Trajectory Planning for UAVs Based on Interfered Fluid Dynamical System and Bézier Curves

Trajectory planning for UAVs based on Interfered Fluid Dynamical System and Bézier curves

Davide Celestini[1], Stefano Primatesta[1] and Elisa Capello[2] .

***Abstract*—In this paper, a 3D trajectory planner for Unmanned Aerial Vehicles (UAVs) based on Interfered Fluid Dynamical System (IFDS) and Bézier curves is introduced. The proposed strategy joints the potentialities of IFDS with the use of Bézier curves to obtain optimized trajectories with continuous curvature. While IFDS computes an initial trajectory to safely avoid fixed and dynamic obstacles, Bézier curve optimization generates a trajectory satisfying kinematic constraints. This combination is computationally efficient for online applications with limited hardware and a smoothed path is obtained, for safe and flyable trajectories. Simulations are performed for a fixed-wing UAV in a complex and dynamic environment.**

***Index Terms*—Constrained Motion Planning, Aerial Systems, Mechanics and Control, Collision Avoidance**

I. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) have been widely used to perform several applications even in complex and critical scenarios. Nevertheless, the development of autonomous guidance systems for UAVs operating in these scenarios represents a challenging problem. To this aim, trajectory planning represents one of the most important elements to achieve high levels of autonomy with UAVs. Such planner must be capable of computing safe and feasible trajectories avoiding fixed and dynamic obstacles, as well as satisfying kinematic constraints.

Often, trajectory planning works jointly with path planning. In this context, a path planner provides a global path to be followed by a trajectory planner. Sampling-based techniques, e.g. Probabilistic Road-Map (PRM) and Rapidly-Exploring Random Tree (RRT), and searching-based methods, e.g. Dijkstra's algorithm and A*, are widely exploited for the identification of this collision-free global path [1], [2]. However, such reference path may not be flyable by the UAV because of kinematic constraints and unexpected dynamic obstacles. Hence, the role of the trajectory planner is to compute a flyable trajectory following the global path, avoiding the obstacles and satisfying the kinematic constraints of the UAV at the same time. To this aim, algorithms based on mathematical models, Potential Fields methods, linear and non-linear Model Predictive Control (MPC) and optimization techniques constitute the state of the art [1], [2]. Among these methods, polynomial curves represent a popular choice due to the efficiency and simplicity in controlling their shape modifying their control points [3], [4], [5]. Bézier curves, in particular, are greatly adopted due to the intuitive correlation between their control point locations and the overall shape of the curve [3], [4]. Several approaches based on Bézier curves are proposed in the literature to generate smooth trajectories for UAVs and ground robots in 2D environments [6], [7], [8], [9]. In [10] and [11], low-order 3D Bézier curves are joined together to create a smooth trajectory following a given waypoint-based path in a complex scenario. Anyway, UAV's constraints are not fully included in the problem. In [12], higher-order 3D Bézier curves are exploited to compute a smooth feasible trajectory, but obstacles are not taken into account.

An interesting method is proposed in [13], where quartic Bézier curves and quadratic speed profiles are used to guide a ground vehicle along a predefined route, optimizing local trajectories previously estimated through an Artificial Potential Fields (APF). APF grants efficiency and obstacle avoidance capabilities to the method, but requires ad-hoc modifications to deal with non-spherical objects [14], [15]. Similarly, the promising approach of Interfered Fluid Dynamical Systems (IFDS) is a fluid-based methodology with high computational efficiency. IFDS, however, is able to safely handle both static and dynamic obstacles with different shapes even in 3D environments [16], [17], [18], [19].

In this work, therefore, we propose a trajectory planning strategy combining IFDS and Bézier curves to follow a waypoint-based path. First, IFDS is used to compute an initial estimate of the local trajectory. Then, quartic Bézier curves are exploited to obtain an optimal local trajectory satisfying kinematic constraints. Unlike [13], we extend the optimization with Bézier curves to 3D environments. This approach skillfully combines the advantages of the two methods and guides the UAV along a given global path with a safe and smooth trajectory, as well as avoiding dynamic obstacles.

The paper is organized as follows. Sec. II includes the problem statement and defines the UAV nonlinear model. In Sec. III the proposed trajectory planning strategy, the IFDS and the Bézier curves are detailed. Simulation results are presented in Sec. IV. Conclusions are drawn in Sec. V.

II. Problem Statement

In this paper, we aim at solving a trajectory planning problem for UAVs following a waypoint-based reference path. Given a list of waypoints computed by any path planner,

Manuscript received: February 24, 2022; Revised: April 24, 2022; Accepted: June 26, 2022. This paper was recommended for publication by T. Asfour and H. Kurniawatie upon evaluation of the Associate Editor and Reviewers' comments. *(Corresponding author: S. Primatesta.)*

[1]D. Celestini and S. Primatesta are with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino (email: davide.celestini@polito.it; stefano.primatesta@polito.it).

[2]E. Capello is with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino and CNR-IEIIT (email: elisa.capello@polito.it).

Digital Object Identifier (DOI): see top of this page.

the trajectory planner (TP) computes an optimal and safe reference profile for the UAV flight control system in order to follow each waypoint sequentially. Moreover, the TP faces an environment characterized by static and dynamic obstacles, which requires a periodic update of the references. Hence, the adopted methodology must satisfy the following conditions: *(i)* be executed online; *(ii)* plan a local trajectory with a continuous curvature w.r.t. the current UAV state; *(iii)* evaluate the trajectory along a time horizon; and, *(iv)*, have obstacle avoidance capabilities to handle fixed and dynamic obstacles.

The TP problem is particularly critical for the case of fixed-wing UAVs, whose motion is typically characterized by high speed and strict kinematic constraints. Inspired by [20], the nonlinear model of a fixed-wing UAV is defined as follow

$$\begin{cases} \dot{x}(t) = V(t)\cos(\psi(t))\cos(\gamma(t)) \\ \dot{y}(t) = V(t)\sin(\psi(t))\cos(\gamma(t)) \\ \dot{z}(t) = V(t)\sin(\gamma(t)) \\ \dot{\psi}(t) = V(t)\mathcal{K}_H(t) \\ \dot{\gamma}(t) = V(t)\mathcal{K}_V(t) \\ \dot{V}(t) = a(t) \end{cases}, \tag{1}$$

where $\mathbf{P} = [x, y, z]^T$ is the UAV position in an inertial reference system, V and a are flight speed and acceleration, ψ and γ are heading and flight path angles. $\mathcal{K}_H = \frac{1}{R_H}$ and $\mathcal{K}_V = \frac{1}{R_V}$ are curvatures in the horizontal and vertical plane, reciprocals of the corresponding curvature radii.

The main kinematic constraints affecting the nonlinear model in (1) are summarized as

$$R_{H_{min}}(V) = \frac{V^2}{g\sqrt{n_{max}^2 - 1}}, \tag{2}$$

$$V_{min} \leq V(t) \leq V_{max}, \tag{3}$$

$$a_{min} \leq a(t) \leq a_{max}, \tag{4}$$

$$\gamma_{min} \leq \gamma(t) \leq \gamma_{max}, \tag{5}$$

where (2) is caused by the aircraft maximum load factor n_{max}, while (3), (4) and (5) are related to aerodynamic features and thrust limits of the UAV. Note that constraint (2) dynamically varies, since V is varying during flight. These limitations represent a great challenge in the trajectory planning problem.

III. Trajectory Planning based on IFDS and Bézier Curves

In order to solve the trajectory planning problem of Sec. II, we propose the efficient multi-step approach defined in Alg. 1. Given a waypoint-list path, the algorithm is executed until the UAV reaches the last waypoint.

At first, the TP gets the updated information regarding the environment, the UAV state $X_0 = (\mathbf{P}_0, \psi_0, \gamma_0)$, its velocity V_0 and acceleration a_0. It also computes the local goal point $\mathbf{P}_g = [x_g, y_g, z_g]^T$, a point moving along the waypoint-based path with a desired cruise speed V_c. Its speed is set to V_{max} when its distance w.r.t. $\mathbf{P}_0$ is smaller than $d_{min} = 6R_{H_{min}}(V_0)$ to grant a minimum time span for the trajectory planning and enlarge d_{min} as V_0 increases. Then, the efficient IFDS method is exploited to compute a forward simulation in which the UAV moves towards $\mathbf{P}_g$ avoiding static and dynamic obstacles. The time span for this simulation is ΔT_{sim}, adaptively computed as $2\frac{R_{H_{min}}(V_0)}{V_0}$ in order to be proportional to the UAV speed V_0. Afterwards, Bezier curves are used to compute the optimal feasible path connecting the initial state X_0 with the final state of the forward simulation, called $X_4 = (\mathbf{P}_4, \psi_4, \gamma_4)$. Finally, the trajectory is obtained through the computation of the velocity profile. The resulting trajectory is followed over a fixed $\Delta T_{upd} \ll \Delta T_{sim}$. Hence, this iteration is repeated until the last waypoint is reached. In the following paragraphs, each step of Alg. 1 is detailed.

Algorithm 1: Main Trajectory Planning algorithm

Input: Waypoint-list path
repeat every ΔT_{upd}
  Update environment data, X_0, V_0, a_0 and $\mathbf{P}_g$;
  Compute IFDS forward simulation over ΔT_{sim};
  Compute optimized Bézier curve;
  Compute velocity profile;
  Execute trajectory only for ΔT_{upd};
until *last Waypoint is reached*;

A. Interfered Fluid Dynamical System

The Interfered Fluid Dynamical System (IFDS) solves the obstacle avoidance problem computing the streamlines of a fluid field flowing around stones. Without any threats in the environment, the planned path is a straight line towards the goal point. Otherwise, the streamlines of the interfered fluid field smoothly and safely guide the UAV towards the goal point flowing around all obstacles [16], as shown in Fig. 1. The IFDS method has been adopted by several studies in the literature, both considering static and dynamic obstacles. An exhaustive analysis of static IFDS can be found in [16], [17], [18], while in-depth details related to the dynamic version of the method are reported in [19], [21].

At first, the method calculates the initial velocity vector field $\mathbf{u}(\mathbf{P}) \in \mathbb{R}^3$ pointing towards the current position of the local goal point $\mathbf{P}_g$ as

$$\mathbf{u}(\mathbf{P}) = -V_c \frac{\mathbf{P} - \mathbf{P}_g}{\|\mathbf{P} - \mathbf{P}_g\|}, \tag{6}$$

with V_c being the desired cruise speed of the UAV.

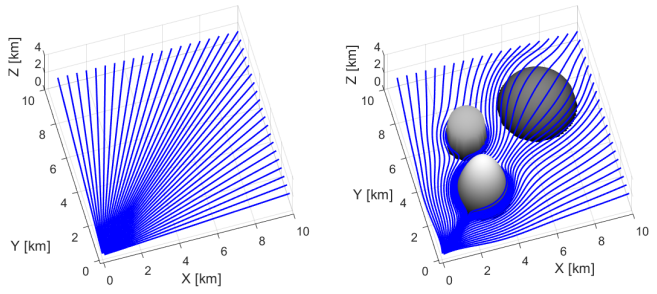

Fig. 1. On the left the streamlines characterizing the initial velocity field. On the right, the streamlines characterizing the interfered velocity field computed by the IFDS.

Then, assuming K obstacles in the environment exist, the overall permutation matrix $M(\mathbf{P})$ can be computed as

$$M(\mathbf{P}) = \sum_{k=1}^{K} \omega_k(\mathbf{P}) M_k(\mathbf{P}). \tag{7}$$

where $\omega_k \in \mathbb{R}, M_k \in \mathbb{R}^{3\times 3}$ are the normalized weighting coefficient and permutation matrix of each obstacle, which are computed as

$$\omega_k(\mathbf{P}) = \frac{\omega'_k(\mathbf{P})}{\sum_{i=1}^{K} \omega'_i(\mathbf{P})}, \tag{8}$$

$$\omega'_k(\mathbf{P}) = \begin{cases} 1 & K = 1 \\ \prod_{i=1, i\neq k}^{K} \frac{\Gamma_i(\mathbf{P})-1}{(\Gamma_i(\mathbf{P})-1)+(\Gamma_k(\mathbf{P})-1)} & K > 1 \end{cases}, \tag{9}$$

$$M_k(\mathbf{P}) = \left(1 - \frac{1}{|\Gamma_k(\mathbf{P})|^{\frac{1}{\rho_k}}}\right)\mathbb{I}_3 + - \frac{2\mathbf{n}_k(\mathbf{P})\mathbf{n}_k(\mathbf{P})^T}{|\Gamma_k(\mathbf{P})|^{\frac{1}{\rho_k}}\mathbf{n}_k(\mathbf{P})^T\mathbf{n}_k(\mathbf{P})}, \tag{10}$$

$$\Gamma_k(\mathbf{P}) = \left(\frac{x-x_k}{a_k}\right)^{2p_k} + \left(\frac{y-y_k}{b_k}\right)^{2q_k} + \left(\frac{z-z_k}{c_k}\right)^{2r_k}, \tag{11}$$

where $\Gamma_k(\mathbf{P}) \in \mathbb{R}$ is the modeling function of the k^{th} obstacle located in $[x_k, y_k, z_k]^T$, with dimensions (a_k, b_k, c_k) and form factors (p_k, q_k, r_k). $\rho_k > 0$ is the constant repulsive coefficient, $\mathbb{I}_3$ is the third-order identity matrix, $\mathbf{n}_k(\mathbf{P}) = \nabla\Gamma_k(\mathbf{P})$ is perpendicular to the obstacle surface and directed outward. For further details about static IFDS, see [17].

Further considerations are required to evaluate dynamic obstacles. For a clear and simple description, we assume spherical dynamic obstacles. At the beginning of the forward simulation, for each moving obstacle, current position $[x_k, y_k, z_k]_0^T$ and velocity $\mathbf{v}_k \in \mathbb{R}^3$ are used to compute a "prediction sphere" which contains the predicted motion over the time span ΔT_{sim}, as shown in Fig. 2. This sphere is centred in $[x_k, y_k, z_k]^T = [x_k, y_k, z_k]_0^T + \mathbf{v}_{k_0}\frac{\Delta T_{sim}}{2}$ and has radius $R = R_0 + \|\mathbf{v}_{k_0}\|\frac{\Delta T_{sim}}{2}$. In this way, the prediction sphere can be considered as a fixed object during the current forward simulation. Nevertheless, the overall transport velocity $\mathbf{v}(\mathbf{P}) \in \mathbb{R}^3$, which is the perturbation velocity field induced by the movement of the dynamic obstacles, must be considered and computed as

$$\mathbf{v}(\mathbf{P}) = \tilde{\mathbf{v}}_{\tilde{k}}(\mathbf{P}) \tag{12}$$

$$\tilde{k} = \arg\max_{k\in\{1\cdots K\}} \tilde{\mathbf{v}}_k(\mathbf{P}) \tag{13}$$

$$\tilde{\mathbf{v}}_k(\mathbf{P}) = \tilde{\omega}_k(\mathbf{P}) e^{\frac{-1}{\lambda_k}(\Gamma_k(\mathbf{P})-1)}\mathbf{v}_k \tag{14}$$

$$\tilde{\omega}_k(\mathbf{P}) = \frac{\omega_k(\mathbf{P})}{\max_{i\in\{1\cdots K\}}\omega_i(\mathbf{P})}, \tag{15}$$

where $\tilde{k}$ is the index of the obstacle causing the maximum weighted transport velocity $\tilde{\mathbf{v}}_k(\mathbf{P}) \in \mathbb{R}^3$, $\tilde{\omega}_k(\mathbf{P}) \in \mathbb{R}$ is the normalized dynamic weighting factor, $\lambda_k > 0$ determines how much the current velocity of the k^{th} obstacle $\mathbf{v}_k \in \mathbb{R}^3$ affects the field. For further details, refer to [19].

Finally, the interfered velocity vector field is computed as

$$\bar{\mathbf{u}}(\mathbf{P}) = M(\mathbf{P})\left(\mathbf{u}(\mathbf{P}) - \mathbf{v}(\mathbf{P})\right) + \mathbf{v}(\mathbf{P}). \tag{16}$$

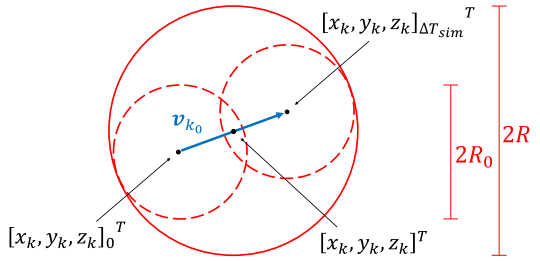

Fig. 2. Dashed spheres: on the left the initial location of the dynamic target; on the right the final estimated location after ΔT_{sim}. Lined sphere: prediction sphere for the moving obstacle for the current forward simulation.

Since $\bar{u}(\mathbf{P}) = \|\bar{\mathbf{u}}(\mathbf{P})\|$ can exceed the velocity constraints of the UAV, the magnitude of (16) must be saturated at the limits V_{min}, V_{max}. This field smoothly and safely guides the UAV towards the target, as proved in [16], [17], [18], [19].

Hence, the outputs of the forward simulation performed following the velocity field of (16) are: *(i)* the state achieved at ΔT_{sim}, $X_4 = (\mathbf{P}_4, \psi_4, \gamma_4)$; and, *(ii)* the modulus of the interfered velocity at $2\Delta T_{upd}$, $V_{2\Delta T_{upd}} = \|\bar{\mathbf{u}}(\mathbf{P})\|_{2\Delta T_{upd}}$.

B. Quartic Bézier curves optimization

Bézier curves are parametric curves, whose shape is modified through the selection of control points [3], [4]. The general definition of a n-degree Bézier curve $\mathbf{C}(\tau) \in \mathbb{R}^3$ is

$$\mathbf{C}(\tau) = \sum_{i=0}^{n} B_i^n(\tau)\mathbf{P_i}, \quad \tau \in [0,1], \tag{17}$$

where $B_i^n(\tau) = \binom{n}{i}\tau^i(1-\tau)^{n-i}$ denotes the i^{th} Bernstein polynomial and $\mathbf{P_i} \in \mathbb{R}^3$ the i^{th} control point. These curves are characterized by some features which are extremely useful for trajectory planning purposes: *(i)* the curve passes through its endpoints, hence $\mathbf{C}(\tau=0) = \mathbf{P_0}$, $\mathbf{C}(\tau=1) = \mathbf{P_n}$; *(ii)* the curve is tangent to the lines connecting $\mathbf{P_0} \to \mathbf{P_1}$ and $\mathbf{P_{n-1}} \to \mathbf{P_n}$ in its endpoints $\mathbf{P_0}$ and $\mathbf{P_n}$, respectively; *(iii)* the curve is invariant under affine transformations. For an exhaustive analysis of all the properties characterizing Bézier curves, refer to [4].

In this work, quartic Bézier curves are proposed to compute an optimal, safe and flyable path connecting an initial state $X_0 = (x_0, y_0, z_0, \psi_0, \gamma_0, \mathcal{K}_{H_0}, \mathcal{K}_{V_0})$ to a final state $X_4 = (x_4, y_4, z_4, \psi_4, \gamma_4)$ obtained as in Sec. III-A. Initial path curvatures $\mathcal{K}_{H_0}$ and $\mathcal{K}_{V_0}$ have been added to the initial state for curvature continuity purposes. The Bézier curve optimization problem is inspired by [13] and expanded to 3D applications. The curve is firstly built in a 2D (horizontal) form and, then, extended to the third dimension, as shown in Fig. 3. The heading angle $\psi(\tau)$ and horizontal curvature $\mathcal{K}_H(\tau)$ of the 2D Bezier curve (plane $\mathcal{O}_{x-y}$) are computed as

$$\psi(\tau) = \arctan\left(\frac{y'(\tau)}{x'(\tau)}\right), \tag{18}$$

$$\mathcal{K}_H(\tau) = \frac{\psi'(\tau)}{V_{\tau_H}(\tau)} = \frac{x'(\tau)y''(\tau) - x''(\tau)y'(\tau)}{(x'^2(\tau) + y'^2(\tau))^{3/2}}, \tag{19}$$

where $x'(\tau), y'(\tau), x''(\tau), y''(\tau)$ are the first and second-order τ-derivatives of $\mathbf{C}(\tau)$ in the x, y directions, $V_{\tau_H}(\tau) =$

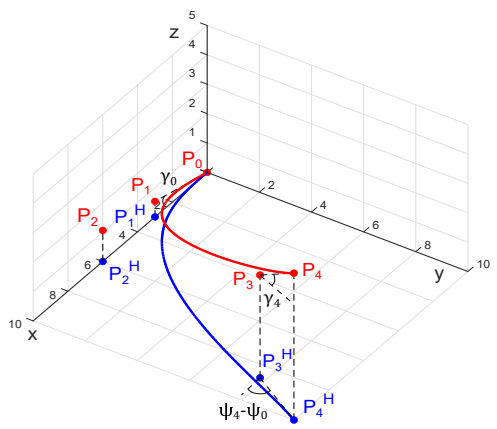

Fig. 3. Example of 3D Bézier curve with $\mathbf{P_0} = \mathbf{0}$, $\psi_0 = 0$ (red) and its projection in the horizontal plane (blue).

$\sqrt{x'^2(\tau) + y'^2(\tau)}$ is the bi-dimensional "parametric speed", $\psi'(\tau)$ is the τ-derivative of $\psi(\tau)$. Then, we consider the third dimension analogously exploiting the longitudinal "plane" $\mathcal{O}_{xy-z}$, which contains both the red and blue curves of Fig. 3. Hence, considering curvilinear abscissa $xy(\tau) = \int_0^\tau V_{\tau_H}(\tau)d\tau$ and ordinate z, analogous definitions of flight path angle $\gamma(\tau)$ and vertical curvature $\mathcal{K}_V(\tau)$ can be derived

$$\gamma(\tau) = \arctan\left(\frac{z'(\tau)}{V_{\tau_H}(\tau)}\right), \tag{20}$$

$$\mathcal{K}_V(\tau) = \frac{\gamma'(\tau)}{V_\tau(\tau)} = \frac{V_{\tau_H}(\tau)z''(\tau) - V'_{\tau_H}(\tau)z'(\tau)}{\left(x'^2(\tau) + y'^2(\tau) + z'^2(\tau)\right)^{3/2}}, \tag{21}$$

where $z'(\tau), z''(\tau)$ are the first and second-order τ-derivatives of $\mathbf{C}(\tau)$ in the z direction; $V'_{\tau_H}(\tau)$ and $\gamma'(\tau)$ are the τ-derivatives of $V_{\tau_H}(\tau)$ and $\gamma(\tau)$. $V_\tau(\tau) = \sqrt{x'^2(\tau) + y'^2(\tau) + z'^2(\tau)}$ is the 3D "parametric speed". For 3D applications, $V_\tau(\tau)$ should also be used for the computation of $\mathcal{K}_H(\tau)$, but the conservative formulation of (19) is still valid since it always overestimates the horizontal curvature ($V_\tau(\tau) \geq V_{\tau_H}(\tau)$).

In order to simplify mathematical derivation, the state $X = (x, y, z, \psi, \gamma, \mathcal{K}_H, \mathcal{K}_V)$ is transformed into $\bar{X} = (\bar{x}, \bar{y}, \bar{z}, \psi - \psi_0, \gamma, \mathcal{K}_H, \mathcal{K}_V)$ through the following coordinate affine transformation

$$\begin{bmatrix}\bar{x}\\ \bar{y}\\ \bar{z}\end{bmatrix} = \overbrace{\begin{bmatrix}\cos(\psi_0) & \sin(\psi_0) & 0\\ -\sin(\psi_0) & \cos(\psi_0) & 0\\ 0 & 0 & 1\end{bmatrix}}^{R_\Psi}\begin{bmatrix}x - x_0\\ y - y_0\\ z - z_0\end{bmatrix}. \tag{22}$$

Hence, the initial state X_0 is transformed into $\bar{X}_0 = (0, 0, 0, 0, \gamma_0, \mathcal{K}_{H_0}, \mathcal{K}_{V_0})$. This transformation allows to recreate the frame reported in Fig. 3, so that motion in horizontal plane, $\mathcal{O}_{x-y}$, and vertical plane, $\mathcal{O}_{xy-z}$, can be easily identified. The first control point of the curve is $\bar{\mathbf{P}}_\mathbf{0} = \mathbf{0}$ to secure $\mathcal{C}^0$ continuity with the previous path. Then, as in [13], second and third control point locations in $\mathcal{O}_{x-y}$ are computed in order to achieve heading and horizontal curvature continuity. Hence

$$\begin{aligned}\bar{\mathbf{P}}^\mathbf{H}_\mathbf{1} &= \left[s_{H_0}, 0\right]^T,\\ \bar{\mathbf{P}}^\mathbf{H}_\mathbf{2} &= \left[\bar{x}_2, \tfrac{4}{3}\mathcal{K}_{H_0}s^2_{H_0}\right]^T,\end{aligned} \tag{23}$$

with $s_{H_0} = \|\bar{\mathbf{P}}^\mathbf{H}_\mathbf{1} - \bar{\mathbf{P}}_\mathbf{0}\| = \|\bar{\mathbf{P}}_\mathbf{1} - \bar{\mathbf{P}}_\mathbf{0}\|\cos(\gamma_0) = s_0\cos(\gamma_0)$. The vertical coordinates of these points must be computed imposing $\gamma(\tau = 0) = \gamma_0$ and $\mathcal{K}_V(\tau = 0) = \mathcal{K}_{V_0}$. Hence

$$\begin{aligned}\bar{\mathbf{P}}_\mathbf{1} &= \left[\bar{\mathbf{P}}^\mathbf{H}_\mathbf{1}{}^T, s_{H_0}\tan(\gamma_0)\right]^T,\\ \bar{\mathbf{P}}_\mathbf{2} &= \left[\bar{\mathbf{P}}^\mathbf{H}_\mathbf{2}{}^T, \tfrac{4}{3}\mathcal{K}_{V_0}\tfrac{s^2_{H_0}}{(\cos(\gamma_0))^3} + \bar{x}_2\tan(\gamma_0)\right]^T.\end{aligned} \tag{24}$$

Finally, the last two control points are computed in order to achieve the desired state $\bar{X}_4 = (\bar{x}_4, \bar{y}_4, \bar{z}_4, \psi_4 - \psi_0, \gamma_4)$

$$\begin{aligned}\bar{\mathbf{P}}_\mathbf{4} &= R_\Psi\left[x_4 - x_0, y_4 - y_0, z_4 - z_0\right]^T,\\ \bar{\mathbf{P}}_\mathbf{3} &= \bar{\mathbf{P}}_\mathbf{4} - s_4\begin{bmatrix}\cos(\gamma_4)\cos(\psi_4 - \psi_0)\\ \cos(\gamma_4)\sin(\psi_4 - \psi_0)\\ \sin(\gamma_4)\end{bmatrix},\end{aligned} \tag{25}$$

with $s_4 = \|\bar{\mathbf{P}}_\mathbf{4} - \bar{\mathbf{P}}_\mathbf{3}\|$. In such way, despite its tridimensionality, only three independent parameters $(s_0, \bar{x}_2, s_4)$ are required to determine a smooth curve, i.e. characterized by continuous curvature w.r.t. the previous path. Furthermore, it becomes clear that quartic Bézier curves represent the lowest-order Bézier curves to achieve curvature continuity and acquire the desired state $\bar{X}_4$.

The selection of $(s_0, \bar{x}_2, s_4)$ is performed solving an optimization problem aiming at minimizing the overall curvature $\mathcal{K}^2(\tau) = \mathcal{K}^2_H(\tau) + \mathcal{K}^2_V(\tau)$ and length of Bezier curve, while maximizing the distance from the closest moving obstacle under kinematic and safety constraints. Note that the number of optimization variables does not depend on the number of obstacles. The problem is solved using Sequential Quadratic Programming (SQP) and is defined as

$$\underset{s_0, \bar{x}_2, s_4}{\text{minimize}} \quad c_1\int_0^1 \mathcal{K}^2(\tau)d\tau + c_2\int_0^1 V_\tau(\tau)d\tau + \tag{26a}$$

$$+ c_3\max_{i\in\mathbb{K}_{dyn}}\left(\int_0^1 \frac{1}{\Gamma_i(\mathbf{C}(\tau))}d\tau\right) \tag{26b}$$

$$\text{subject to} \quad |\mathcal{K}_H(\tau)|_{max} \leq \frac{1}{R_{H_{min}}}, \tag{26c}$$

$$\gamma_{min} \leq \gamma(\tau) \leq \gamma_{max}, \tag{26d}$$

$$\Gamma_k(\mathbf{C}(\tau)) \geq 1, \qquad k = 1\cdots K, \tag{26e}$$

$$s_0, \bar{x}_2, s_4 > 0, \tag{26f}$$

where c_1, c_2, c_3 are the weighting factors, $\mathbb{K}_{dyn} \subseteq \{1\cdots K\}$ is the subset of indexes identifying moving obstacles, $R_{H_{min}}$ is conservatively computed using $\max(V_0, V_{2\Delta T_{upd}})$ and (26e) evaluates the "prediction sphere" of moving obstacles to make the optimization problem time-independent. Once $(s_0, \bar{x}_2, s_4)$ are obtained, the control points in the original reference system are computed inverting the coordinate transformation of (22).

C. Speed profile computation

The computed Bézier curve $\mathbf{C}(\tau)$ is still not time dependent. Therefore, speed and acceleration profiles must be calculated to obtain $\mathbf{C}(\tau(l(t)))$, where $l(t) = \int_0^t V(t)dt$ is the arc length of the curve covered at the time t. The speed and acceleration profiles are defined exploiting the output of the IFDS forward simulation. This choice allows to avoid further computation and to obtain a speed profile which does not depend on the polynomial form of the Bézier curve.

The IFDS method, in fact, takes into account the dynamicity of the obstacles and the speed along the streamlines of (16) varies accordingly. To reproduce this speed trend, the velocity reached during the IFDS forward simulation at the time $t = 2\Delta T_{upd}$ is used to compute a linear acceleration profile and a quadratic velocity profile to satisfy $a(t=0) = a_0, V(t=0) = V_0, V(t = 2\Delta T_{upd}) = V_{2\Delta T_{upd}}$. Hence

$$a(t) = a_0 + Bt \tag{27}$$

$$V(t) = V_0 + a_0 t + \frac{B}{2}t^2 \tag{28}$$

$$B = 2\frac{V_{2\Delta T_{upd}} - V_0 - a_0(2\Delta T_{upd})}{(2\Delta T_{upd})^2}, \tag{29}$$

where $t = 0$ is the time at the beginning of each TP iteration and a_0, V_0 are the initial acceleration and speed of the UAV. The time interval $2\Delta T_{upd}$ is selected after the tuning of the proposed strategy to achieve a good compromise between smoothness of the speed profile and accuracy in following the speed initially defined by the IFDS. It must be pointed out that acceleration profile of (27) must be saturated at a_{min}, a_{max}. In this way, the UAV tries to recreate the speed trend computed in the IFDS forward simulation compatibly with its constraints.

IV. Simulation Results

The proposed method is implemented and tested in a simulation environment using Matlab. The developed TP is executed iteratively with $\Delta T_{upd} = 0.5\ s$ to follow a waypoint-based path in a complex scenario, characterized by the presence of both static and dynamic obstacles. The global path is computed offline and does not avoid dynamic obstacles. In this way, the TP is fully exploited to perform dynamic threats avoidance under UAV's constraints.

The test environment is populated with the obstacles listed in Tab. I. DO1 and DO2 are dynamic obstacles moving with constant velocity $\mathbf{v}_k$, initially located in x_k, y_k, z_k. The repulsive coefficient is selected as $\rho_k = 1$ for static obstacles, while dynamic ones are characterized by $\rho_k = 5, \lambda_k = 100$.

The UAV starts from $\mathbf{P} = [16, 16, 0.2]^T$ km, with $\psi_0 = -135$ deg, $\gamma_0 = 0$ deg, $V_0 = 400\ \frac{\text{km}}{\text{h}}, a_0 = 0\ \frac{\text{m}}{\text{s}^2}$. Its motion is limited by the constraints $n_{max} = 6, |\gamma(t)| \leq 60$ deg, $200\ \frac{\text{km}}{\text{h}} \leq V(t) \leq 1000\ \frac{\text{km}}{\text{h}}, |a(t)| \leq 5\ \frac{\text{m}}{\text{s}^2}$. The last waypoint of the global path is located in $\mathbf{P}_{end} = [1, 1, 1]^T$ km and the desired cruise speed along the path is $V_c = 450\ \frac{\text{km}}{\text{h}}$. Finally, the weighting factors are selected as $c_1 = 0.1 \cdot 0.2, c_2 = 1 \cdot 0.4, c_3 = 5 \cdot 0.4$, where the first term of each factor balances the order of magnitude, while the second term sets the relative weights.

A. Trajectory planning with static and moving obstacles

Simulation results are shown in Fig. 4, where red spheres and orange triangles are respectively the positions of dynamic obstacles and UAV, sampled every 25 s.

First of all, the proposed algorithm follows the waypoint-based path (green line and stars) with a smooth motion. The resulting trajectory does not cross exactly each waypoint. However, this behavior is required to follow the global path smoothly, while avoiding obstacles and satisfying constraints.

TABLE I
Obstacles features

ID	x_k, y_k, z_k $[km]$	a_k, b_k, c_k $[km]$	p_k, q_k, r_k	$\mathbf{v}_k$ $[km/h]$
SO1	$10, 12, 0$	$2, 2, 2$	$1, 1, 1$	$[0, 0, 0]^T$
SO2	$12, 9, 0$	$1.5, 2, 1.5$	$2, 2, 2$	$[0, 0, 0]^T$
SO3	$9, 7, 0$	$2, 2, 3$	$1, 1, 0.5$	$[0, 0, 0]^T$
SO4	$3.5, 6, 0$	$2.5, 2.5, 4$	$1, 1, 0.5$	$[0, 0, 0]^T$
SO5	$6, 3, 0$	$2, 2, 5$	$1, 1, 0.5$	$[0, 0, 0]^T$
DO1	$16, 8, 0.8$	$0.5, 0.5, 0.5$	$1, 1, 1$	$[-360, 720, 36]^T$
DO2	$3, 14, 1.5$	$0.5, 0.5, 0.5$	$1, 1, 1$	$[132, -227, 0]^T$

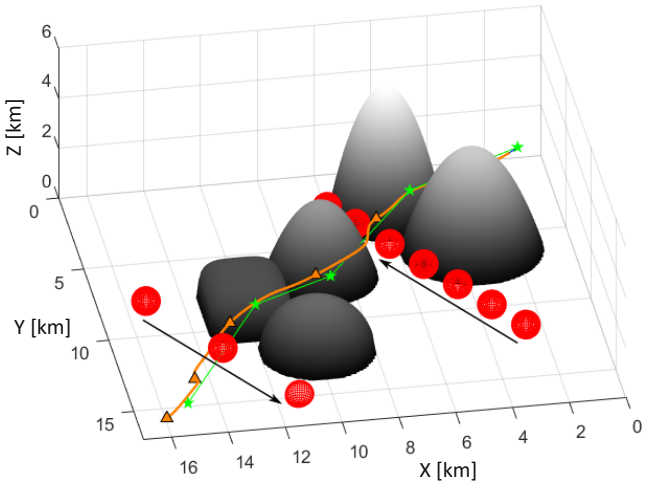

Fig. 4. Simulation results of the proposed trajectory planner.

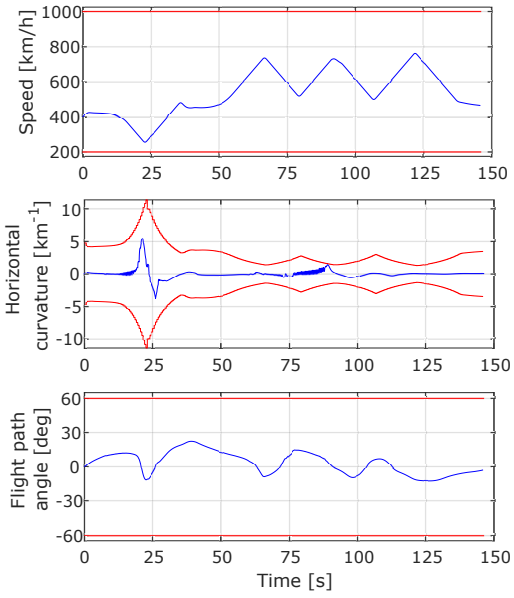

Fig. 5. In blue speed, horizontal curvature and flight path angle over time. In red the constraints of each quantity.

Furthermore, it is possible to notice that the reference global path slightly intersects the static obstacle SO3, which means the local goal point $\mathbf{P}_g$ temporarily penetrates inside the obstacle. Nevertheless, the planned trajectory correctly avoids the object, since obstacle impenetrability feature of IFDS

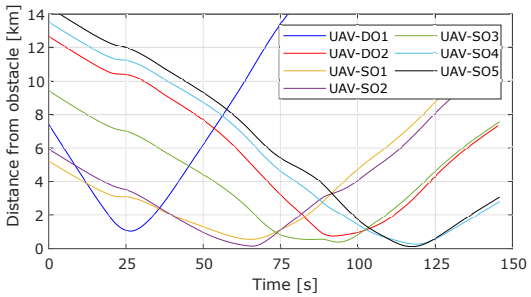

Fig. 6. Distance between the UAV and the obstacles during the simulation.

remains valid even if the goal point is placed inside the volume of an obstacle. As for moving obstacles, the TP leads the UAV to pass behind DO1, which is moving faster, and in front of DO2, which is much slower. This behaviour emerges analyzing Fig. 5, too. Initially, both V and γ are increased to reach V_c and align with the global path. Afterwards, the UAV slows down to make a tight curve and a quick descent to pass behind DO1 (around $T = 20\ s$), and, then, accelerate overcoming the first three static obstacles. At $T = 65\ s$, UAV decelerates again while approaching DO2, but, then, accelerates at $T = 80\ s$ to pass in front of it. In the end, once the UAV has surpassed the two final conic obstacles at $T = 120\ s$, it slows down again to return to V_c.

Finally, the distance between the UAV and the obstacles is reported in Fig. 6. Focusing on the dynamic objects, the minimum distance w.r.t. DO1 is greater than the one w.r.t. DO2 since DO2 is moving in a crowded area of the scenario, hence, the UAV has less space to perform the avoidance.

In this simulation the algorithm has been iterated more than 290 times on a computer with Intel Core i7-8750H (2.20 GHz) and 16 GB RAM. Maximum and mean computation time for a single iteration ($0.255\ s$ and $0.163\ s$) are smaller than trajectory execution time $\Delta T_{upd} = 0.5\ s$, hence the algorithm is suitable for online computation.

V. Conclusions

In this work, we propose a trajectory planning algorithm to face the path following problem for fixed-wing UAVs. The algorithm efficiently combines IFDS and quartic Bézier curves optimization to generate smooth and safe trajectories. First, the IFDS efficiently computes an initial estimate of local trajectory and the speed profile necessary to safely avoid dynamic obstacles. Even if IFDS suffers from local minima, this is not a problem in our application, since it is used to follow a given global path. Then, the proposed 3D Bézier curves optimization method returns an optimal and feasible local trajectory using a fixed number of optimization variables and requiring little computation time.

As result, the proposed method is able to compute a local trajectory, periodically updated to deal with dynamic scenarios. Simulation results corroborate the proposed trajectory planning strategy in a complex scenario in which the UAV safely reaches a target position avoiding obstacles and satisfying kinematic constraints, even considering the correlation between curvature and flight speed.

Future works will include a better management of the velocity profile to reduce excessive speed variations when avoiding dynamic obstacles.

References

[1] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, 2020.

[2] C.-I. Zjup, L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of uav motion planning," *IET Cyber-systems and Robotics*, vol. 2, pp. 14–21, 03 2020.

[3] L. H. Pérez, M. C. M. Aguilar, N. M. Sánchez, and A. F. Montesinos, "Path planning based on parametric curves," in *Advanced Path Planning for Mobile Entities*. Rijeka: IntechOpen, 2018, ch. 7.

[4] R. T. Farouki, "The bernstein polynomial basis: A centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[5] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[6] J.-w. Choi, R. Curry, and G. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering - IAENG*, 2008, pp. 158–166.

[7] J. Arcos-Legarda and M. Calderon-Diaz, "Optimal trajectory planning with dynamic obstacles avoidance for mobile robots navigation," in *AETA 2019 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*. Cham: Springer International Publishing, 2021, pp. 639–648.

[8] A. Askari, M. Mortazavi, H. Talebi, and A. Motamedi, "A new approach in uav path planning using bezier-dubins continuous curvature path," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, 09 2015.

[9] H. Satai, M. A. Zahra, Z. I. Rasool, R. S. Abd-Ali, and C. Pruncu, "Bézier curves-based optimal trajectory design for multirotor uavs with any-angle pathfinding algorithms," *Sensors*, vol. 21, no. 7, 2021.

[10] D. Sartori, D. Zou, and W. Yu, "An efficient approach to near-optimal 3d trajectory design in cluttered environments for multirotor uavs," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1016–1022.

[11] J. Faigl and P. Váňa, "Surveillance planning with bézier curves," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 750–757, 2018.

[12] H. Escamilla Nunez and F. Mora-Camino, "Generation of Optimal Smooth Trajectories using Bezier Curves for Transport Aircraft," in *Brazilian Air Transport Research Society Symposium : "Opportunities and Challenges for the Growth of Air Transport"*., Rio de Janeiro, Brazil, Oct. 2017.

[13] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1882–1891, 2020.

[14] H. Sun, J. Qi, C. Wu, and M. Wang, "Path planning for dense drone formation based on modified artificial potential fields," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 4658–4664.

[15] F. Raheem and M. M.Badr, "Development of modified path planning algorithm using artificial potential field (apf) based on pso for factors optimization," *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 37, pp. 316–328, 11 2017.

[16] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, pp. 433–454, 2012.

[17] H. Wang, W. Lyu, P. Yao, X. Liang, and C. Liu, "Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system," *Chinese Journal of Aeronautics*, vol. 28, no. 1, pp. 229–239, 2015.

[18] P. Yao and S. Zhao, "Three-dimensional path planning for auv based on interfered fluid dynamical system under ocean current (june 2018)," *IEEE Access*, vol. 6, pp. 42 904–42 916, 2018.

[19] P. Yao, H. Wang, and C. Liu, "3-d dynamic path planning for uav based on interfered fluid flow," in *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014, pp. 997–1002.

[20] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. USA: Princeton University Press, 2012.

[21] P. Yao, W. Honglun, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerospace Science and Technology*, vol. 47, 10 2015.