IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 July 2022, accepted 27 July 2022, date of publication 31 August 2022, date of current version 14 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203069

RESEARCH ARTICLE

Smooth Three-Dimensional Route Planning for Fixed-Wing Unmanned Aerial Vehicles With Double Continuous Curvature

GLORIA VANEGAS[1], LEOPOLDO ARMESTO[1], VICENT GIRBÉS-JUAN[2], (Member, IEEE), AND JOAQUÍN PÉREZ[2], (Member, IEEE)

[1]Instituto de Diseño y Fabricación, Universitat Politècnica de València, 46022 Valencia, Spain
[2]Departament d'Enginyeria Electrònica, Universitat de València, 46100 Burjassot, Spain

Corresponding author: Vicent Girbés-Juan (vicent.girbes@uv.es)

This work was supported in part by the Generalitat Valenciana under Grant GV/2021/074, and in part by the MCIN/AEI/10.13039/501100011033 under Grant PID2020-116585GB-I00, Grant PID2020-118071GB-I00, and Grant PID2020-113785RB-100.

ABSTRACT This paper presents a smooth flight path planner for maneuvering in a 3D Euclidean space, which is based on two new space curves. The first one is called ''Elementary Clothoid-based 3D Curve (ECb3D)'', which is built by concatenating two symmetric Clothoid-based 3D Curves (Cb3D). The combination of these curves allows to reach an arbitrary orientation in 3D Euclidean space. This new curve allows to generate continuous curvature and torsion profiles that start and finish with a null value, which means that they can be concatenated with other curves, such as straight segments, without generating discontinuities on those variables. The second curve is called ''Double Continuous Curvature 3D Curve (DCC3D)'' which is built as a concatenation of three straight line segments and two ECb3D curves, allowing to reach an arbitrary configuration in position and orientation in the 3D Euclidean space without discontinuities in curvature and torsion. This trajectory is applied for autonomous path planning and navigation of unmanned aerial vehicles (UAVs) such as fixed-wing aircrafts. Finally, the results are validated on the FlightGear 2018 flight simulator with the UAV kadett 2400 platform.

INDEX TERMS Nonholonomic motion planning, motion and path planning, constrained motion planning, autonomous vehicle navigation.

I. INTRODUCTION

In the last decades, the aeronautics industry has maintained a continuous and vertiginous development, particularly in the military field [1]. The literature in the different fields of research is extensive. A relevant field of study focuses on path planning, which has been approached from different perspectives, such as communication networks [2], [3] or computational intelligence based on path planning algorithms [4], [5]. On the other hand, applications for civilian missions are currently in high demand [6], such as rescue missions [7], [8], or work in agriculture [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenbao Liu.

The aim of flight planning for aerial vehicles is to generate a path joining initial and final configurations, while passing through several intermediate target points. In case of potential collisions with static or dynamic objects, such as other UAVs flying nearby, the initial route must be replanned to guarantee collision-free paths [10], [11].

In particular, this paper addresses the 3D path planning task for UAVs with nonholonomic characteristics [12], [13], i.e., fixed-wing aircrafts. In this sense, it should be highlighted that one of the most relevant particularities of this kind of paths is the continuity of the curve, since a fixed-wing UAV cannot perform abrupt maneuvers during its flight time. Therefore, the starting point to create a smooth flight path must consider the particular maneuverability capabilities of the UAV. Several studies have been proposed in this context,

where 3D path planning algorithms use cubic Bezier spiral curves to satisfy the curvature constraint are presented in [14], [15], [16], while [17] proposes a seventh-order Bézier curve as a continuous curvature path approximation, which does not exceed the kinematic constraints of an aerial vehicle. The authors of [18] have performed a fusion between two heuristic methodologies, with the aim of solving the smooth path planning problem in a mountainous environment, for which the characteristics of B-spline curves are exploited.

The authors of [19] study curvature constraints in path planning and solve this problem through Dubins curves. In [20], the smooth trajectory planning problem with continuous curvature is solved through an optimization algorithm based on Pythagorean curves, which satisfies the kinematic constraints of the UAV, in a similar approach, the authors of [21] propose a path planning generated through a multiobjective optimization problem operating with standard genetic operators. In [22] coordinated path planning for multiple UAVs is performed, starting from an ant colony optimization algorithm smoothed through a k-degree smoothing method. While in [23] a Rauch-Tung-Striebel (RTS), smoothing is used, a procedure that permits smoothing the path produced by a Particle Swarm Optimization (PSO) algorithm, while in [24] an RTS smoothing is also used for the control of mobile robots with nonholonomic wheels. A particular approach is presented in [25], where a smooth path planning algorithm based on a Gaussian spectrum function is built, which aims to optimize the smooth path. Finally, in [26] model-based smooth paths are proposed for the estimation of the optimal geometric parameters, through polynomial spline curves, the results are used in industrial robots, with the aim of improving the productivity.

Focusing on works with good results based on clothoid curves, [27] solves the problem of generating continuous curvature paths by composing multiple clothoids. The relevance of clothoid curves and their application to nonholonomic vehicles can also be appreciated in [28], [29], and [30]. In [31] and [32], clothoids are approximated using Bézier curves to minimize curvature profiles and thus guarantee higher-order geometric continuity while minimizing error. In [33], smooth paths based on clothoid curves are proposed for planning high-speed wheeled vehicle paths, for which a numerical optimization is performed within the constraints of convex regions. In [34], autonomous valet parking service path planning is performed. Finally, the authors of [35] propose a path smoothing, based on clothoid curves, parameterized by the arc length.

The aim of this work is to generate a smooth flight path in the context of fixed-wing UAV autonomous navigation. A new 3D smooth curve, called Elementary Clothoid-based 3D Curve or ECb3D, is proposed, which is built by combining two symmetric Cb3D [36]. An ECb3D is capable of reaching an arbitrary direction in 3D space, being its curvature and torsion profiles equal to zero at both the beginning and the end of the curve. That property allows to build more complex curves combining them with straight line segments or other ECb3D curves. In this sense, a second 3D smooth curve is introduced, coined as Double Continuous Curvature 3D Curve (DCC3D), which is a concatenation of two ECb3D curves and three straight line segments. A DCC3D curve can reach any arbitrary position and orientation in 3D Euclidean space. Finally, it should be emphasized that collision avoidance is out of the scope of this paper, although the proposed path could be used as a primitive in both global and local planners to generate collision-free paths.

This paper is organized as follows: in section II, the formulations of the preliminary works related to this article are discussed. Section III describes the problem to be solved. In Section IV, the methodology to design smooth curves to reach arbitrary target orientation is explained in depth, whereas Section V describes how to generate 3D Double Continuous Curvature Curves that allow to reach arbitrary position and orientation. Section VI presents the results that validate the application of this new curve through flight simulations performed on a fixed-wing UAV. Finally, conclusions and further work are described in Section VII.

II. PRELIMINARIES

A nonholonomic constraint cannot be expressed only in position variables but includes the time derivative of one or several variables. In direct reference to fixed-wing UAVs, these constraints are directly related to their maneuverability in flight.

There are different approaches used for the construction of smooth paths, whether they are heuristic [37] or geometrical [38], [39], [40]. Thus, this work takes a geometric approach for constructing smooth paths as a starting point, based on the criterion of 3D continuous curves. Thereafter, a set of concepts necessary for the development of this article are defined.

A. CURVES IN SPACE

A curve in space $\mathbb{R}^n$, can be defined as a vector function [41] such that:

$$\mathbf{C} : [a, b] \to \mathbb{R}^n, \quad \mathbf{C}(s) = (x_1(s), \ldots, x_n(s)) \tag{1}$$

where, the points $\mathbf{C}(a)$ and $\mathbf{C}(b)$ are the initial and final boundaries of the curve. In particular, a curve in the three-dimensional space $\mathbb{R}^3$ can be defined as $\mathbf{C}(s) = (x(s)_\mathbf{i} + y(s)_\mathbf{j} + z(s)_\mathbf{k})$, where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ refer to the unit vectors of the global reference frame.

Tangent, normal and *binormal* vectors are defined as:

- $\mathbf{T}(s)$ is the unit vector tangent to the curve, pointing tp the direction of movement:

$$\mathbf{T}(s) = \frac{\mathbf{C}'(s)}{\|\mathbf{C}'(s)\|}. \tag{2}$$

- $\mathbf{N}(s)$ is the unit normal vector, given by the ratio of the derivative of $\mathbf{T}(s)$ to its length:

$$\mathbf{N}(s) = \frac{[\mathbf{C}'(s) \times \mathbf{C}''(s)] \times \mathbf{C}'(s)}{\|[\mathbf{C}'(s) \times \mathbf{C}''(s)] \times \mathbf{C}'(s)\|}. \tag{3}$$

- $\mathbf{B}(s)$ is the binormal unit vector, represented by the cross product of $\mathbf{T}(s)$ and $\mathbf{N}(s)$:

$$\mathbf{B}(s) = \mathbf{T}(s) \times \mathbf{N}(s) = \frac{\mathbf{C}'(s) \times \mathbf{C}''(s)}{\|\mathbf{C}'(s) \times \mathbf{C}''(s)\|}, \tag{4}$$

where $\mathbf{C}'(s) = \mathrm{d}\mathbf{C}(s)/\mathrm{d}s$, $\mathbf{C}''(s) = \mathrm{d}^2\mathbf{C}(s)/\mathrm{d}s^2$, and $\mathbf{C}'''(s) = \mathrm{d}^3\mathbf{C}(s)/\mathrm{d}s^3$ are the derivatives of the position vector $\mathbf{C}(s)$.

On the other hand, $\kappa(s)$ defines the curvature of the curve, such that:

$$\kappa(s) = \frac{\|\mathbf{C}'(s) \times \mathbf{C}''(s)\|}{\|\mathbf{C}'(s)\|^3}, \tag{5}$$

and $\tau(s)$ is the torsion, defined as:

$$\tau(s) = \frac{\mathbf{C}'(s) \cdot (\mathbf{C}''(s) \times \mathbf{C}'''(s))}{\|\mathbf{C}''(s)\|^2} \tag{6}$$

Therefore, for a continuous curve defined in $\mathbb{R}^3$, as the one shown in Figure 1, $\mathbf{T}(s)$ is a unit vector defining the direction of the curve, $\mathbf{N}(s)$ is perpendicular to $\mathbf{T}(s)$, while $\mathbf{B}(s)$ forms a right-handed system between $\mathbf{T}(s)$ and $\mathbf{N}(s)$. Based on Frenet-Serret frame, the derivative of equations (2)-(4) depends on (5) and (6), and can be computed as follows:

$$\begin{bmatrix} \mathbf{T}'(s) \\ \mathbf{N}'(s) \\ \mathbf{B}'(s) \end{bmatrix} = \begin{bmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T}(s) \\ \mathbf{N}(s) \\ \mathbf{B}(s) \end{bmatrix}, \tag{7}$$

where $\mathbf{T}'(s) = \mathrm{d}\mathbf{T}(s)/\mathrm{d}s$, $\mathbf{N}'(s) = \mathrm{d}\mathbf{N}(s)/\mathrm{d}s$ and $\mathbf{B}'(s) = \mathrm{d}\mathbf{B}(s)/\mathrm{d}s$ are the first derivatives of such vectors.

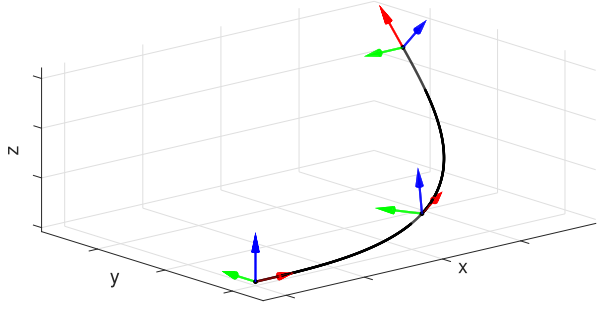

FIGURE 1. Curve in the space $\mathbb{R}^3$, where three orthogonal local systems are shown, which are defined by T vectors (red arrows), N vectors (green arrows) and B vectors (blue arrows).

Then, the orthogonal basis of the system is defined as $\mathbf{R}(s) := [\mathbf{T}(s)\ \mathbf{N}(s)\ \mathbf{B}(s)]$, which can be integrated from (7), based on the functions of $\kappa(s)$ and $\tau(s)$, from an initial value, such that, $\mathbf{R}(0) := [\mathbf{T}(0)\ \mathbf{N}(0)\ \mathbf{B}(0)]$. Hence, the position can be determined by integrating the tangent vector.

$$\mathbf{C}(s) := \mathbf{C}(0) + \int_0^s \mathbf{T}(\xi)\mathrm{d}\xi \tag{8}$$

B. CLOTHOID IN SPACE $\mathbb{R}^2$ (C2D)

A planar clothoid (C2D) [42], also known as the Euler Spiral, defined in $\mathbb{R}^2$ (see Figure 2), is a curve whose curvature varies linearly with respect to the arc length, being:

$$\kappa(s) := \sigma_\kappa s \tag{9}$$

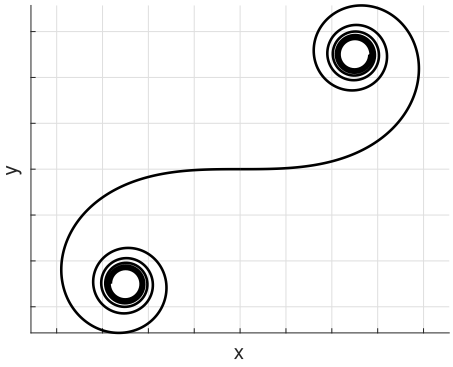

FIGURE 2. Clothoid curve in space $\mathbb{R}^2$.

where, $\sigma_\kappa := \mathrm{d}\kappa(s)/\mathrm{d}s$ is referred to as the curvature sharpness, which is related to the homotopy factor K, being $\sigma_\kappa := \pi/K^2$. Hence, the tangent angle of the clothoid is defined as:

$$\beta(s, \sigma_\kappa) := \frac{\sigma_\kappa}{2} s^2 \tag{10}$$

The C2D is a curve that has contributed in various aspects, both in development and construction of roads and/or railroads [43], [44], and also in research [45], giving nonholonomic vehicles a good tracking control, due to its various geometric properties such as curvature and tangent angle. Being the tangent vector computed as:

$$\mathbf{T}(s, \sigma_\kappa) := \begin{bmatrix} \cos(\beta(s, \sigma_\kappa)) \\ \sin(\beta(s, \sigma_\kappa)) \end{bmatrix} \tag{11}$$

Consequently, a planar clothoid curve contained in the plane XY, is defined by the equations (8), (10) and (11). Hence, $\mathbf{C}(s, \sigma_\kappa)$ can be solved using the Fresnel integrals, as follows:

$$\mathbf{C}(s, \sigma_\kappa) := \begin{bmatrix} \mathcal{C}(s, \sigma_\kappa) \\ \mathcal{S}(s, \sigma_\kappa) \end{bmatrix} := \begin{bmatrix} \int_0^s \cos(\frac{\sigma_\kappa}{2}\xi^2)\mathrm{d}\xi \\ \int_0^s \sin(\frac{\sigma_\kappa}{2}\xi^2)\mathrm{d}\xi \end{bmatrix} \tag{12}$$

where $\mathcal{C}(s, \sigma_\kappa)$ and $\mathcal{S}(s, \sigma_\kappa)$ are the Fresnel integrals in cosine and sine, respectively. Finally, it should be noted that without loss of generality, it is assumed that the clothoid starts from the origin of coordinates, being $\mathbf{C}(0) = 0$ according to the equation (8).

C. CLOTHOID IN SPACE $\mathbb{R}^3$ (C3D)

The concept of Euler Spiral defined in $\mathbb{R}^3$, also known as a 3D clothoid (C3D), was introduced by [46]. The curvature of a C3D varies as a function of the equation (9), while its torsion is defined as:

$$\tau(s) := \sigma_\tau s \tag{13}$$

where $\sigma_\tau := \mathrm{d}\tau/\mathrm{d}s$ is the first geometric derivative (also known as torsion sharpness). The development of this 3D smooth curve allows arbitrary configurations to be achieved, either in position or orientation in 3D space, but not both at the same time.

III. PROBLEM DEFINITION

Let us assume $\mathcal{R}$ as a UAV of nonholonomic characteristics, such as a fixed-wing aircraft, whose state space $\mathbf{q}_\mathcal{R} = [x_\mathcal{R}\ y_\mathcal{R}\ z_\mathcal{R}\ \theta_\mathcal{R}\ \Psi_\mathcal{R}]^T \in \mathbb{R}^3 \times \mathbb{S}^2$, is composed of the coordinates position $\mathbf{p}_\mathcal{R} = [x_\mathcal{R}\ y_\mathcal{R}\ z_\mathcal{R}]^T$, and orientation coordinates $\angle\mathbf{q}_\mathcal{R} = [\theta_\mathcal{R}\ \Psi_\mathcal{R}]^T$, pitch and yaw angles. Whereas the input parameters to the system are given by the curvature sharpness $\sigma_{\kappa_\mathcal{R}}$ and the torsion sharpness $\sigma_{\tau_\mathcal{R}}$. It is important to remark that the roll angle of the curve is not relevant for computing the curve geometrically and, for this reason, it is not considered as part of the configuration.

Assuming that $\mathcal{R}$ can perform motions within its kinematic boundaries of maneuverability (boundaries set by the particular aerodynamic constraints of the UAV) and that the values of the geometric derivatives of curvature and torsion (defined as curvature sharpness and torsion sharpness, respectively), are within the set boundaries, being $\sigma_{\kappa_\mathcal{R}} \in [\sigma_{\kappa_{min}}\ \sigma_{\kappa_{max}}]$ and $\sigma_{\tau_\mathcal{R}} \in [\sigma_{\tau_{min}}\ \sigma_{\tau_{max}}]$. Then, the aim is to build a new smooth curve G^1 to join two arbitrary configurations in position and orientation, starting from $\mathbf{q}_S = [x_S\ y_S\ z_S\ \theta_S\ \Psi_S]^T$ to $\mathbf{q}_G = [x_G\ y_G\ z_G\ \theta_G\ \Psi_G]^T$. In addition to this, curvature and torsion at start and goal configurations must be zero, i.e., $\kappa_S = \kappa_G = 0$ and $\tau_S = \tau_G = 0$.

IV. ELEMENTARY CLOTHOID-BASED 3D CURVE (ECb3D)

The aim is to construct a new three-dimensional curve, in the space $\mathbb{R}^3 \times \mathbb{S}^2$, able to reach an arbitrary configuration in orientation with continuous curvature and torsion (CC). This new curve will be composed of two segments of the Clothoid-based 3D curve (Cb3D) [36].

The procedure starts with the description of the Cb3D curve. Afterwards, the generation process of the new Elementary Clothoid-based 3D Curve (ECb3D), generated from the concatenation of two symmetrical Cb3D curves is detailed. Specifically, the ECb3D will allow reaching an arbitrary orientation, but the position will be given according to the shape and size of the constructive clothoid parameters.

A. CLOTHOID-BASED 3D CURVE (Cb3D)

The authors of [36], propose a new Clothoid-based 3D Curve (Cb3D), capable of achieving an arbitrary configuration in position or orientation in 3D space. The Cb3D is built from two C2D curves generated in the orthogonal *XY* and *XZ* planes. Thus, Cb3D projects a C2D curve in the *XY* plane with arc length *s*, while the clothoid in the orthogonal *XZ* plane depends on the length of the C2D curve in the *XY* plane (see Figure 3). The curve Cb3D achieves relevant results due to its analytical solution. Moreover, the curve presents a set of interesting properties/operations such as scalability, symmetry, monotonicity, and smoothness along the curve. Appendix A includes the most relevant aspects of the curve [36], which have been included in this work to justify the computations of the proposed methodology.

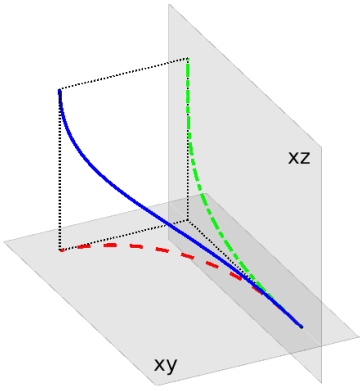

FIGURE 3. Clothoid-based 3D Curve (Cb3D). The dotted red line shows the C2D construction in the *XY* plane, the green line shows the C2D in the *XZ* plane while the blue line shows the Cb3D curve. Image taken from [36].

B. ECb3D CURVE GENERATION

The concept of Elementary Curve in $\mathbb{R}^2$ (E2D) was introduced by [47], a curve developed to build appropriate paths for mobile wheeled vehicles. In that sense, the E2D is built by combining two symmetric C2D curves that have the same homotopy factor and the same length. The goal is to create smooth trajectories that do not exceed certain physical limits associated with comfort and safety in mobile vehicles [45], leading to proper path following.

In particular, the ECb3D curve seeks to extend the original concept of E2D curve to the space $\mathbb{R}^3$, by concatenating two symmetric Cb3D curves [36]. Continuity and smoothness are guaranteed due to the properties of the original Cb3D curve (see Appendix A for a brief description of Cb3D curves). An ECb3D curve has zero curvature and torsion values at start and goal configurations, that is $\kappa_S = \kappa_G = \tau_S = \tau_G = 0$.

Figure 4 shows an example of ECb3D curve, where $\mathbf{C}(s, \mu, \rho)$ (in solid blue) is the first Cb3D curve,

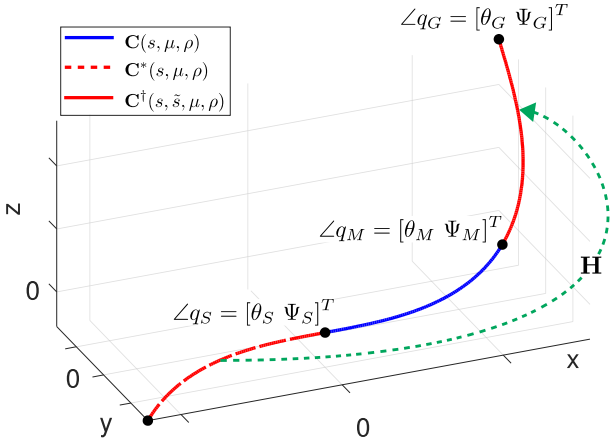

FIGURE 4. ECb3D curve generation from the symmetry of two concatenated Cb3D curves, through an intermediate deflection angle $\angle\mathbf{q}_M$. It is assumes an initial orientation configuration $\angle\mathbf{q}_S = [\theta_S\ \Psi_S]^T = [0\ \ 0]^T$, while the goal orientation configuration is $\angle\mathbf{q}_G = [\theta_G\ \Psi_G]^T$.

$\mathbf{C}^*(s,\mu,\rho) = \mathbf{C}(-s,\mu,\rho)$ (in dashed red) is symmetrical to $\mathbf{C}$, and $\mathbf{C}^\dagger(s,\tilde{s},\mu,\rho)$ (in solid red) is the second Cb3D curve, which is $\mathbf{C}^*$ rotated and translated. Consequently, the curve $\mathbf{C}^\dagger$ starts at configuration $\mathbf{q}_M$ and ends at configuration $\mathbf{q}_G$. Note that in Figure 4 the start configuration $\mathbf{q}_S$ is at the origin of coordinates aligned with x-axis, so $\angle\mathbf{q}_S = [\theta_S\ \Psi_S]^T = [0\ 0]^T$. In a generic case, where start configuration with respect to the global frame is different from zero, that is, $\angle\mathbf{q}_S^0 \neq [0\ 0]^T$, we can compute the *local* tangent vector of $\angle\mathbf{q}_G$, by considering the following rotation:

$$\mathbf{T}_G \equiv \begin{bmatrix} \cos(\theta_G)\cos(\Psi_G) \\ \cos(\theta_G)\sin(\Psi_G) \\ -\sin(\theta_G) \end{bmatrix} := \mathbf{R}^T(0,\theta_S^0,\Psi_S^0) \times \begin{bmatrix} \cos(\theta_G^0)\cos(\Psi_G^0) \\ \cos(\theta_G^0)\sin(\Psi_G^0) \\ -\sin(\theta_G^0) \end{bmatrix} \tag{14}$$

with $\angle\mathbf{q}_S^0 := \left[\theta_S^0\ \Psi_S^0\right]^T$ and $\angle\mathbf{q}_G^0 := \left[\theta_G^0\ \Psi_G^0\right]^T$ being start and goal angles expressed with respect to a global frame and $\mathbf{R}$ is a rotation matrix

$$\mathbf{R}(\phi,\theta,\Psi) := \mathbf{R}_z(\Psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \tag{15}$$

where $\mathbf{R}_x(\bullet)$, $\mathbf{R}_y(\bullet)$ and $\mathbf{R}_z(\bullet)$ are basic rotation matrices around each axis of the global reference frame; being roll ϕ, pitch θ and yaw Ψ the Euler angles.

Then, the deflection angle $\angle\mathbf{q}_G = [\theta_G\ \Psi_G]^T$ can be computed from (14) as follows

$$\theta_G := \arctan\left(\frac{\mathbf{T}_{G,z}}{\sqrt{(\mathbf{T}_{G,x})^2 + (\mathbf{T}_{G,y})^2}}\right) \tag{16}$$

$$\Psi_G := \arctan\left(\frac{\mathbf{T}_{G,y}}{\mathbf{T}_{G,x}}\right) \tag{17}$$

where subscripts x, y and z refer to each component of the tangent vector $\mathbf{T}_G$.

On the other hand, for the computation of the intermediate deflection angle $\angle\mathbf{q}_M$, the following rotations must be applied to ensure that the curve is continuous (in position, curvature and torsion):

$$[\mathbf{T}_G,\mathbf{N}_G,\mathbf{B}_G] = \mathbf{R}_z(\Psi_M)\cdot\mathbf{R}_y(\theta_M)\cdot\mathbf{R}_y(\theta_M)\cdot\mathbf{R}_z(\Psi_M)$$
$$\mathbf{T}_G = \begin{bmatrix} 2\cos^2(\theta_M)\cos^2(\Psi_M) - 1 \\ 2\cos^2(\theta_M)\cos(\Psi_M)\sin(\Psi_M) \\ -2\cos(\theta_M)\cos(\Psi_M)\sin(\theta_M) \end{bmatrix}. \tag{18}$$

where the first column of the right-side expression corresponds to the symbolic expression for computing $\mathbf{T}_G$. Thus, with equations (14) and (18) we can compute the intermediate deflection angle $\angle\mathbf{q}_M = [\theta_M\ \Psi_M]^T$:

$$\theta_M := \arctan\left(\frac{\tan(\theta_G)\sin(\Psi_M)}{\sin(\Psi_G)}\right)$$
$$\Psi_M := \arctan\left(\frac{\cos(\theta_G)\sin(\Psi_G)}{\cos(\theta_G)\cos(\Psi_G)+1}\right). \tag{19}$$

Therefore, the parameters necessary to build Cb3D curves are computed by means of the following expressions (see Appendix):

$$\rho^* := \frac{2\theta_M^*}{\tilde{s}^2} \tag{20}$$

$$\mu^* := \frac{2\Psi_M^*}{\mathcal{C}^2(\tilde{s},\rho^*)}, \tag{21}$$

being $\tilde{s}$ the length of one Cb3D, that is, the half length of the ECb3D curve.

On the other hand, the transformation governing the rotation and translation of the curve $\mathbf{C}^*(s,\mu,\rho)$ to obtain $\mathbf{C}^\dagger(s,\mu,\rho)$ (see Figure 4) is expressed as:

$$\mathbf{H} = \begin{bmatrix} 2\mathbf{T}(\tilde{s},\mu,\rho)\mathbf{T}^T(\tilde{s},\mu,\rho) - \mathbf{I} & \mathbf{0}^T \\ 2\mathbf{T}(\tilde{s},\mu,\rho)\mathbf{T}^T(\tilde{s},\mu,\rho)\mathbf{C}(\tilde{s},\mu,\rho) & 1 \end{bmatrix}^T \times \begin{bmatrix} \boldsymbol{C}^\dagger(s,\tilde{s},\mu,\rho) \\ 1 \end{bmatrix} = \mathbf{H}\cdot\begin{bmatrix} \boldsymbol{C}(-s,\mu,\rho) \\ 1 \end{bmatrix}^T, \tag{22}$$

where, it can be seen that, as the $\mathbf{C}^\dagger$ curve has been defined, when $s = 0$ the curve is in the $\mathbf{q}_M$ configuration, while when $s = \tilde{s}$, the curve ends in the $\mathbf{q}_G$ configuration.

Finally, as a result of the concatenation of the curves $\mathbf{C}$ and $\mathbf{C}^\dagger$, the ECb3D curve is defined as:

$$\mathbf{E}(s,\tilde{s},\mu,\rho) = \begin{cases} \mathbf{C}(s,\mu,\rho) & \text{if } s < \tilde{s} \\ \mathbf{C}^\dagger(2\tilde{s}-s,\tilde{s},\mu,\rho) & \text{otherwise} \end{cases} \tag{23}$$

Hence, the ECb3D curve is evaluated for $s \in [0\ \ 2\tilde{s}]$, with no discontinuity.

Figure 5, shows a comparative example of the ECb3D curve (blue line) and the Cb3D curve (dashed red line), both of which point to a goal orientation of $\theta_G = -\pi/4$ and $\Psi_G = \pi/2$ with $\tilde{s} = 0.5$. In Figure 5(b), the variation of the pitch θ and yaw Ψ Euler angles is displayed, while in Figures 5(c) and 5(d) the behavior in the curvature profiles and their sharpness in the orthogonal XY and XZ planes can be appreciated. It can be highlighted that the ECb3D curve starts and ends with curvature and torsion equal to zero ($\kappa = \tau = 0$), characteristic that allows the ECb3D to link with another curve without loss of continuity, a property lacking in the Cb3D.

V. DOUBLE CONTINUOUS CURVATURE 3D CURVE (DCC3D)

This section defines a curve that reaches an arbitrary configuration in position and orientation in the 3D Euclidean space. The new Double Continuous Curvature 3D Curve (DCC3D) is built by three straight line segments and two ECb3D curves, as can be seen in Figure 6.

In this sense, since there are two ECb3D paths in a DCC3D, a subscript has been added in order to refer to the corresponding ECb3D. Therefore, configurations $\mathbf{q}_S$, $\mathbf{q}_M$ and $\mathbf{q}_G$ of an ECb3D have been renamed as $\mathbf{q}_{S,i}$, $\mathbf{q}_{M,i}$ and $\mathbf{q}_{G,i}$, respectively, where $i = 1$ refers to the first ECb3D, E_1, and $i = 2$ to the second one, E_2. Thus, the following equalities

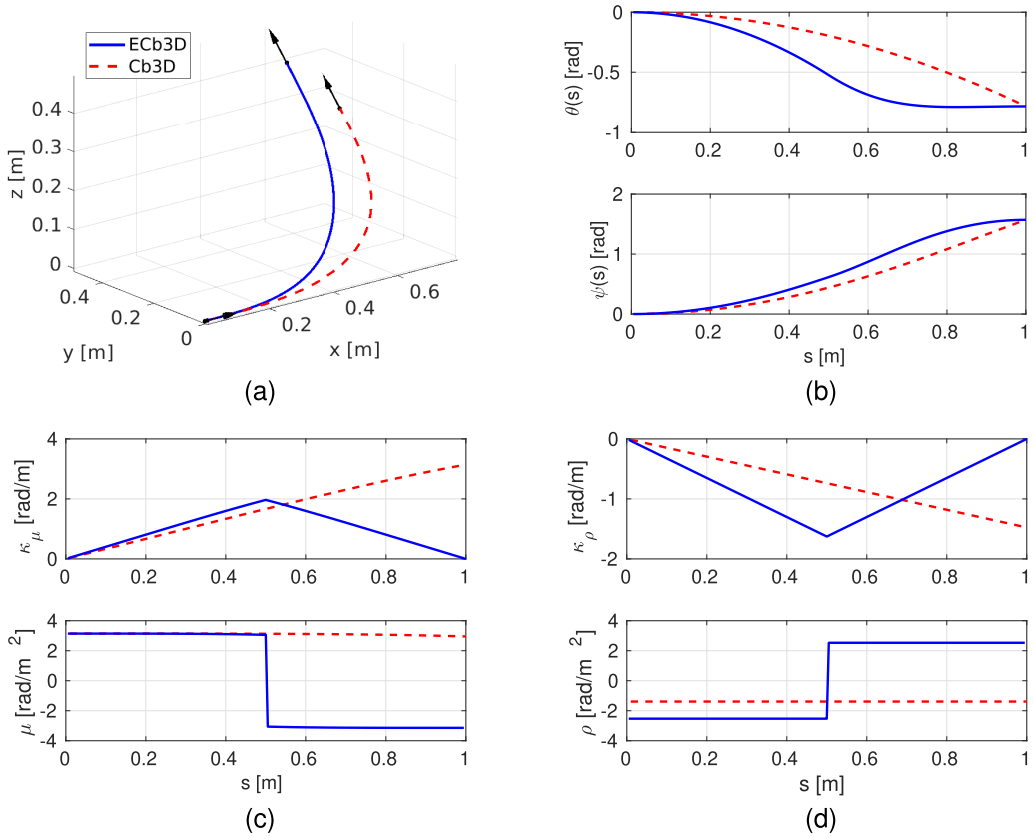

FIGURE 5. Case study with final orientation $\theta_G = -\pi/4$ rad and $\Psi_G = \pi/2$ rad, with $\tilde{s}=0.5$. ECb3D (blue) vs. Cb3D (dashed red): (a) position, (b) Euler angles, (c) curvature profile and its sharpness in the orthogonal plane *XY*, (d) curvature profile and its sharpness in the orthogonal plane *XZ*.

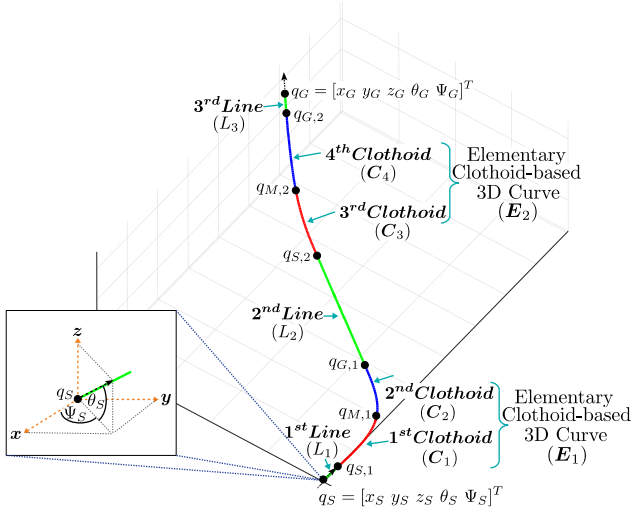

FIGURE 6. Double continuous curvature 3D curve (DCC3D).

relative to configurations' orientations hold: $\angle\mathbf{q}_S \equiv \angle\mathbf{q}_{S_1}$, $\angle\mathbf{q}_{G_1} \equiv \angle\mathbf{q}_{S_2}$, $\angle\mathbf{q}_{G_2} \equiv \angle\mathbf{q}_G$.

Since there are multiple solutions to the problem, the deflection angle $\angle\mathbf{q}_{G,1}$ is assumed to be known, which allows determining the straight lines and clothoids' lengths. That is possible because derivatives of curvature and torsion are performed at the maximum admissible sharpness, that is $|\mu| \leq \mu_{max}$ and $|\rho| \leq \rho_{max}$. This allows to obtain the shortest path for the given deflection angle in a preliminary step. Thus, this states a numerical optimization problem that aims to obtain the shortest curve by finding the intermediate orientation $\angle\mathbf{q}_{G,1}$ that minimizes the overall length, while ensuring that the path satisfies some constraints to avoid abrupt changes in curvature and torsion. As shown later on, the lengths of the first and third straight lines act as *relaxation variables* in order to satisfy position constraints for a given $\angle\mathbf{q}_{G,1}$.

A. DCC3D CURVE GENERATION

Let us assume that the intermediate orientation of the DCC3D curve is given, that is, $\angle\mathbf{q}_{G,1}$. Let us also assume that in order to satisfy position constraints imposed by $\mathbf{q}_G$, we allow maneuvers, that is, changes in the direction of the curve that would force a robot stop to avoid abrupt changes in curvature and torsion. The steps to generate the DCC3D curve are described next.

1) **Compute the first ECb3D**: the intermediate deflection angle $\angle\mathbf{q}_{M,1} = [\theta_{M,1}\ \Psi_{M,1}]^T$ is determined from

equation (19) using the orientation of the goal configuration $\angle\mathbf{q}_{G,1} = [\theta_{G,1}\ \Psi_{G,1}]^T$ (assuming that the start configuration $\angle\mathbf{q}_{S,1} = [0\ 0]^T$). Otherwise, a rotation must be applied to compute $\angle\mathbf{q}_{G,1}$ with respect to the local frame of $\angle\mathbf{q}_{S,1}$, as in equation (14).

Assuming that the curve is computed with the maximum allowable torsion sharpness $\rho_1 := \rho_{max}$, the arc-length of each Cb3D curve (of the ECb3D), and the curvature sharpness to reach the deflection angle $\angle\mathbf{q}_{M,1}$ can be obtained from (20) and (21) as follows:

$$\tilde{s}_1 := \sqrt{\frac{2|\theta_{M,1}|}{\rho_{max}}}, \quad \mu_1 := \frac{2\Psi_{M,1}}{\mathcal{C}^2(\tilde{s}_1, \rho_{max})} \tag{24}$$

If $|\mu_1| > \mu_{max}$, then it means that the assumption of the previous point was incorrect and, therefore, the curve with the maximum admissible curvature sharpness $\mu_1 := \mu_{max}$ must be computed as:

$$\tilde{s}_1 := \frac{\sqrt{\frac{2|\Psi_{M,1}|}{\mu_{max}}}}{\mathcal{C}(1, 2|\theta_{M,1}|)}, \quad \rho_1 := \frac{2\theta_{M,1}}{\tilde{s}_1^2}. \tag{25}$$

2) **Compute the second ECb3D**: repeat the previous step to obtain curvature μ_2 and torsion ρ_2 sharpness parameters, as well as the arc-length $\tilde{s}_2$ based on deflection angle between $\angle\mathbf{q}_{S,2} \equiv \angle\mathbf{q}_{G,1}$ and $\angle\mathbf{q}_{G,2} \equiv \angle\mathbf{q}_G$.
3) **Compute line segments**: it is interesting to remark that the resulting curve, as a consequence of concatenating the two ECb3D, satisfies orientation constraints, but it does not satisfy position constraints. Indeed, the position displacement imposed by the clothoid arcs is:

$$\mathbf{p}_{E_1,E_2} := \mathrm{E}_1(2\tilde{s}_1) + \mathrm{E}_2(2\tilde{s}_2) \tag{26}$$

Thus, line segments must generate the remainder position displacement in order to satisfy position constraints:

$$\Delta\mathbf{p} = \mathbf{p}_G - \mathbf{p}_S - \mathbf{p}_{E_1,E_2} \tag{27}$$

As a consequence, lengths of line segments can be computed as:

$$\mathbf{L} = \left[\mathbf{T}_S\ \mathbf{T}_{G,1}\ \mathbf{T}_G\right]^\dagger \Delta\mathbf{p} \tag{28}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse, $\mathbf{T}_S := [1\ 0\ 0]^T$ and

$$\mathbf{T}_{G,1} := \begin{bmatrix} \cos(\theta_{G,1})\cos(\Psi_{G,1}) \\ \cos(\theta_{G,1})\sin(\Psi_{G,1}) \\ -\sin(\theta_{G,1}) \end{bmatrix} \tag{29}$$

Obviously, when the rank of $\left[\mathbf{T}_S\ \mathbf{T}_{G,1}\ \mathbf{T}_G\right]$ is not 3, i.e., the matrix has one or two singular values close to zero, it implies that line segments are redundant. In those cases, we can force to zero one of the line segments and compute the others. In particular, we propose to force to zero L_1 and L_3 before forcing to zero L_2, details left to the reader.

B. SHORTEST CURVE

The aim now is to compute the curve with the shortest length, being $\angle\mathbf{q}_{G,1}$ decision variables of an optimization process:

$$\begin{aligned} \angle\mathbf{q}^*_{G,1} = \arg\min_{\angle\mathbf{q}_{G,1}} \quad & L_1 + 2\tilde{s}_1 + L_2 + 2\tilde{s}_2 + L_3, \\ \text{subject to} \quad & L_1, L_2, L_3 \geq 0 \end{aligned} \tag{30}$$

Note that values of L_1, L_2, L_3, $\tilde{s}_1$ and $\tilde{s}_2$ depend on the deflection angle as discussed in the previous section. Since this procedure looks for solutions of maximum sharpness in torsion or curvature of the clothoid arcs, the curve resulting after solving (30) will produce the shortest length that joins the initial and goal configurations without discontinuities. Also note that the positiveness constraints imposed to L_1, L_2 and L_3 ensure that the obtained curve will imply no maneuvers, and thus are suitable for fixed-wing UAVs.

Lemma 1: The optimal solution will imply that lengths for the first and third lines are zero.

Proof: For the unbounded problem, where $\rho_{max} = \infty$ and $\mu_{\max} = \infty$, the clothoid lengths have zero length, i.e.: $\tilde{s}_1 = 0$ and $\tilde{s}_2 = 0$, which implies an instantaneous change of orientation between the line segments. In that case, it is obvious that the optimal solution joining two points in the Cartesian space is $\angle\mathbf{q}^*_{G,1} = [\arctan(\frac{\Delta\mathbf{p}_y}{\Delta\mathbf{p}_x})\ \arctan(\frac{-\Delta\mathbf{p}_z}{\sqrt{\Delta\mathbf{p}_x^2+\Delta\mathbf{p}_y^2}})]^T$, which implies that the lengths of the line segments are $L_1 = 0$, $L_2 = |\Delta\mathbf{p}|$ and $L_3 = 0$. The optimal solution of the bounded problem will generate clothoids with the shortest possible length such as the angle between $\mathbf{T}_{G,1}$ and $\Delta\mathbf{p}$ is minimum, which implies that numerically $L_1 \to 0$ and $L_3 \to 0$, details left for brevity. ■

C. COMPUTATION TIME

In order to test the computational effort to generate a DCC3D curve, a total of 10^4 arbitrary goal positions and orientations have been used to analyse time performance. The computational time has been divided into two stages. In the first stage, the optimization process to find the intermediate configuration angle $\angle\mathbf{q}^*_{G,1}$ that minimizes the overall length, obtaining an average time of $\bar{t}_{\mathbf{q}^*_{G,1}} = 0.110$ s. The second timing analysis focuses on determining the computational time to get μ and ρ values for each Elementary segment, $E1$ and $E2$. The mean time to compute such parameters is $\bar{t}_E = 0.0024$ s.

The computer used for this analysis has the following specifications: CPU Intel i5-9400F 4.100GHz, GPU NVIDIA GeForce GT 610 and 8 GB DDR4 memory, under an Ubuntu 18.04.4 LTS x86_64 OS with Kernel 5.4.0-120-generic.

D. CASE STUDY

To achieve a better understanding of the results obtained, a specific case study with three variants is described next. Figure 7 shows an example of DCC3D curves joining the initial configuration $\mathbf{q}_S = [0\ 0\ 0\ 0\ 0]^T$ with a goal configuration $\mathbf{q}_G = [170\ 120\ 90\ \pi/4\ \pi/6]^T$. The aim is to describe the behavior of the DCC3D curve by setting three different

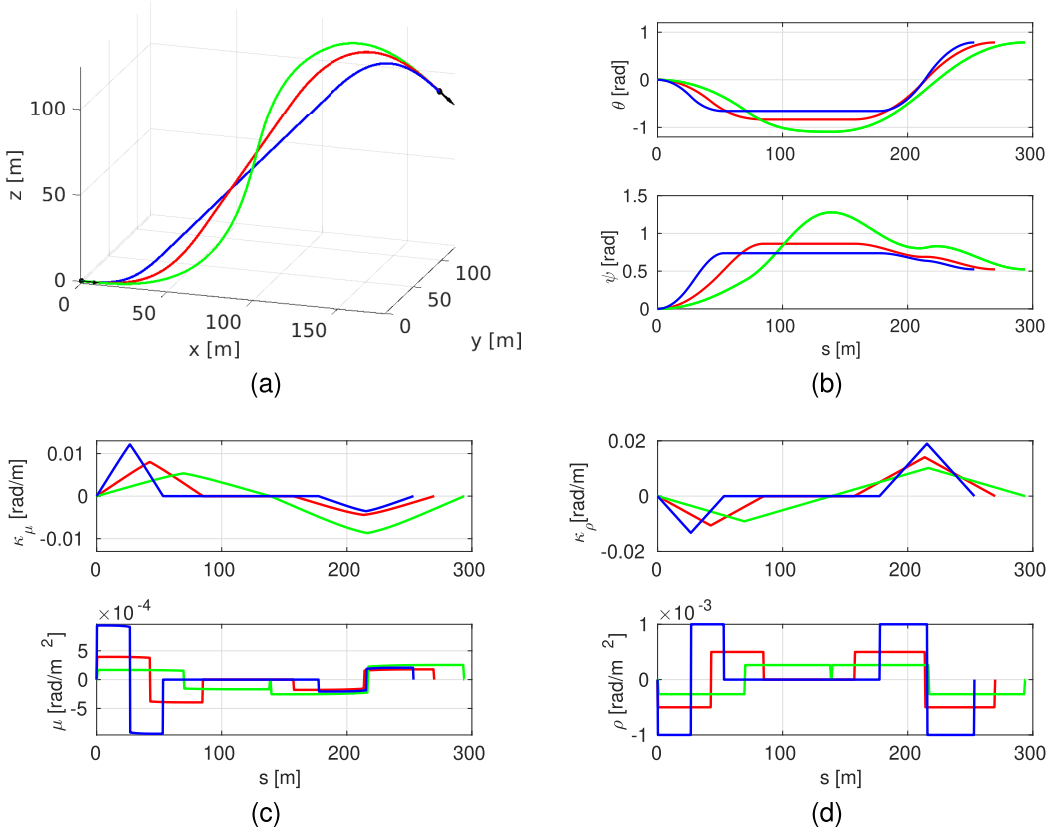

FIGURE 7. DCC3D curve with final orientation $\theta_G = \pi/4$ rad and $\Psi_G = \pi/6$ rad, final position $p_G = [170\ 120\ 90]^T$ m, with $\rho_{max} = \mu_{max} = 0.001$ (blue), $\rho_{max} = \mu_{max} = 0.0005$ (red), and $\rho_{max} = \mu_{max} = 0.00025$ (green). (a) Smooth path 3D, (b) Euler angles, (c) curvature profile and its sharpness in the orthogonal plane *XY*, (d) Curvature profile and its sharpness in the orthogonal plane *XZ*.

values for the maximum curvature sharpness and torsion sharpness. Figure 7(a) shows the curves obtained in 3D space. In Figure 7(b), the Euler orientation angles along the curves length are shown, while in Figures 7(c) and 7(d), the curvature and torsion profiles together with their sharpness are depicted.

As expected, lower values of maximum curvature sharpness and torsion sharpness yield longer and smoother ECb3D curves. That implies shorter straight lines and, consequently, longer DCC3D curves. It is interesting to remark that, reaching positions too close would imply curvy trajectories with a very pronounced S shape (or even an 8 shape) or a complete loop, i.e. turning to the opposite direction. If we do not seek for such type of curves, then the target configuration must be consistent with the sharpness constraints and the initial deflection angle can be set to $\angle \mathbf{q}_{G,1}^{initial} = [\arctan(\frac{y_G}{x_G})\ \arctan(\frac{-z_G}{\sqrt{x_G^2+y_G^2}})]^T$.

VI. FLIGHT SIMULATION EXPERIMENTS AND RESULTS

In order to realistically reproduce the Kadett 2400 fixed-wing UAV dynamic model, a setup with the Flight-Gear 2019 simulator and Matlab R2020b has been used for the low-level control of the different subsystems shown in Figure 8. For this experimentation the same computer as in Section V-B has been used.

The UAV has four actuators δ_e (elevation), δ_a (aileron), δ_r (rudder) and δ_{th} (throttle), which are controlled by three PID regulators implementing low-level control of the yaw Ψ^* and pitch θ^* angles, as well as the ground speed vel^*. The yaw and pitch references are obtained from the yaw and pitch angles of the designed smooth curve, while the forward velocity reference is kept constant at $vel^* = 18$ m/s throughout the simulation. Notice that in our implementation the aileron and rudder angles are actuated by the same controller, which directly affects the yaw orientation.

As mentioned before in the article, a nice characteristic of the DCC3D curve is to maintain zero curvature and torsion values, both at the beginning and at the end of the path. Therefore, a concatenation of DCC3D curves can be easily performed, producing a smooth flight path without any in-flight lift loss. Figure 9 shows an example of a smooth flight, where three DCC3D paths have been concatenated, being the maximum values of Cb3D sharpness parameters $\mu_{max} = 0.001$ rad/m^2 and $\rho_{max} = 0.001$ rad/m^2. Without loss of generality, the path begins at initial configuration $\mathbf{q}_S$ (the

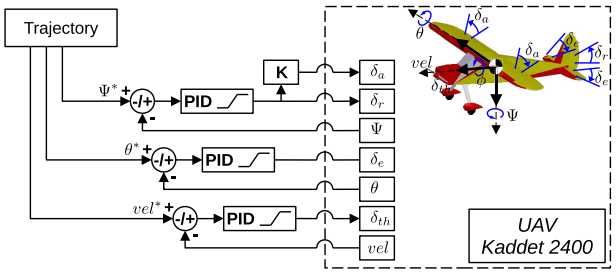

FIGURE 8. Kadett 2400 control model with 3 inputs to control roll, yaw and pitch which modify the aerodynamic surfaces δ_e (elevator), δ_a (aileron), δ_r (rudder).

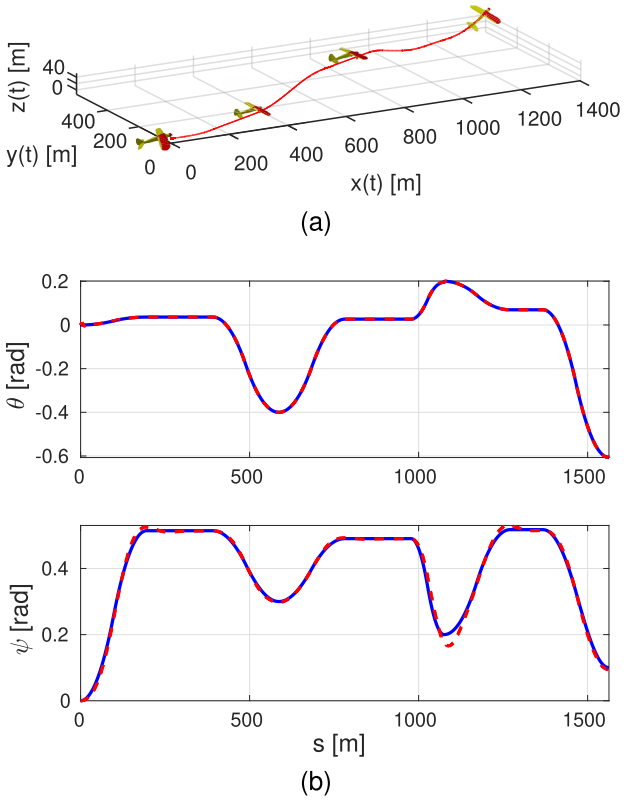

FIGURE 9. DCC3D curve and flight simulation with four concatenated configurations.

UAV in horizontal flight), passes through two intermediate configurations $\mathbf{q}_A$ and $\mathbf{q}_B$, and finishes at goal configuration $\mathbf{q}_G$:

$$\mathbf{q}_S = \begin{bmatrix}0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0\end{bmatrix}^T$$
$$\mathbf{q}_A = \begin{bmatrix}480.0 & 200.0 & 20.0 & 0.0 & -0.4 & 0.3\end{bmatrix}^T$$
$$\mathbf{q}_B = \begin{bmatrix}1000.0 & 440.0 & 28.0 & 0.0 & 0.2 & 0.2\end{bmatrix}^T$$
$$\mathbf{q}_G = \begin{bmatrix}1400.0 & 600.0 & 56.0 & 0.0 & -0.6 & 0.1\end{bmatrix}^T$$

Figure 9(a) shows the reference path followed by the UAV. Figure 9(b) shows the Euler angles, where the reference angles of the DCC3D curve are the blue lines, and the UAV angles generated by the in-flight tracking system are the dashed red lines. In Figure 9 it can also be seen that the errors during the curve tracking in yaw and pitch angles are small, being the mean square error of $\epsilon_\theta = 0.00042$ rad and $\epsilon_\Psi = 0.006$ rad, respectively, where, $\epsilon_\theta = \frac{1}{s_T}\int_0^{s_T}(\theta^* - \theta)\mathrm{d}s$ and $\epsilon_\Psi = \frac{1}{s_T}\int_0^{s_T}(\Psi^* - \Psi)\mathrm{d}s$. Finally, it is important to mention that the flight time is $t = 86.68$ s with a full flight path of $s_T = 1560.28$ m.

VII. CONCLUSION

The work developed in this paper presents a novel 3D smooth path planner based on a concatenation of Clothoid-based 3D curves (Cb3D) and straight line segments. The planner solves the problem of joining two arbitrary configurations (position and orientation) in 3D space (without loss of continuity and smoothness), making it suitable for navigation of fixed-wing UAVs.

To generate DCC3D paths, a new 3D smooth curve called Elementary Clothoid-based 3D Curve (ECb3D) has been presented. An ECb3D can achieve any orientation with zero curvature and torsion values both at initial and final configurations. That feature allows ECb3D paths to be used as primitives for path generation, since it is possible to smoothly concatenate several ECb3D paths and straight lines without loss of curvature and torsion continuity.

The computational time of the curve, including finding the curve with the shortest distance takes, in average about 0.1 on an Intel i5 (seed details in section 5). Hence, the fast computation of the proposed DCC3D, allows to generate 3D smooth paths in real-time.

As further work, we aim to include the proposed trajectory in global planners such as Randomized Path Planner (RPP), Probabilistic Road Map Method (PRM), Rapidly Exploring Random Tree (RRT). A different use of DCC3D curve would be the development of a local path planner for obstacle avoidance in 3D maneuvers. Finally, the same methodology could be implemented using other transition curves in order to compare the performance against the Cb3D curves used in this work.

APPENDIX

A. CLOTHOID-BASED 3D CURVE

Without loss of generality it is assumed that the Cb3D starts from the origin, therefore, the generic parameterization of the curve is defined by the vector $\mathbf{p}$, being $\mathbf{C}(s, \mathbf{p})$. Thus, $\mathbf{p}$ contains the vector of design parameters of the curve, such that $\mathbf{p} := \{\mu, \rho\}$ represent the sharpness parameters of the 2D clothoids contained in the XY and XZ planes. Therefore, the objective is to compute $\mathbf{p}^* := \{\mu^*, \rho^*\}$ given a target tangent vector $\mathbf{T}^*$ to be achieved.

Specifically, the curve C2D contained in the XY plane is computed as:

$$\mathbf{C}_1(q, \mu) := \begin{bmatrix}\mathcal{C}(q, \mu) \\ \mathcal{S}(q, \mu)\end{bmatrix} \in \mathbb{R}^2 \tag{31}$$

where, q is the arc length, while:

$$\mathbf{C}_2(\omega, \rho) := \begin{bmatrix} \mathcal{C}(\omega, \rho) \\ \mathcal{S}(\omega, \rho) \end{bmatrix} \in \mathbb{R}^2 \tag{32}$$

corresponds to the second C2D, contained in the XZ-plane, where the arc length is defined by ω with sharpness ρ. Consequently, the curve Cb3D presents a solution combining both clothoids, such that:

$$\mathbf{C}(s, \mathbf{p}) \equiv \mathbf{C}(s, \rho, \mu) := \begin{cases} \mathcal{C}(\mathcal{C}(s, \rho), \mu) \\ \mathcal{S}(\mathcal{C}(s, \rho), \mu) \\ -\mathcal{S}(s, \rho) \end{cases} \tag{33}$$

The orientation angles of the reference system associated to the Cb3D curve are given by:

$$\theta(s, \rho) := \frac{\rho}{2} s^2 \tag{34}$$

$$\Psi(s, \mu, \rho) := \frac{\mu}{2} \mathcal{C}^2(s, \rho) \tag{35}$$

$$\phi(s, \mu, \rho) := -\arcsin\left(\frac{\theta'(s, \rho)}{\sqrt{\omega(s, \mu, \rho)^2 + \theta'(s, \rho)^2}} \right) \tag{36}$$

being, $\omega(s, \mu, \rho) := \cos(\theta(s, \rho))\Psi'(s, \mu, \rho)$, $\theta'(s, \rho) := \mathrm{d}\theta(s, \rho)/\mathrm{d}s$ and $\Psi'(s, \mu, \rho) := \mathrm{d}\Psi(s, \mu, \rho)/\mathrm{d}s$. Similarly, the tangent, normal and binormal vectors of the reference frame associated to the curve are defined by:

$$\mathbf{T}(s, \mu, \rho) := \begin{bmatrix} c\cos(\Psi(s, \mu, \rho))\cos(\theta(s, \rho)) \\ \sin(\Psi(s, \mu, \rho))\cos(\theta(s, \rho)) \\ -\sin(\theta(s, \rho)) \end{bmatrix} \tag{37}$$

$$\mathbf{N}(s, \mu, \rho) := \frac{\mathbf{T}'(s, \mu, \rho)}{\|\mathbf{T}'(s, \mu, \rho)\|} \tag{38}$$

$$\mathbf{B}(s, \mu, \rho) := \mathbf{T}(s, \mu, \rho) \times \mathbf{N}(s, \mu, \rho). \tag{39}$$

Consequently, the target tangent vector $\mathbf{T}^*$, is defined by the angles $\theta^* \in [-\pi/2, \pi/2]$ rad, and $\Psi^* \in [-\pi, \pi]$ rad, so that, for a given value of arc length $s > 0$, the clothoid parameters will be computed from (34) and (35):

$$\rho^* = \frac{2\theta^*}{s^2}, \quad \mu^* = \frac{2\Psi^*}{\mathcal{C}^2(s, \rho^*)}. \tag{40}$$

ACKNOWLEDGMENT

(Gloria Vanegas, Leopoldo Armesto, and Vicent Girbés-Juan contributed equally to this work.)

REFERENCES

[1] R. Chaurasia and V. Mohindru, "Unmanned aerial vehicle (UAV): A comprehensive survey," in *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*, V. Mohindru, Y. Singh, R. Bhatt, and A. K. Gupta, Eds. 2021, pp. 1–27, doi: 10.1002/9781119769170.ch1.

[2] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Jan. 2019.

[3] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang, "A survey on 5G millimeter wave communications for UAV-assisted wireless networks," *IEEE Access*, vol. 7, pp. 117460–117504, 2019, doi: 10.1109/ACCESS.2019.2929241.

[4] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, vol. 158, pp. 54–64, Dec. 2018, doi: 10.1016/j.knosys.2018.05.033.

[5] T. M. Cabreira, L. B. Brisolara, and R. F. Paulo, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, pp. 1–38, 2019, doi: 10.3390/drones3010004.

[6] C. A. Wargo, G. C. Church, J. Glaneueski, and M. Strout, "Unmanned aircraft systems (UAS) research and future analysis," in *Proc. IEEE Aerosp. Conf.*, Mar. 2014, pp. 1–16, doi: 10.1109/AERO.2014.6836448.

[7] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective UAV path planning for search and rescue," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5569–5574, doi: 10.1109/ICRA.2017.7989656.

[8] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, "An overview of emerging results in cooperative UAV control," in *Proc. IEEE Conf. Decis. Control*, vol. 1, Dec. 2004, pp. 602–607, doi: 10.1109/CDC.2004.1428700.

[9] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV)," *Biosyst. Eng.*, vol. 108, no. 2, pp. 174–190, Feb. 2011, doi: 10.1016/j.biosystemseng.2010.11.010.

[10] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Control*, vol. 48, pp. 147–164, 2019, doi: 10.1016/j.arcontrol.2019.10.001.

[11] S. Huang and R. S. H. Teo, "Computationally efficient visibility graph-based generation of 3D shortest collision-free path among polyhedral obstacles for unmanned aerial vehicles," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2019, pp. 1218–1223, doi: 10.1109/ICUAS.2019.8798322.

[12] J. Velasco-Carrau, S. García-Nieto, J. Salcedo, and R. H. Bishop, "Multi-objective optimization for wind estimation and aircraft model identification," *J. Guid., Control, Dyn.*, vol. 39, no. 2, pp. 372–389, 2016, doi: 10.2514/1.G001294.

[13] J. Shen, Y. Su, Q. Liang, and X. Zhu, "Calculation and identification of the aerodynamic parameters for small-scaled fixed-wing UAVs," *Sensors*, vol. 18, no. 1, pp. 1–18, 2018, doi: 10.3390/s18010206.

[14] K. Yang and S. Sukkarieh, "3D smooth path planning for a UAV in cluttered natural environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 794–800, doi: 10.1109/IROS.2008.4650637.

[15] K. Yang and S. Sukkarieh, "Real-time continuous curvature path planning of UAVS in cluttered environments," in *Proc. 5th Int. Symp. Mechatronics Its Appl.*, May 2008, pp. 1–6, doi: 10.1109/ISMA.2008.4648836.

[16] D. González, J. Pérez, R. Lattarulo, V. Milanés, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," in *Proc. 17th IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1430–1435, doi: 10.1109/ITSC.2014.6957887.

[17] A. A. Neto, D. G. Macharet, and M. F. M. Campos, "Feasible RRT-based path planning using seventh order Bézier curves," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1445–1450, doi: 10.1109/IROS.2010.5649145.

[18] Q. Liu, Y. Zhang, M. Li, Z. Zhang, N. Cao, and J. Shang, "Multi-UAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network," *IEEE Access*, vol. 9, pp. 124670–124681, 2021, doi: 10.1109/ACCESS.2021.3109879.

[19] Z. Wang, Y. Li, and W. Li, "An approximation path planning algorithm for fixed-wing UAVs in stationary obstacle environment," in *Proc. 33rd Chin. Control Conf.*, Jul. 2014, pp. 664–669, doi: 10.1109/ChiCC.2014.6896704.

[20] L. Tan, Y. Liu, X. Lian, and X. Ren, "Optimization of UAV 3D trajectory smoothing based on improved PH curves," in *Proc. 7th Int. Conf. Intell. Comput. Signal Process. (ICSP)*, Apr. 2022, pp. 1219–1222, doi: 10.1109/ICSP54964.2022.9778454.

[21] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Clear and smooth path planning," *Appl. Soft Comput.*, vol. 32, pp. 568–579, Jul. 2015, doi: 10.1016/j.asoc.2015.04.017.

[22] L. Huang, H. Qu, P. Ji, X. Liu, and Z. Fan, "A novel coordinated path planning method using K-degree smoothing for multi-UAVs," *Appl. Soft Comput.*, vol. 48, no. 1, pp. 182–192, 2016, doi: 10.1016/j.asoc.2016.06.046.

[23] X. Wu, W. Bai, Y. Xie, X. Sun, C. Deng, and H. Cui, "A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs," *Appl. Soft Comput.*, vol. 73, pp. 735–747, Dec. 2018, doi: 10.1016/j.asoc.2018.09.011.

[24] L. Armesto, V. Girbés, A. Sala, M. Zima, and V. Šmídl, "Duality-based nonlinear quadratic control: Application to mobile robot trajectory-following," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1494–1504, Jul. 2015, doi: 10.1109/TCST.2014.2377631.

[25] P. Liu and Q. Hu, "Gaussian pseudospectral optimization method with smoothing penalty function path constraint handling for UAV obstacle avoidance planning," in *Proc. China Autom. Congr. (CAC)*, Oct. 2021, pp. 7530–7534, doi: 10.1109/CAC53003.2021.9728229.
[26] B. Kaiser and A. Verl, "Planning of curvature-optimal smooth paths for industrial robots using neural networks," in *Proc. 4th Int. Conf. Artif. Intell. Industries (AII)*, Sep. 2021, pp. 10–15, doi: 10.1109/AI4I51902.2021.00011.
[27] S. Gim, L. Adouane, S. Lee, and J.-P. Dérutin, "Clothoids composition method for smooth path generation of car-like vehicle navigation," *J. Intell. Robotic Syst.*, vol. 88, no. 1, pp. 129–146, Oct. 2017, doi: 10.1007/s10846-017-0531-8.
[28] P. F. Lima, M. Trincavelli, J. Martensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 2983–2990, doi: 10.1109/ECC.2015.7330991.
[29] J. A. Silva and V. Grassi, "Clothoid-based global path planning for autonomous vehicles in urban scenarios," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2018, pp. 4312–4318, doi: 10.1109/ICRA.2018.8461201.
[30] D. J. Kim and C. C. Chung, "Automated perpendicular parking system with approximated clothoid-based local path planning," *IEEE Control Syst. Lett.*, vol. 5, no. 6, pp. 1940–1945, Dec. 2021, doi: 10.1109/LCSYS.2020.3044254.
[31] Y. Chen, Y. Cai, J. Zheng, and D. Thalmann, "Accurate and efficient approximation of clothoids using Bézier curves for path planning," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1242–1247, Oct. 2017, doi: 10.1109/TRO.2017.2699670.
[32] G. Vanegas, F. Samaniego, V. Girbes, L. Armesto, and S. Garcia-Nieto, "Smooth 3D path planning for non-holonomic UAVs," in *Proc. 7th Int. Conf. Syst. Control (ICSC)*, Oct. 2018, pp. 1–6, doi: 10.1109/ICoSC.2018.8587835.
[33] E. D. Lambert, R. Romano, and D. Watling, "Optimal smooth paths based on clothoids for car-like vehicles in the presence of obstacles," *Int. J. Control, Autom. Syst.*, vol. 19, no. 6, pp. 2163–2182, Jun. 2021, doi: 10.1007/s12555-020-0179-1.
[34] S. Liyang, H. Yu, C. Xuezhi, J. Changhao, and H. Miaohua, "Path planning based on clothoid for autonomous valet parking," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2020, pp. 389–393, doi: 10.1109/ICCWAMTIP51612.2020.9317391.
[35] X. Huang, F. Zhao, T. Tao, and X. Mei, "A newly developed corner smoothing methodology based on clothoid splines for high speed machine tools," *Robot. Comput.-Integr. Manuf.*, vol. 70, Aug. 2021, Art. no. 102106, doi: 10.1016/j.rcim.2020.102106.
[36] V. Girbés, G. Vanegas, and L. Armesto, "Clothoid-based three-dimensional curve for attitude planning," *J. Guid., Control, Dyn.*, vol. 42, no. 8, pp. 1886–1898, Aug. 2019, doi: 10.2514/1.G003551.
[37] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robot. Auto. Syst.*, vol. 86, pp. 13–28, Dec. 2016, doi: 10.1016/j.robot.2016.08.001.
[38] S. Hota and D. Ghose, "Optimal geometrical path in 3D with curvature constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 113–118, doi: 10.1109/IROS.2010.5653663.
[39] S. Hota and D. Ghose, "Optimal path planning for an aerial vehicle in 3D space," in *Proc. 49th IEEE Conf. Decision Control (CDC)*, Dec. 2010, pp. 4902–4907, doi: 10.1109/CDC.2010.5717246.
[40] S. Upadhyay and A. Ratnoo, "Smooth path planning for passages with heading and curvature discontinuities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 2672–2677, doi: 10.1109/IROS.2015.7353742.
[41] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, 2nd ed. New York, NY, USA: Dover, 2016.
[42] R. Levien. (2008). *The Euler Spiral: A Mathematical History*. [Online]. Available: http://raph.levien.com/phd/euler_hist.pdf
[43] E. Lambert, R. Romano, and D. Watling, "Optimal path planning with clothoid curves for passenger comfort," in *Proc. 5th Int. Conf. Vehicle Technol. Intell. Transp. Syst. (VEHITS)*, vol. 1. SciTePress, May 2019, pp. 609–615.
[44] M. E. Vázquez-Méndez and G. Casal, "The clothoid computation: A simple and efficient numerical algorithm," *J. Surveying Eng.*, vol. 142, no. 3, pp. 1–18, Aug. 2016, doi: 10.1061/(ASCE)SU.1943-5428.0000177.
[45] V. Girbés, L. Armesto, and J. Tornero, "Path following hybrid control for vehicle stability applied to industrial forklifts," *Robot. Auto. Syst.*, vol. 62, no. 6, pp. 910–922, Jun. 2014, doi: 10.1016/j.robot.2014.01.004.
[46] G. Harary and A. Tal, "3D Euler spirals for 3D curve completion," in *Proc. 26th Annu. Symp. Comput. Geometry*, vol. 45, no. 3, Jun. 2012, pp. 393–402.
[47] A. Scheuer and T. Fraichard, "Planning continuous-curvature paths for car-like robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 1996, pp. 1304–1311.

GLORIA VANEGAS received the B.Eng. degree in electronic and computation engineering from the Escuela Superior Politécnica de Chimborazo (ESPOCH), Ecuador, in 2008, and the M.Sc. degree in robotics and automation from the Universidad Carlos III de Madrid (UC3M), Spain, in 2013. She is currently pursuing the Ph.D. degree in automation, robotics, and industrial computer science with the Universitat Politècnica de València (UPV), Spain. She is currently a Professor of industrial robotics with the Universidad Internacional de Ecuador (UIDE), Ecuador. Her research interests include mobile robotics, smooth path planning, and obstacle avoidance among others.

LEOPOLDO ARMESTO received the B.Sc. degree in electronic engineering, the M.Sc. degree in control systems engineering, and the Ph.D. degree in automation and industrial computer science from the Universitat Politècnica de València. He is currently a member of the Department of Systems Engineering and Control, Universitat Politècnica de València. His current research interests include mobile robotics, optimal control, advanced driving assistance systems, and reinforcement learning.

VICENT GIRBÉS-JUAN (Member, IEEE) received the B.Eng. and M.Sc. degrees in industrial electronics and control from the UPV, in 2009 and 2011, respectively, and the Ph.D. degree in automation, robotics and industrial computer science, in 2016. He was worked at the Robotics and Automation Research Group, UPV, from 2009 to 2019. He has been a Visiting Researcher with the University of Manchester and Imperial College London. He is currently an Assistant Professor with the Department of Electronic Engineering, Universitat de València (UV). His research interests include intelligent and adaptive robots, autonomous driving, human–robot interaction, and Bayesian optimization.

JOAQUÍN PÉREZ (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from UPV, in 2004 and 2009, respectively. He is currently a Lecturer with the Department of Electronic Engineering, UV, since 2019. Under the topic of optical communications and electrical telecommunications systems he has participated in international research projects, e.g., FP6-IST-UROOF, FP7-ICT-UCELLS, H2020 EU COST IC0802, IC1101 or NewFocus CA19111; and national projects as RAMONET, AICAR, MEMES, ULTRADEF, CARACTER or SEQMUST; and published more than 70 works in recognized international journals and conferences. His current research interests include advanced optical communications in OWC, vehicular communication systems, and secure systems.