fixed-wing aircrafts, the main goal is to guarantee stall-free maneuvers. The general idea behind the method is to model the path as a seventh order spatial Bézier curve which is iteratively computed. This path satisfies all the required constraints and, at the same time, with

a satisfactory length. It also guarantees smoothness of the acceleration profile of the entire path by controlling the curvature values of the curves that composes it. An algorithm with fast convergence to the final result is also described and evaluated, since the Pythagorean Hodograph computation is cast as an optimization problem.

Section 2 constitutes an overview of the related work in the literature, where the theoretical foundations of this work is discussed. Then, Sect. 3 presents the methodology, where the problem is formally posed along with the improvements, and the proposed optimization algorithm to calculate the paths. Results obtained for a virtual UAV and for a simulated model of an actual UAV called AqVS developed at Universidade Federal de Minas Gerais are presented and discussed in Sect. 4. Finally, Sect. 5 concludes the article and points to future research directions.

2 Related work

One of the most important factors for path planning is to produce paths that are feasible of being executed by the targeted vehicle. This means that during path generation, the movement restrictions of the vehicle must be considered (e.g., non-holonomic constraints). This problem has been thoroughly studied, and the literature available in the area abounds specially for manipulators and two-dimensional mobile robots [18].

Some approaches of single vehicle path planning in less restrictive environments can be found in the literature [17, 23]. Voronoi diagrams [5, 7] and Vector Field [13, 19] are widely used techniques to generate paths for aerial robots with such constraints. However, very few works deal with the three-dimensional case.

Rapidly-exploring Random Trees (RRTs) are also widely used, especially for solving the path problem for nonholonomic vehicles in broader scenarios. In [6], the authors present trajectory planning for both an automobile and a spacecraft. In the latter example, even though an obstacle-free environment is considered, the focus remains on the motion constraints that need to be satisfied for a safe entry of the spacecraft in Earth's atmosphere. Other works like [14] use this technique to generate nominal paths for miniature aerial vehicles. The authors present an algorithm for terrain avoidance for the air platform BYU/MAV, which, among other things, enables the vehicle to fly through canyons and urban environments. And although these studies consider some restrictions inherent to aerial robots, none of them take into account important constraints such as climbing rate and spatial torsion on fixed-wing vehicles.

There are also some works that only deal with the generation of safe paths for vehicles assuming an obstacle-free environment. Among those is the work of [22], which presents

¹ Defined as the angle that the chord of the wing, viewed laterally, makes with respect to the wind. This is also another way to define the climb angle.

one of the first methods based on the Pythagorean Hodograph (a special type of Bézier curve) for the path planning for UAVs. The author discusses numerous advantages of using such a curve in the modeling of paths for vehicles with nonholonomic constraints.

In this study, we use a special technique to compute feasible paths, called PH Interpolation. The PH provides several properties that can be considered advantageous in the path planning problem such as uniformity in the parametric distribution, continuous parametric speed and capability to admit real-time interpolation. More details about these features will be detailed later.

The PH curves were presented for the first time for the two-dimensional case by Farouki and Sakkalis [12]. Initially, only fifth order curves were considered since they are much simpler. A Hermite Interpolation algorithm was proposed in [11], where the authors demonstrated that there are four possible solutions for the curve in \mathbb{R}^2 space. The chosen solution is the one that minimizes the cost function (bending energy function), based on the integral of the modulus of the curvature function (torsion in their case is assumed to be null).

The three-dimensional case was initially presented by Farouki et al. [10]. In [9], the quaternion representation was used to deal with Hermite Interpolation issues, and the authors claim that the infinite cardinality of the set of solutions for the problem is due to an underdetermined system of equations that are needed to compute the final curve.

In order to significantly reduce the solution space, the authors suggest assigning a small set of values to the unknown variables, reducing the number of possible solutions. The best solution is obtained from the minimization of the cost function defined in [10], that is based on the integral of the sum of the curvature and torsion functions modulus of each curve.

Two of the first works involving the use of PH curve for robot path planning are Shanmugavel et al. [22] for the two-dimensional case and in [21] for the three-dimensional case. In order to guarantee that the PH did not violate the kinematic constraints of the vehicle, the authors proposed a modification where the PH curve was computed iteratively. For every step of the algorithm, gain values were increased until $\mathbf{r}(t)$ becomes realizable. However, only the curvature and torsion constraints were considered. As it will be shown later, for vehicles with bounded climb (or dive) angle values, it is not possible to minimize the problem for the spatial cost functions.

In [2], the authors present an improved cost function based on [10] that considers all three constraints: curvature, torsion and climb angle. This study represents an improvement with respect to [21] since it considers the limitation in the climb angle of fixed-wing UAVs. A new cost function, computationally less expensive, was presented in [1], where only the climb angle constraint is used in the minimization function.

Other works, such as [4,22], present the application of PH curves in the path planning and trajectory planning problem for multiple vehicles. Both, however, deal only with the two-dimensional case, without taking into account torsion and climb angle constraints.

The adaptation of the fifth order PH curve was firstly proposed by Alves Neto et al. [3]. The main objective was to create Bézier curves with continuous curvature profiles among several waypoints in \mathbb{R}^2 . That work presents a methodology based on a variation of the RRTs that generate feasible trajectories for autonomous vehicles with nonholonomic constraints in environments with obstacles.

In this study, we expand to \mathbb{R}^3 the methodology described in [3], keeping the continuity in the curvature profile and taking into account all three constraints, namely curvature, torsion and climb (or dive) angle. Our method also allows the connection of two different curves without violating the constraints of the whole path. We propose a new cost function that will include just the climb angle constraint, indirectly ensuring the others constraints, that effectively reduce the computational cost of the method. We also present an optimization function based on the bending energy function in order to minimize the total path length.

3 Methodology

In this section, we initially present the theoretical formalization of our problem. We describe mathematically the robot's constraints and the Pythagorean Hodograph formulation. We then provide the foundations for calculating the PH curves for which the union of several paths keeps the continuous curvature profile in the final plan. Finally, we show an optimization problem formulation that will allow us to generate short paths that are feasible by the robot.

3.1 Problem formalization

We assumed an obstacle-free environment, where the only restrictions for navigation are imposed by its own kinematic constraints. A pair of waypoints, $\langle p_i, p_f \rangle$, mark the initial and final poses, respectively, which define the position and (partially) the orientation of the robot at the extreme points of a single path. This path may be defined mathematically as a parametric curve $\mathbf{r}(t)$ in \mathbb{R}^3 , where t is a parameter that continuously varies in \mathbb{R} . In this manner, planning a path between two waypoints can be formally posed as:

$$\begin{aligned} p_i(x_i, y_i, z_i, \psi_i, \theta_i) &= \mathbf{r}(t_i), \\ p_f(x_f, y_f, z_f, \psi_f, \theta_f) &= \mathbf{r}(t_f), \end{aligned} \quad (1)$$

where t_i and t_f are the initial and final values, respectively, for the curve parameter t .

Each waypoint p is described by three-position (x, y, z) and two-orientation (ψ, θ) variables. The variable ψ is an angle that describes the waypoint orientation parallel to the XY plane in relation to the \mathbf{X} axis. We define θ as the waypoint orientation measured in relation to the \mathbf{X} axis and parallel to XZ plane. Here, we must say that the roll angle (ϕ in conventional aeronautical notation) is not considered, since it has no physical sense regarding the curve in the 3D space. In other words, the curve represents a trajectory to be followed, not the actual trajectory being executed by the robot.

Now, given a sequence of P waypoints, $\mathcal{P} = \langle p_1, \dots, p_P \rangle$, that must be reached by the robot, a total feasible path \mathcal{R} will be the union of each $\mathbf{r}(t)$ calculated between each sequence pair in \mathcal{P} . This is formalized in Eq. (2):

$$\mathcal{R} = \left\{ \bigcup_{k=1}^{P-1} \mathbf{r}_k \in \mathbb{R} : \mathbf{r}_k \text{ is a single feasible path} \right\}. \quad (2)$$

3.1.1 Constraints

In order to be considered a feasible path for a given robot, the curve $\mathbf{r}(t)$ must respect the kinematic constraints of the vehicle. The three motion constraints mentioned before are the maximum curvature (κ_{\max}), the maximum torsion (τ_{\max}) and the maximum climb (or dive) angle (θ_{\max}) that are realizable by the robot in the three-dimensional space. It is possible to completely define a curve in \mathbb{R}^3 as a function of curvature and torsion only [16].

As far as the underlying physics of the system is concerned, the curvature may be defined as a quantity that is directly proportional to the lateral acceleration of the robot in the space. The value of κ_{\max} is inversely proportional to the minimum curvature radius (ρ_{\min}) of the curve that the vehicle is capable of executing, which is also proportional to the maximum lateral acceleration of the vehicle.

The *curvature function* of a curve in the n -dimensional space is given by Eq. (3):

$$\kappa(t) = \frac{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|}{|\dot{\mathbf{r}}(t)|^3}. \quad (3)$$

The torsion of the curve is directly proportional to the angular moment (roll moment) of the robot, which is also physically limited. Thus, the value of τ_{\max} is given as a function of the minimum torsion radius (σ_{\min}) that the robot can describe in the space. The torsion of a curve can be computed by Eq. (4):

$$\tau(t) = \frac{[\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)] \cdot \ddot{\mathbf{r}}(t)}{|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)|^2}. \quad (4)$$

Finally, the climb (or dive) angle is proportional to the ascent (or descent) rate of the robot in the three-dimensional space. In other words, it captures the variation of the altitude (z) throughout the path. For vehicles with limited values of

inclination angle, such as fixed-wing aircrafts, this is a key variable. The value of θ_{\max} may depend on many factors, as the translation speed and the spatial orientation of the robot. The climb angle function of a three-dimensional parametric curve is given by Eq. (5):

$$\theta(t) = \tan^{-1} \left(\frac{\dot{z}(t)}{\sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}} \right). \quad (5)$$

It is possible to show that this function is mathematically limited to the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. The same is true for the value of θ_{\max} .

Finally, the path $\mathbf{r}(t)$ is valid for a vehicle if the modulus of the curvature, torsion and climb angle functions are smaller than the vehicle’s absolute maximum values, as stated in Eq. (6):

$$\mathbf{r}(t) = \{ \mathbf{r} \in \mathbb{R} : |\kappa(t)| < \kappa_{\max}, |\tau(t)| < \tau_{\max}, |\theta(t)| < \theta_{\max} \}. \quad (6)$$

It is important to note that the above maximum values result from the dynamic constraints of the vehicle. Therefore, since our kinematic approach guarantees continuity of the curvature, torsion, and climb angle functions, by respecting the maximum variable values, indirectly the vehicle dynamics are accounted for. Discontinuities in the curvature function, for example, may induce infinite variations of lateral acceleration, which of course, are not realizable. The same is true for torsional values. In the case of the climb angle function, the lack of continuity implies in the tangential discontinuity of the curve itself.

In the next section, we detailed the necessary steps to realize the union in Eq. (2) in order to guarantee curvature and climb angle continuity for \mathcal{R} . As expressed by Eq. 3, the curve produced by the path planning algorithm must be continuously derivable, and it must also be second order differentiable (C^2).

3.1.2 Spatial Pythagorean Hodograph curves

Spatial Pythagorean Hodograph curves are a special kind of parametric polynomial curves defined in \mathbb{R}^3 . They are represented in general as:

$$\mathbf{r}(t) = [x(t), y(t), z(t)],$$

and their derivatives (Hodograph components) satisfy the Pythagorean condition presented in Eq. (7):

$$\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2 = h(t)^2, \quad (7)$$

for some polynomial $h(t)$. This means that the parametric “speed”, $\dot{s}(t)$, of the curve can be described by a polynomial, making it possible to compute the length of the path, s , as shown in Eq. (8):

$$s = \int_{t_i}^{t_f} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} dt = \int_0^1 |h(t)| dt. \tag{8}$$

Spatial PH curves are further shaped as n -th order Bézier curves presented in Eq. (9):

$$\mathbf{r}(t) = \sum_{k=0}^n \mathbf{p}_k \binom{n}{k} (1-t)^{n-k} t^k; \quad t \in [0, 1] \tag{9}$$

where $\mathbf{p}_k = [x_k, y_k, z_k]$ is the k -th control point of the Bézier function. N -th order Bézier curves always have $n + 1$ control points that define the curve shape according to their spatial position [16].

The path planning problem is then reduced to find a solution to an Hermite Interpolation problem. One important advantage of using this model is that the resulting curve is infinitely continuous, so that the curvature, torsion and inclination functions are always smooth.

To guarantee that the PH does not violate the kinematic constraints of the vehicle, Shanmugavel et al. [21] propose a modification to iteratively compute the PH. It uses gain factors to modify the position of some of the control points in every step of an iterative algorithm, until $\mathbf{r}(t)$ becomes realizable. We will discuss these characteristics later in this article. However, Shanmugavel’s method only consider the curvature and torsion constraints. As it will be shown later, for vehicles with bounded values of climb (or dive) angle, it is not possible to minimize the problem for that spatial cost function. Therefore, we propose a new cost function that incorporates all three variables along with limitations on their maximum values. The θ_{\max} constraint will be included in the PH computation, which will then generate feasible paths for robots in three-dimensional space, under all the aforementioned constraints.

3.2 Smooth path calculation

In this section, we present a method to calculate Bézier curves that fulfills the three constraint conditions mentioned before. We begin by mentioning an interesting characteristic of Bézier curves. They are infinitely differentiable functions, which mean that one curve has continuous curvature profiles. However, if we connect two or more curves to compose a path \mathcal{R} among several waypoints, it is possible that discontinuity in accelerations will show up at the connecting points, violating the C^2 condition. Then, in order to keep the continuity, we must calculate Bézier curves whose curvature values at extreme points are identical.

According to [20], the initial curvature of a Bézier function can be mathematically determined by Eq. (10):

$$\kappa(t)|_{t=0} = \frac{(n-1) |\mathbf{p}_2 - \mathbf{p}_1|}{n |\mathbf{p}_1 - \mathbf{p}_0|} \sin \sigma_i, \tag{10}$$

where n is the Bézier order and σ_i represents the angle between the vectors $\overrightarrow{\mathbf{p}_0\mathbf{p}_1}$ and $\overrightarrow{\mathbf{p}_1\mathbf{p}_2}$.

It is possible to guarantee a null curvature value at the initial point by forcing collinearity of these vectors. The same idea can be extended to the final curvature value, as shown in Eq. (11):

$$\kappa(t)|_{t=1} = \frac{(n-1) |\mathbf{p}_7 - \mathbf{p}_6|}{n |\mathbf{p}_6 - \mathbf{p}_5|} \sin \sigma_f, \tag{11}$$

where σ_f is the angle between the vectors $\overrightarrow{\mathbf{p}_5\mathbf{p}_6}$ and $\overrightarrow{\mathbf{p}_6\mathbf{p}_7}$.

Then, we will always have zero curvature value at the extreme points of each Bézier, making its composition with others curves (\mathcal{R}) a C^2 function, as required. The main reason which led us to choose a seventh order function is now clear. We must keep the three initial and three final control points of the Bézier aligned. So we need at least other two points to create a function with at least one point of inflexion, in order to make a more flexible path.

With this in mind, we use an adaptation of the Pythagorean Hodograph formalism to compute these eight control points. Therefore, we may use the set of Eq. 12, which depends only on the initial and final configuration points, and represent an Hermite Interpolation system.

$$\mathbf{p}_0 = [x_i, y_i, z_i], \tag{12}$$

$$\mathbf{p}_1 = \mathbf{p}_0 + c_0 \mathcal{A}_0 \mathbf{i} \mathcal{A}_0^*, \tag{13}$$

$$\mathbf{p}_2 = \mathbf{p}_1 + c_0 \mathcal{A}_0 \mathbf{i} \mathcal{A}_0^*, \tag{14}$$

$$\mathbf{p}_3 = \mathbf{p}_2 + \frac{c_0}{2} (\mathcal{A}_0 \mathbf{i} \mathcal{A}_1^* + \mathcal{A}_1 \mathbf{i} \mathcal{A}_0^*), \tag{15}$$

$$\mathbf{p}_4 = \mathbf{p}_3 - \frac{c_7}{2} (\mathcal{A}_2 \mathbf{i} \mathcal{A}_1^* + \mathcal{A}_1 \mathbf{i} \mathcal{A}_2^*), \tag{16}$$

$$\mathbf{p}_5 = \mathbf{p}_6 - c_7 \mathcal{A}_2 \mathbf{i} \mathcal{A}_2^*, \tag{17}$$

$$\mathbf{p}_6 = \mathbf{p}_7 - c_7 \mathcal{A}_2 \mathbf{i} \mathcal{A}_2^*, \tag{18}$$

$$\mathbf{p}_7 = [x_f, y_f, z_f]. \tag{19}$$

where c_0 and c_7 are gain factors that determine the magnitude of the vectors and have unit values for PH with no constraints. The above set of equations was derived from [9], and according to the authors, it represents an underdetermined system. In other words, there are infinitely many spatial PH curves that interpolate the initial and final waypoints.

The PH provides several properties that can be considered advantageous in the path planning problem, of which, the most relevant for this study are: (i) the uniform distribution of points along the curve, which contributes to the smoothness of the path; (ii) the parametric speed (first order derivative) that provides a continuous velocity profile; and finally (iii) the capability to admit real-time interpolator algorithms for computer numerical control.

Once the Bézier points are computed, we must guarantee that the generated curve does not violate the kinematic

constraints of the vehicle, according to the constraints conditions of Eq. (6). Hence, we must determine the values of c_0 and c_7 in Eq. (12). As there is no closed form solution, these values will be estimated, in the next section, as the solution to an optimization problem.

The problem is initially approached by assuming that the extreme points of the curve (\mathbf{p}_0 and \mathbf{p}_7) are directly determined by the initial and final poses p_i and p_f , respectively. All the remaining points will depend on these pose vectors and on the c_0 and c_7 gains.

In our development, we use the quaternion formulation, which allows for a more compact and elegant representation of the system, as well as providing a clearer geometric view of the solution. The parameters \mathcal{A}_k , with $k = \{1, 2, 3\}$, are quaternions described by Eq. (20),

$$\mathcal{A}_k = a + a_x \mathbf{i} + a_y \mathbf{j} + a_z \mathbf{k} = \begin{bmatrix} a \\ a_x \\ a_y \\ a_z \end{bmatrix}, \tag{20}$$

where a corresponds to the scalar part and $[a_x, a_y, a_z]$ to the vector elements of the quaternion. The conjugate of \mathcal{A}_k is denoted as \mathcal{A}_k^* . A set of equations of this type presents a solution \mathbf{c} which also is a pure vector quaternion, as shown by Eq. (21):

$$\mathcal{A}_i \mathcal{A}_f^* = \mathbf{c}, \tag{21}$$

where \mathbf{i} is a pure vector quaternion since its scalar part is null and $\mathbf{i}^2 = -1$.

Hence, a system which is composed of three equations and four unknown variables can be formed, leaving one degree of freedom yet to be determined. With this, the solution for every \mathcal{A}_k is parameterized by an angular variable ϕ_k (for a detailed explanation, the reader is referred to [10]). Deriving Eq. (21) with respect to \mathcal{A}_0 , one obtains:

$$\mathcal{A}_0(\phi_0) = \sqrt{\frac{|\mathbf{d}_i|}{2}(1 + \lambda_i)} \begin{bmatrix} -\sin(\phi_0) \\ \cos(\phi_0) \\ \frac{\mu_i \cos(\phi_0) + \nu_i \sin(\phi_0)}{1 + \lambda_i} \\ \frac{\nu_i \cos(\phi_0) - \mu_i \sin(\phi_0)}{1 + \lambda_i} \end{bmatrix}, \tag{22}$$

where \mathbf{d}_i is the vector direction of the pose p_i , computed as

$$\mathbf{d}_i = c_0[\cos(\psi_i) \cos(\theta_i), \sin(\psi_i) \cos(\theta_i), \sin(\theta_i)]$$

and

$$[\lambda_i, \mu_i, \nu_i] = \frac{\mathbf{d}_i}{|\mathbf{d}_i|}.$$

Equivalently, \mathcal{A}_2 can be expressed as:

$$\mathcal{A}_2(\phi_2) = \sqrt{\frac{|\mathbf{d}_f|}{2}(1 + \lambda_f)} \begin{bmatrix} -\sin(\phi_2) \\ \cos(\phi_2) \\ \frac{\mu_f \cos(\phi_2) + \nu_f \sin(\phi_2)}{1 + \lambda_f} \\ \frac{\nu_f \cos(\phi_2) - \mu_f \sin(\phi_2)}{1 + \lambda_f} \end{bmatrix}, \tag{23}$$

where \mathbf{d}_f is the vector direction of the pose p_f , given by:

$$\mathbf{d}_f = c_7[\cos(\psi_f) \cos(\theta_f), \sin(\psi_f) \cos(\theta_f), \sin(\theta_f)]$$

and

$$[\lambda_f, \mu_f, \nu_f] = \frac{\mathbf{d}_f}{|\mathbf{d}_f|}.$$

Once the values of \mathcal{A}_0 and \mathcal{A}_2 are determined by setting the values of ϕ_0 and ϕ_2 , \mathcal{A}_1 can be defined as a function of a third angle variable ϕ_1 :

$$\mathcal{A}_1(\phi_1) = -\frac{3}{4}(\mathcal{A}_0 + \mathcal{A}_2) + \sqrt{\frac{|\mathbf{c}|}{2}(1 + \lambda)} \mathcal{A}_c(\phi_1), \tag{24}$$

where

$$\mathcal{A}_c(\phi_1) = \begin{bmatrix} -\sin(\phi_1) \\ \cos(\phi_1) \\ \frac{\mu \cos(\phi_1) + \nu \sin(\phi_1)}{1 + \lambda} \\ \frac{\nu \cos(\phi_1) - \mu \sin(\phi_1)}{1 + \lambda} \end{bmatrix},$$

\mathbf{c} is a pure vector quaternion given by

$$\mathbf{c} = 120(\mathbf{p}_f - \mathbf{p}_i) - 15(\mathbf{d}_i + \mathbf{d}_f) + 5(\mathcal{A}_0 \mathbf{i} \mathcal{A}_2^* + \mathcal{A}_2 \mathbf{i} \mathcal{A}_0^*) \tag{25}$$

and

$$[\lambda, \mu, \nu] = \frac{\mathbf{c}}{|\mathbf{c}|}.$$

3.3 Iterative algorithm

To compute the spatial PH, it is necessary to determine the values of the angles ϕ_0, ϕ_1, ϕ_2 and the gains c_0 and c_7 . The angular variables are defined for the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. In [10], authors claim (empirically) that it is possible to represent the solution space in extensive form enough, by the quantization of the values of each of these angles according to:

$$\phi_k = \left[-\frac{\pi}{2}, -\frac{\pi}{4}, 0, \frac{\pi}{4}, \frac{\pi}{2} \right].$$

The combinatorial arrangement of these five values for each ϕ_k leads to a total of 125 solutions for the PH curve. The authors also argue that for most cases, ϕ_1 may be equal to $-\frac{\pi}{2}$, without the loss of generality, reducing the number of solutions to 25. This value, generally, makes the final result

Table 1 Initial and final waypoints for the virtual UAV

Waypoint	x(m)	y(m)	z(m)	ψ (rad)	θ (rad)
p_i	0	0	0	$-\frac{\pi}{2}$	$\frac{\pi}{6}$
p_f	50	20	50	$-\frac{\pi}{2}$	0

to minimize the variation under the **Z** axis, what also reduces the climb angle function.

Each of these solutions has different curvature, torsion and climb angle functions. The best solution is the one that minimizes the cost function of the path, or the elastic bending energy function [8]:

$$\mathcal{E} = \int_0^1 \omega(t)^2 |\dot{\mathbf{r}}(t)| dt, \tag{26}$$

where $\omega(t)$ is the total curvature of the spatial PH, which is given by:

$$\omega(t) = \sqrt{\kappa(t)^2 + \tau(t)^2}. \tag{27}$$

This solution, however, is not satisfactory for the problem considered here, since it does not take into account the climb angle function in the energy computation. As it will be shown later, the PH curves that minimize the cost function, may present climb (or dive) angles unattainable for a given robot.

To tackle this issue, the following elastic bending energy function is proposed in [2]:

$$\omega(t) = \sqrt{\left(\frac{\kappa(t)}{\kappa_{\max}}\right)^2 + \left(\frac{\tau(t)}{\tau_{\max}}\right)^2 + \left(\frac{\theta(t)}{\theta_{\max}}\right)^2}. \tag{28}$$

Besides minimizing the rate of growth of the vehicle’s climbing action in three-dimensional space, this function still

takes into account the three aforementioned constraints and their maximum values in a normalized form. The problem with this is the increase in the computational complexity in relation to the previous equation.

We show in [1] that a more satisfactory result can be obtained using the following cost function:

$$\omega(t) = |\theta(t)|. \tag{29}$$

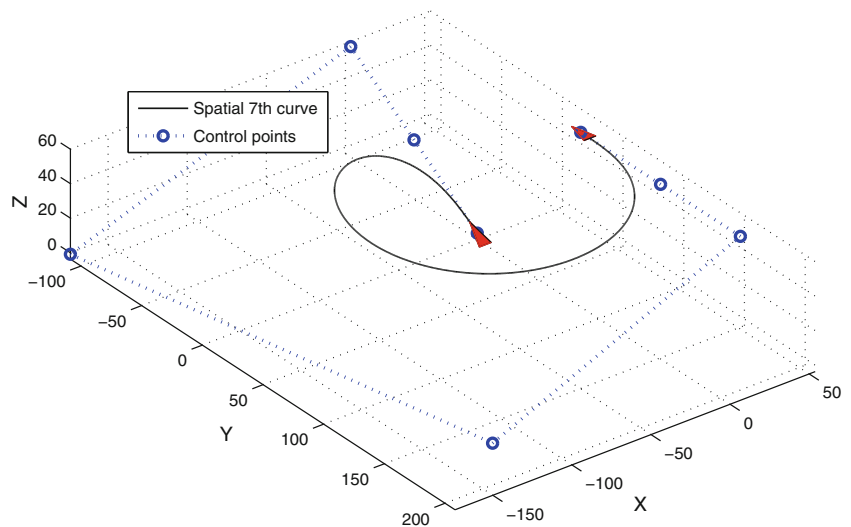
Instead of considering the curvature and torsion profiles, this new function takes into account only the aforementioned climb constraint of the vehicle, which minimizes the cost of algorithm and contributes to the smoothing of the climb rate of the curve. It is possible to demonstrate that, by minimizing $\theta(t)$, which means to reduce the $\dot{z}(t)$ over the *XY* plane, the curvature and, mainly, the torsion function also decrease.

We verified experimentally that, in most cases, the smallest values of \mathcal{E} were obtained when $\phi_1 = -\frac{\pi}{2}$ (as seen in the previous case), and when the difference between ϕ_0 and ϕ_2 was the largest possible. Therefore, taking ϕ_0 in the ϕ_k interval and $\phi_2 = -\phi_0$, makes it possible to reduce the number of solutions to only five.

Finally, the values of c_0 and c_7 (the gains for which the spatial PH fulfills the requirements described in Eq. (6) still remains to be determined. This naturally leads to an optimization approach, since there is no closed solution for this problem. Increasing these gains tends to minimize the function $\omega(t)$ as a whole, which is indirectly linked to the inverse of the parametric “speed” of the curve. However, that produces, as a consequence, an increase in path length. Thus, the ideal values of c_0 and c_7 are those which produce a feasible curve for a given robot, but also minimizes the length of *s*.

In the first step of the algorithm, the values of these gain factors are set to one. In each iteration of the algorithm, the

Fig. 1 Single path curve between $p_i = [0, 0, 0, -\frac{\pi}{2}, \frac{\pi}{6}]$ and $p_f = [50, 20, 50, -\frac{\pi}{2}, 0]$ for a virtual UAV with $\rho_{\min} = 10$ m, $\sigma_{\min} = 100$ m and $\theta_{\max} = \frac{\pi}{6}$ rad



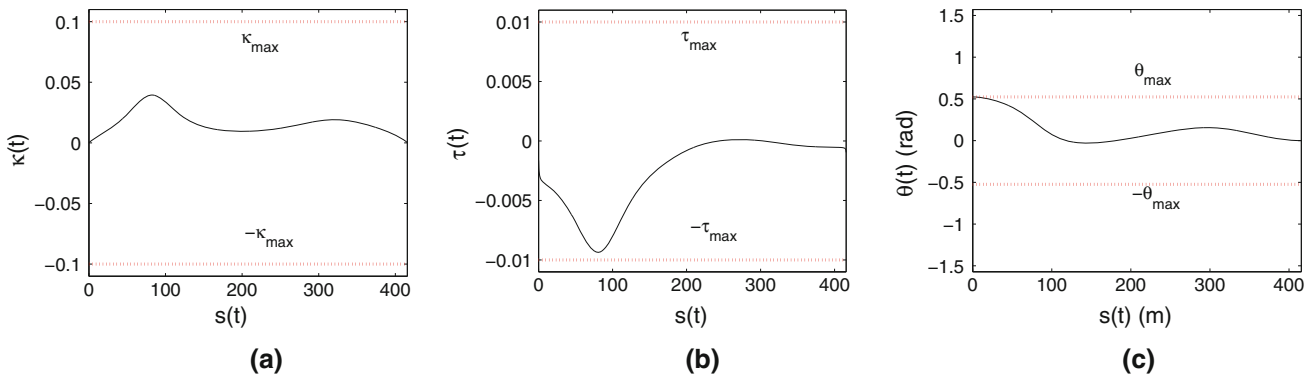


Fig. 2 Constraint functions of **a** curvature, **b** torsion and **c** climb angle of the single path

Table 2 Waypoint set $\langle p_a, p_h \rangle$ for the virtual UAV

Waypoint	$x(m)$	$y(m)$	$z(m)$	ψ (rad)	θ (rad)
p_a	0	0	0	0	0
p_b	200	0	100	$\frac{\pi}{2}$	$\frac{\pi}{6}$
p_c	500	500	400	$\frac{\pi}{2}$	0
p_d	500	1,000	200	$-\frac{\pi}{2}$	0
p_e	500	-500	500	π	$-\frac{\pi}{6}$
p_f	-300	200	300	$\frac{\pi}{4}$	0
p_g	0	300	200	π	0
p_h	-500	1,000	100	π	0

where

$$\mathcal{E}_0 = \int_0^{0.5} \omega(t)^2 |\dot{\mathbf{r}}(t)| dt$$

and

$$\mathcal{E}_7 = \int_{0.5}^1 \omega(t)^2 |\dot{\mathbf{r}}(t)| dt.$$

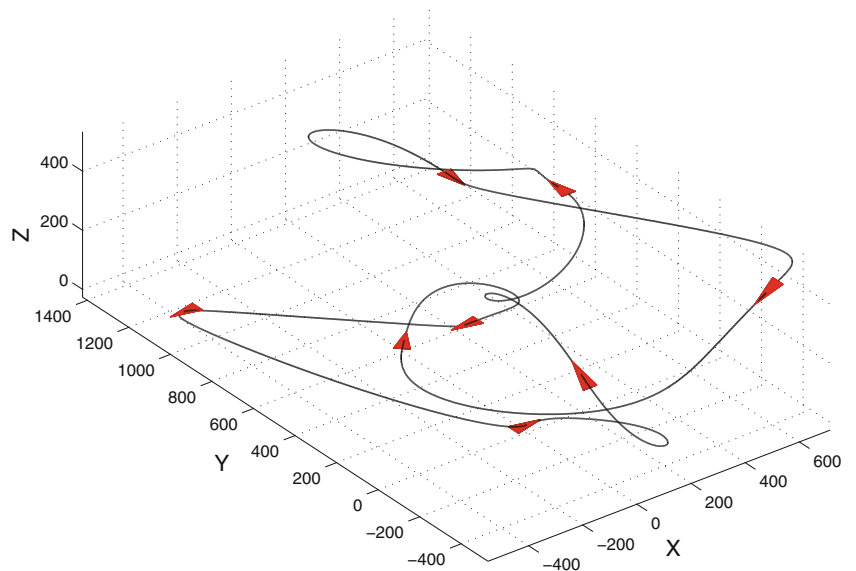
This method promotes a convergence to the result with a very small number of iterations, producing a curve of reasonable length.

conditions of Eq. (6) are observed, and gains are increased by

$$c_0 = c_0 + \frac{\mathcal{E}_0}{\mathcal{E}} (\rho_{\min} + \sigma_{\min}), \tag{30}$$

$$c_7 = c_7 + \frac{\mathcal{E}_7}{\mathcal{E}} (\rho_{\min} + \sigma_{\min}), \tag{31}$$

Fig. 3 \mathcal{R} path among waypoints $\langle p_a, p_h \rangle$ for the single UAV



4 Experiments and results

In this section, we present and discuss the results of our experimental validation. Firstly, we describe the experiments for a simulated airplane, modeled only by the three kinematic constraints, namely curvature, torsion and climb angle. Then, we

Fig. 4 Curvature profile of \mathcal{R} from Fig. 3

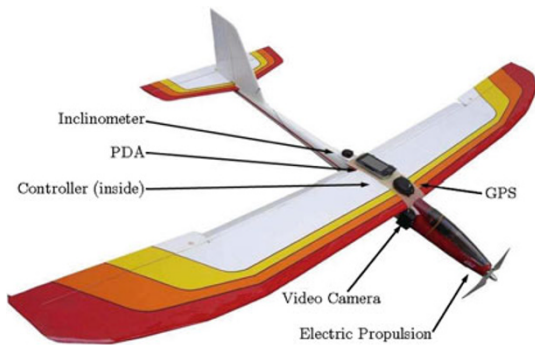
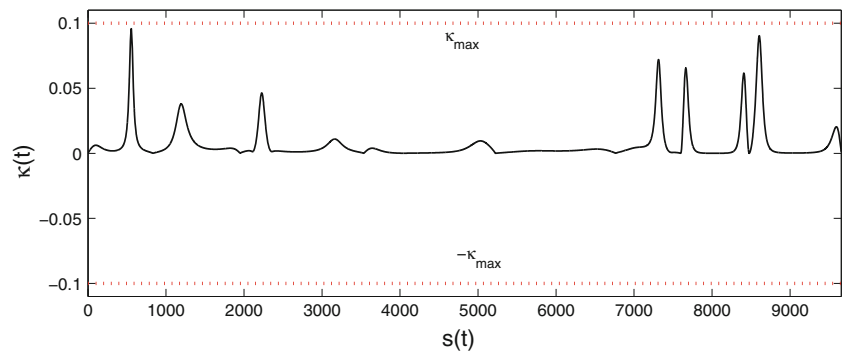


Fig. 5 AqVS, a UAV from the Universidade Federal de Minas Gerais, Brazil [15]

Table 3 AqVS main characteristics [15]

Variable	Value
Total weight	25 N
Wingspan	2 m
Length	1.6 m
Wing loading	27–29 N/m ²
Wing area	0.42 m ²
Cruise speed	13.9 m/s (50 km/h)
Operational ceiling	150 m (above ground level)
Operational radius	10 km
Curvature, radius (ρ_{\min})	150 m
Torsion, radius (σ_{\min})	300 m
Maximum, climb, angle (θ_{\max})	$\frac{\pi}{30}$ rad

apply the proposed method to a simulated model based on a real UAV developed at the Universidade Federal de Minas Gerais/Brazil. All experiments were performed using the Matlab® software (Version R2008b) running on a computer with Ubuntu/Linux 10.04 operating system.

4.1 Virtual airplane

Firstly, we use a single simulated UAV. This step was designed to show the applicability of seventh order Bézier curves to paths of aircraft with relatively small values of

maximum climb angle and with torsion radius much smaller than curvature radius. This is a key feature of our approach, since it would be otherwise very complex to be handled by other methods.

Our vehicle presents the following constraints:

$$\begin{aligned} \rho_{\min} &= 10 \text{ m,} \\ \sigma_{\min} &= 100 \text{ m,} \\ \theta_{\max} &= \frac{\pi}{6} \text{ rad.} \end{aligned}$$

These values were arbitrarily chosen, so that the vehicle would present a relatively small climb angle of about 30°, and a minimum torsion radius ten times larger than the minimum radius of curvature.

At first, we have also established two spatial configurations representing the initial and final waypoints of a single path to be planned, where position variables units are in meters and the orientation ones in radians (Table 1).

A very important point is to insure that the values of the climb angles θ_i and θ_f at these waypoints are smaller in magnitude than the maximum climb angle. Otherwise, it becomes impossible to find values of c_0 and c_7 that guarantee the condition 6 for $\mathbf{r}(t)$. Figure 1 shows the result of our methodology for a single path.

Besides the generated path, it is also possible to see the configuration of the control points computed for the curve. Those points were obtained using the values $\phi_0 = \frac{\pi}{2}$, $\phi_1 = -\frac{\pi}{2}$ and $\phi_2 = -\frac{\pi}{2}$ for our seventh order Bézier curve.

Figure 2 shows the constraint functions of the single path. Curvature (Fig. 2a), torsion (Fig. 2b) and climb angle (Fig. 2c) functions are continuous and bounded by the maximum κ_{\max} , τ_{\max} and θ_{\max} values, respectively.

Also, it is important to notice that the extreme points in the curvature profile of Fig. 2a are both null, as we explained before. This means that, when we use this method to generate a path among several waypoints, all curvature values between single paths will be zero, and the total curvature profile of \mathcal{R} will be continuous.

We next apply our technique to generate feasible paths for a sequence of eight waypoints $\langle p_a, p_h \rangle$. Again, position variables units are in meters and orientation ones in radians. The configurations used are presented at (Table 2).

Fig. 6 AqVS trajectory over the planned paths using Spatial seventh order Bézier curves

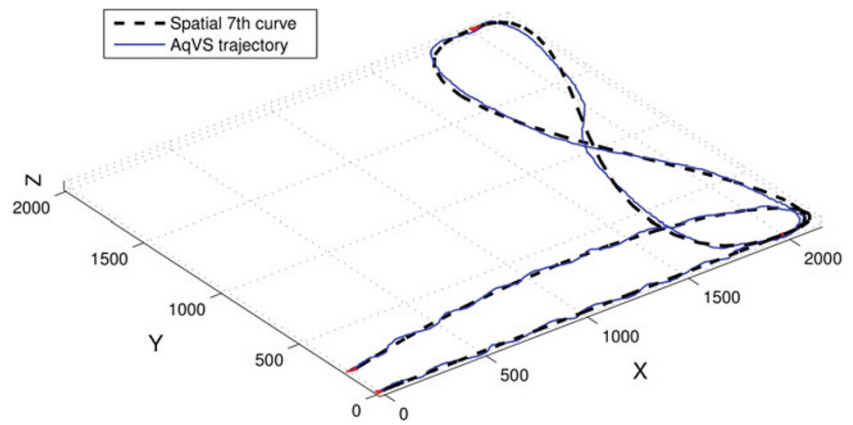


Table 4 Waypoint set $\langle p_a, p_e \rangle$ for the AqVS/UAV

Waypoint	$x(m)$	$y(m)$	$z(m)$	ψ (rad)	θ (rad)
p_a	0	0	1,013	0	0
p_b	2,000	0	1,023	0	$\frac{\pi}{40}$
p_c	2,000	2,000	1,033	0	0
p_d	2,000	0	1,023	0	0
p_e	0	200	1,013	π	0

Figures 3 and 4 present the results obtained for the final path \mathcal{R} and its curvature profile. The mean time and the standard deviation of the method in the generation of all these single paths were about 0.2 and 0.07 s, respectively. It can be easily seen that it produced a continuous and smooth curvature function.

4.2 AqVS UAV

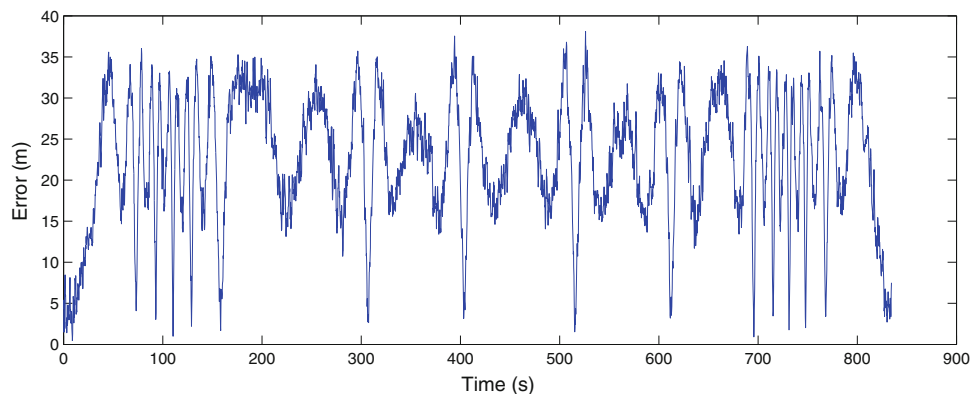
The technique was also used to plan paths for a small unmanned fixed-wing aircraft called AqVS (Fig. 5), developed at Universidade Federal de Minas Gerais. This is a small hand launched hybrid electric motor sail plane, equipped with a GPS receptor, barometric altimeter, infrared inclinometer, airspeed sensor and a CCD camera. It is controlled by a

set of PID stabilizers for autonomous navigation (see [15] for further information). In this part of the work, we use the AqVS simulation environment, implemented using the Matlab/Simulink[®] graphical language. It was based on the assumption that the vehicle can be correctly represented as a rigid body with six degrees of freedom. Table 3 presents the main characteristics of the AqVS autonomous airplane.

The above values were determined using real data from actual flights of the SUAV, considering a speed of approximately 50 km/h. This vehicle has shown to be a good choice for testing our methodology because of its small climb angle (about 6°). This represents a very challenging issue in the context of the problem of generating paths for this vehicle. We used a simulated model of the AqVS to test our methodology. Figure 6 shows the result of the AqVS/SUAV flight over a set of five points, randomly chosen as presented in Table 4.

The blue line in the graph represents the actual trajectory executed by the SUAV when tracking the planned paths. The experiment was realized with wind perturbations of about 10 km/h at the North direction. The flight path was smooth and with the specified constraints. In spite of the noise, which is mainly due to the altitude and GPS sensors and the actions of the control system, the vehicle is able to tightly track the path. Figure 7 presents the absolute error between the real

Fig. 7 Absolute error between the position of the simulated SUAV through time and the planned path



position of the simulated SUAV through time and the planned path, which shows a mean error of approximate 20 m. This error is a result of the wind speed disturbance and the biased sensors uncertainties.

5 Conclusion and future work

We presented a methodology for path planning in three-dimensional environments for autonomous vehicles that consider at least three motion constraints: maximum curvature, maximum torsion and maximum climb angles. The methodology is an extension of the Spatial Pythagorean Hodographs where such constraints are explicitly taken into account.

The use of analytical curves, such as Bézier curves, allowed for greater flexibility of this model with a low computational cost. The design of these curves takes into account very simple kinematics and dynamics constraints, which imply the simplification of the model of the vehicle at few points of operation.

The proposed methodology uses an elastic bending energy function for the resolution of the Spatial Pythagorean Hodograph that minimizes the climb angle function of the curve $\mathbf{r}(t)$, generating a solution that is adequate for vehicles with limited climb (or dive) angle capability. The methodology was also used to generate paths with satisfactory results for a simulated model of a real SUAV. As part of our ongoing research, we plan on incorporating other constraints to the cost function, such as maximum translation speed.

As further investigation, we will also consider environments with static and dynamic objects, which may demand path replanning in real time, which is of major relevance to the case of multiple and cooperating vehicles moving in three-dimensional space.

Acknowledgments The authors gratefully thank Prof. Paulo Iscold from the Center for Aeronautics Studies (CEA) of UFMG for the flight data of the AqVS/UAV. This work was developed with the support of CNPq, CAPES and FAPEMIG.

References

- Alves Neto A, Campos MFM (2009) A path planning algorithm for UAVs with limited climb angle. In: The 2009 IEEE/RSJ international conference on intelligent robots and systems (IROS'09), St. Louis, USA
- Alves Neto A, Campos MFM (2009) On the generation of feasible paths for aerial robots with limited climb angle. In: Proceedings of the IEEE international conference on robotics and automation (ICRA'09). Kobe, Japan
- Alves Neto A, Macharet DG, Campos MFM (2010) Feasible RRT-based path planning using seventh order bézier curves. In: The 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS'10). Taipei, Taiwan
- Alves Neto A, Macharet DG, Campos MFM (2010) On the generation of trajectories for multiple UAVs in environments with obstacles. *J Intell Robotic Syst* 57(4):123–141
- Bortoff S (2000) Path planning for UAVs. In: Proceedings of the American control conference, vol 1, Chicago, Illinois, USA, pp 364–368. doi:10.1109/ACC.2000.878915
- Cheng P, Shen Z, Lavalle SM (2001) RRT-based trajectory design for autonomous automobiles and spacecraft. *Arch Control Sci* 11(3–4):167–194
- Dogan A (2003) Probabilistic path planning for UAVs. In: Proceedings of 2nd AIAA unmanned unlimited systems, technologies, and operations—aerospace, land, and sea conference and workshop exhibition. San Diego, CA, USA
- Farouki RT (1996) The elastic bending energy of Pythagorean hodograph curves. *Comput Aided Geom Des* 13:227–241
- Farouki RT, Han CY (2006) Algorithms for spatial Pythagorean-hodograph curves. In: Klette R, Kozera R, Noakes L, Weickert J (eds) *Geometric properties for incomplete data*, Springer pp 43–58
- Farouki RT, al Kandari M, Sakkalis T (2002) Hermite interpolation by rotation-invariant spatial Pythagorean-hodograph curves. *Adv Comput Math* 17:369–383
- Farouki RT, Neff CA (1995) Hermite interpolation by Pythagorean hodograph quintics. *Math Comput* 64:1589–1609
- Farouki RT, Sakkalis T (1990) Pythagorean hodographs. *IBM J Res Dev* 34(5):736–752
- Gonçalves VM, Pimenta LCA, Maia CA, Dutra BCO, Pereira GAS (2010) Vector fields for robot navigation along time-varying curves in n-dimensions. *Trans Robotics* 26:647–659. doi:10.1109/TRO.2010.2053077
- Griffiths S, Saunders J, Curtis A, Barber B, McLain T, Beard R (2006) Maximizing miniature aerial vehicles. *IEEE Robotics Autom Mag* 13(3):34–43. doi:10.1109/MRA.2006.1678137
- Iscold P, Pereira G, Torres L (2010) Development of a hand-launched small UAV for ground reconnaissance. *IEEE Trans Aerospace Electron Syst* 6(1):335–348. doi:10.1109/TAES.2010.5417166
- Kreyszig E (1991) *Differential Geometry*, vol 1. Dover Publications, New York
- Kuwata Y, Richards A, Schouwenaars T, How JP (2005) Robust constrained receding horizon control for trajectory planning. In: Proceedings of the AIAA guidance, navigation and control conference
- LaValle SM (2006) *Planning algorithms*. Cambridge University Press, Urbana
- Nelson DR, Blake D, Timothy B, McLain W, Beard RW (2006) Vector field path following for small unmanned air vehicles. *IEEE Trans Control Syst Technol* 48: 5788–5794
- Sederberg TW (2007) *Computer aided geometric design*, chap. 2. Brigham Young University, Provo, Utah
- Shanmugavel M, Tsourdos A, Zbikowski R, White BA (2007) 3D path planning for multiple UAVs using Pythagorean hodograph curves. In: Proceedings of the AIAA guidance, navigation and control conference and exhibit (AIAA-GNC). Hilton Head, South Carolina
- Shanmugavel M, Tsourdos A, Zbikowski R, White BA, Rabbath CA, Lechevin N (2006) A solution to simultaneous arrival of multiple UAVs using Pythagorean hodograph curves. In: Proceedings of the IEEE American control conference (ACC), pp 2813–2818, Minneapolis, Minnesota, USA
- Wzorek M, Doherty P (2006) Reconfigurable path planning for an autonomous unmanned aerial vehicle. In: International conference on hybrid information technology (ICHIT '06), vol 2, pp 242–249