Constrained PSO-splines trajectory generation for an indoor nanodrone

Vincent Marguet, Cong Khanh Dinh, Ionela Prodan, Florin Stoican

Constrained PSO-splines trajectory generation for an indoor nanodrone

Vincent Marguet[1], Cong Khanh Dinh[1], Ionela Prodan[1] and Florin Stoican[2]

Abstract— The trajectory generation problem is still challenging as it involves determining a smooth and feasible path considering convoluted constraints. This paper builds upon the strengths of fast computation of PSO (Particle Swarm Optimization) and the B-splines properties to solve an optimization problem for trajectory planning. The precomputed trajectory of an indoor quadcopter is parameterized by B-splines functions allowing to enforce operation constraints (i.e., position, velocity, angles, thrust, angular velocity, waypoint passing) while minimizing the input effort. Simulation and experimental testing with a nanodrone validate our approach.

Index Terms– Trajectory generation, B-spline curves, Quadcopter, PSO (Particle Swarm Optimization).

I. Introduction

Nowadays, Unmanned Aerial Vehicles (UAV) are used in various applications including search and rescue [1], firefighting [2] and precision agriculture [3]. The common challenge behind these applications is to generate a feasible trajectory that can be tracked by the UAV. While, necessarily, the trajectory tracking is carried on-line, the trajectory generation can be done either on-line or off-line. The first case has the advantage of considering the real position of the UAV if re-computation of the trajectory is needed, however the computational effort may prove too large for real-time implementation (i.e., there are multiple constraints to take into account, many of them, non-linear in the decision variables). The second case gives the opportunity to the user to balance between computational effort and complexity of the generation procedure. Thus, hereinafter, we will focus on the off-line trajectory generation problem.

Many planners provide the trajectory as a list of waypoints. While easy to manipulate, having the trajectory as a collection of interlinked segments raises issues of feasibility, e.g., obstacle collision, sudden changes in attitude which cannot be tracked. To overcome those issues, Bézier [4] or B-spline [5], [6], [7] curves can be used as they have many mathematical and geometrical properties that are useful to design a smooth and feasible trajectory. [6] gives some of the properties, formulates constraints in function of the control points defining the trajectory, preprocess the optimization problem with the tool Yalmip [8] and solves it using the solver MOSEK [9]. Instead of using any solver that has its own requirements on the formulation of the problem (e.g., convexity, linearity and the like), [7] generates a trajectory using Particle Swarm Optimization. Indeed, stochastic methods like Particle Swarm Optimization (PSO) [7], [10], [11], [12] or Differential Evolution (DE) [13] allow to find an optimal particle (here, the control points define the trajectory) while penalizing in the cost function the constraints that are not satisfied. However, [7] does not take into account constraints on waypoint passing, thrust and angular velocities, does not give the formulation of the constraints in terms of control points and tracks the resulted trajectory only in a simulator.

In this paper, we combine these ideas (with improvements in what regards the conservatism of the constraint formulations) and present our PSO-spline approach which generates off-line a feasible trajectory for a UAV. We test our approach over a realistic scenario, both in simulation and experiment (https://tinyurl.com/icuas2024lcis). Moreover, we provide less conservative expressions for the constraint formulation of the angles and angular velocities. Our presented algorithm solving the optimization problem benefits from the B-spline properties to express all the constraints in function of the control points.

A description of B-spline curves with some useful properties are provided with the UAV model and a general formulation of constrained trajectory optimization problem in section II. In section III, we describe our PSO-spline approach and present the algorithm solving this motion planning problem. Section IV describes in detail the simulation and experiment validating our PSO-spline approach, before drawing the conclusions in section V. The proofs to obtain a less conservative expression of the angular constraint and angular velocity constraint are given in the appendices.

Notations: Throughout the paper, the following notations are used: n corresponds to the number of control points describing a B-spline curve; $\mathbf{B}_{d,\boldsymbol{\xi}}(t)$ are the B-spline basis function of degree d and knot vector $\boldsymbol{\xi}$; τ_k refers to the $(k+1)$-th time instant of the knot vector $\boldsymbol{\xi}$; ${}^xP_k$ is the coordinate on x-axis of the $(k+1)$-th control point; $\mathbf{p}_\ell^{wp}$ is the position of the ℓ-th way-point at time t_ℓ; the transpose of a matrix $\mathbf{X}$ is written $\mathbf{X}^\top$.

* This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissements d'avenir, La Région Auvergne-Rhône-Alpes, Pack Ambition Recherche 2021 - PlanMAV, RECPLAMAL-CIR, Ambition Internationale 2023, Horizon-TA C7H-REG24A10 and by a grant from the National Program for Research of the National Association of Technical Universities - GNAC ARUT 2023; Project ID: 207, UNSTPB, Romania.

[1] Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France {vincent.marguet, ionela.prodan}@lcis.grenoble-inp.fr,
† Institute of Engineering and Management Univ. Grenoble Alpes.
[2] Faculty of Automatic Control and Computers, Politehnica University of Bucharest, ACSE, Bucharest, Romania, florin.stoican@upb.ro

II. Prerequisites

We recapitulate here some basics on splines curves which will be instrumental for the contribution proposed in this paper: when using a standard multicopter model we are able to formulate the constrained optimization problem into a manageable form.

A. B-spline curves in trajectory generation

B-splines of degree up to $d \leq m-2$ are defined recursively over the knot sequence $\boldsymbol{\xi} = \{0 = \tau_0 \leq \tau_1 \leq ... \leq \tau_m = T_f\}$:

$$B_{k,1,\boldsymbol{\xi}}(t) = \begin{cases} 1, & t \in [\tau_k, \tau_{k+1}[\\ 0, & \text{otherwise}, \end{cases} \tag{1a}$$

$$B_{k,d,\boldsymbol{\xi}}(t) = \frac{t-\tau_k}{\tau_{k+d}-\tau_k} B_{k,d-1,\boldsymbol{\xi}}(t) + \frac{\tau_{k+d+1}-t}{\tau_{k+d+1}-\tau_{k+1}} B_{k+1,d-1,\boldsymbol{\xi}}(t). \tag{1b}$$

Subsequently, a B-spline curve is described as a linear combination of control points and B-spline basis functions:

$$\mathbf{z}(t) = \sum_{k=0}^{n-1} \mathbf{P}_k B_{k,d,\boldsymbol{\xi}}(t) = \mathbf{P}\mathbf{B}_{d,\boldsymbol{\xi}}(t), \forall t \in [0, T_f], \tag{2}$$

with $\mathbf{P} = \begin{bmatrix}\mathbf{P}_0 & \dots & \mathbf{P}_{n-1}\end{bmatrix}$ being the matrix gathering the control points as its columns and $\mathbf{B}_{d,\boldsymbol{\xi}}(t) = \begin{bmatrix}B_{0,d,\boldsymbol{\xi}}(t) & \dots & B_{n-1,d,\boldsymbol{\xi}}(t)\end{bmatrix}^\top$ the basis vector. B-spline functions and the associated curve have the properties [14]:

P1) B-splines basis functions have a local support:

$$B_{k,d,\boldsymbol{\xi}}(t) = 0, \quad \forall t \notin [\tau_k, \tau_{k+d+1}). \tag{3}$$

P2) B-splines basis functions partition the unity:

$$\sum_{k=0}^{n-1} B_{k,d,\boldsymbol{\xi}}(t) = 1, \quad \forall t \in [\tau_0, \tau_m] \tag{4a}$$

and

$$B_{k,d,\boldsymbol{\xi}}(t) \geq 0, \quad \forall t \in [\tau_0, \tau_m]. \tag{4b}$$

P3) The 'r' order derivatives of B-spline basis functions are linear combinations of B-splines of lower degree, i.e. there exists a matrix $\mathbf{M}_{d,d-r}$ such that:

$$\mathbf{B}^{(r)}_{d,\boldsymbol{\xi}}(t) = \mathbf{M}_{d,d-r}\mathbf{B}_{d-r,\boldsymbol{\xi}}(t) \tag{5}$$

P4) The B-spline curve (2) lies within the union of all convex hulls defined by subsets of $d+1$ consecutive control points.

P5) On each non-empty sub-interval $[\tau_\ell, \tau_{\ell+1})$, the basis functions of degree $d-1$ can be expressed as combination of the basis functions of degree d, thus, the differentiated B-spline curve may be expressed:

$$\dot{\mathbf{z}}(t) = \dot{\mathbf{P}}\mathbf{B}_{d-1,\boldsymbol{\xi}}(t) = \dot{\mathbf{P}}\mathbf{D}^{\ell}_{d-1,d}\mathbf{B}_{d,\boldsymbol{\xi}}(t), t \in [\tau_\ell, \tau_{\ell+1}), \tag{6}$$

with $\mathbf{D}^{\ell}_{d-1,d}$ computed accordingly. Such a matrix always exist since any polynomial of degree $d-1$, $\dot{\mathbf{z}}(t)$ on the sub-interval $[\tau_\ell, \tau_{\ell+1})$, can be described by polynomials of degree d, the B-splines functions $\mathbf{B}_{d,\boldsymbol{\xi}}(t)$, again on the sub-interval $[\tau_\ell, \tau_{\ell+1})$.

Note that, once the parameters of the B-spline functions are fixed, the resulting B-spline curve depends only on its control points. Thus, the trajectory planning problem is reduced to finding the optimal positions of the control points.

B. Multicopter model

We consider the quadcopter model that neglects aerodynamic effects and external disturbances described as in [6]:

$$\ddot{\boldsymbol{\xi}} = T\mathbf{z}_B - g\mathbf{z}_w, \tag{7a}$$

$$\dot{\boldsymbol{\eta}} = N^{-1}(\boldsymbol{\eta})\boldsymbol{\omega}, \tag{7b}$$

$$\mathbf{N}(\boldsymbol{\eta}) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix}, \tag{7c}$$

$$\mathbf{R}^W_B = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\theta s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\theta c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{7d}$$

with $\boldsymbol{\xi} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ the position vector in the inertial frame, $\boldsymbol{\eta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top$ the Z-Y-X Euler angle vector in the inertial frame, $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ the angular velocity vector in the body frame, T the normalized thrust, $\mathbf{z}_w = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ the world frame's z-axis, $\mathbf{N}(\boldsymbol{\eta})$ the transformation matrix for angular velocities from the inertial frame to the body frame, $\mathbf{z}_B$ the last column of $\mathbf{R}^W_B$ (the rotation matrix from the body frame to the inertial frame) and where c stands for cos and s stands for sin. The inputs are represented by the vector $\mathbf{u} = \begin{bmatrix} T & p & q & r \end{bmatrix}^\top$ and the state vector is $\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^\top$. Hereinafter, vector $\mathbf{z}(t)$ denotes the trajectory of the UAV (components x, y, z of the state vector).

C. Optimal trajectories generation

Typically, the constrained optimization problem generating the trajectory for dynamics such as those in (7) is given as:

$$\min_{\mathbf{z}(t)} \int_0^{T_f} \mathcal{C}(\mathbf{z}(t))\mathrm{d}t \tag{8a}$$

$$\text{s.t.} \quad \mathbf{z}(t) \subset \mathbf{K}_p,\ 0 \leq t \leq T_f, \tag{8b}$$

$$v(t) \leq v_{max},\ 0 \leq t \leq T_f, \tag{8c}$$

$$|\phi(t)| \leq \epsilon,\ |\theta(t)| \leq \epsilon,\ 0 \leq t \leq T_f, \tag{8d}$$

$$T_{min} \leq T(t) \leq T_{max},\ 0 \leq t \leq T_f, \tag{8e}$$

$$|p(t)| \leq \omega_{max},\ |q(t)| \leq \omega_{max},\ 0 \leq t \leq T_f, \tag{8f}$$

$$\|\mathbf{z}(t_\ell) - \mathbf{p}^{wp}_\ell\|_2 \leq d_{wp},\ \ell = 1, \dots, N_{WP}, \tag{8g}$$

$$\mathbf{p}(0) = \mathbf{p}_0,\ \mathbf{v}(0) = \mathbf{v}_0,\ \mathbf{a}(0) = \mathbf{a}_0, \tag{8h}$$

$$\mathbf{p}(T_f) = \mathbf{p}_f,\ \mathbf{v}(T_f) = \mathbf{v}_f,\ \mathbf{a}(T_f) = \mathbf{a}_f, \tag{8i}$$

where (8a) defines the cost function minimizing, e.g., trajectory length, energy spent or time to achieve a mission, (8b) and (8c) constrain the position and velocity, (8d) and (8f) constrain the angles and the angular velocities, (8e) limits the thrust, (8g) defines waypoint constraints and (8h) and (8i) define the initial and terminal conditions.

We express all the constraints and the objective function from (8) in terms of the UAV trajectory and its derivatives. Taking the trajectory as a B-spline curve (2) with the degree and the knot vector of the basis functions a priori fixed and using over-approximations based on P4), we reformulate the optimization problem only in terms of the control points. Thus, the initial continuous-time (infinite dimensional) optimization problem (8) is converted to a finite-dimensional, sub-optimal optimization problem.

III. Trajectory generation problem

The B-spline mediated form of (8) is still a difficult-to solve nonlinear optimization problem. In here, we consider a popular heuristic algorithm, the particle swarm optimization (PSO), to provide a fast and accurate solution. To account for the constraints, PSO is penalizing each of them in the cost with slack penalty terms which measure the deviation from feasible candidate trajectories. To express this value, we will use the ramp function which returns zero if the constraint is satisfied or the 1-norm of the constraint violation components if the constraint is not satisfied. The ramp function is defined as $\boldsymbol{\sigma}$: $\mathbb{R}^n \to \mathbb{R}_+$ such that:

$$\boldsymbol{\sigma}(x) = \sum_{i=1}^{n} \max(0, x_i). \tag{9}$$

The variable $\mathbf{X} = \begin{bmatrix}\mathbf{P}_0 & \mathbf{P}_1 & \dots & \mathbf{P}_{n-1}\end{bmatrix}$ gathering all the control points of a given trajectory as in (2) will be used to express all the penalty terms of the objective function:

$$F_{tot}(\mathbf{X}) = \sum_{q=0}^{6} \gamma_q F_q(\mathbf{X}), \tag{10}$$

where $\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5$ and γ_6 are the weights associated to the penalties.

The objective function (10) is representing the constraint optimization problem (8) as each of the penalty term is associated to one of its sub-equations. To express all the penalties in function of the control points, the B-spline properties are exploited. Some proofs to formulate the constraints are given in [6]. However, we provide less conservative expressions for the angular and angular velocities constraints (proofs in appendices I and II). Once the constraints are obtained, they are converted into the following penalties in (10):

$$F_0(\mathbf{X}) = \int_0^{T_f} \left\| \sum_{k=0}^{n-5} (\mathbf{X}\mathbf{M}_{d,d-4})_k B_{k,d-4,\boldsymbol{\xi}}(t) \right\|_2^2 \mathrm{d}t, \tag{11a}$$

$$F_1(\mathbf{X}) = \sum_{k=0}^{n-1} \boldsymbol{\sigma}\left(\mathbf{x}_{\min} - \mathbf{P}_k\right) + \boldsymbol{\sigma}\left(\mathbf{P}_k - \mathbf{x}_{\max}\right), \tag{11b}$$

$$F_2(\mathbf{X}) = \sum_{k=0}^{n-2} \boldsymbol{\sigma}\left(\|\dot{\mathbf{P}}_k\|_2 - v_{\max}\right), \tag{11c}$$

$$F_3(\mathbf{X}) = \sum_{k=0}^{n-3}\sum_{i=0}^{n-3} \boldsymbol{\sigma}\left(G(\epsilon, i, k, \mathbf{P})\right), \tag{11d}$$

$$F_4(\mathbf{X}) = \sum_{k=0}^{n-3} \Big[\boldsymbol{\sigma}\left(\|\ddot{\mathbf{P}}_k + g\mathbf{z}_w\|_2 - T_{\max}\right) + \boldsymbol{\sigma}\left(-{}^{z}\ddot{P}_k - g + T_{\min}\right)\Big], \tag{11e}$$

$$F_5(\mathbf{X}) = \sum_{\ell=1}^{n-d} \sum_{k=\ell+d-2}^{\ell+d} \sum_{i=\ell+d-2}^{\ell+d} \boldsymbol{\sigma}\left(S(\ell, i, k, \mathbf{P})\right), \tag{11f}$$

$$F_6(\mathbf{X}) = \sum_{l=1}^{N_{WP}} \boldsymbol{\sigma}\left(\left\|\mathbf{p}_l^{wp} - \sum_{k=0}^{n-1} \mathbf{P}_k B_{k,d,\boldsymbol{\xi}}(t_\ell)\right\|_2 - d_{wp}\right). \tag{11g}$$

$$G(\epsilon, i, k, \mathbf{P}) = \cot^2\epsilon\, \ddot{\mathbf{P}}_i^\top \ddot{\mathbf{P}}_k - \left(1 + \cot^2\epsilon\right) \ddot{\mathbf{P}}_i^\top \mathbf{z}_w \mathbf{z}_w^\top \ddot{\mathbf{P}}_k - 2g\mathbf{z}_w^\top \ddot{\mathbf{P}}_k - g^2, \tag{12}$$

$$S(\ell, i, k, \mathbf{P}) = (\dddot{\mathbf{P}}\mathbf{D}^{\ell}_{d-3,d-2})_i^\top (\dddot{\mathbf{P}}\mathbf{D}^{\ell}_{d-3,d-2})_k - \omega_{\max}^2 \left(\ddot{\mathbf{P}}_i + \mathbf{z}_w g\right)^\top \left(\ddot{\mathbf{P}}_k + \mathbf{z}_w g\right). \tag{13}$$

We explain now how the sub-equations from (8) are represented by a penalty in (11):

i) Minimizing the input effort: (8a) is the objective function that aims to minimize the input effort which corresponds to the trajectory's snap: $\min_{\mathbf{z}(t)} \int_0^{T_f} \left\|\mathbf{z}^{(4)}(t)\right\|_2^2 \mathrm{d}t$. Applying the derivative property of the B-spline functions P3), the corresponding penalty is given in (11a).

Remark 1: (11a) may be brought into a quadratic form where only control points appear, through standard manipulations [14]. Moreover, the expression can be simplified if the degree is 4 as the snap function becomes a step function of amplitude $(\mathbf{X}\mathbf{M}_{d,d-4})_k$ and constant time step $T_f/(n-d)$. Thus, the snap of the trajectory becomes $F_0(\mathbf{X}) = \frac{T_f}{n-d}\sum_{k=0}^{n-1}\|(\mathbf{X}\mathbf{M}_{d,d-4})_k\|_2^2$. ♦

ii) Space constraints: (8b) describes the indoor environment limitations. The allowed volume can be represented by a box with lower and upper limits on each axis denoted as $\mathbf{K}_p$. The convexity property of the B-spline curves guarantees that if the control points are inside this box then the trajectory also stays inside this box. Thus, the position penalty is expressed in (11b).

iii) Velocity constraints: (8c) is imposing the velocity of the drone to always stay lower than v_{max}. We remind that the respective derivative control points are obtained by multiplying the initial control points by the matrix appearing in property P3): $\dot{\mathbf{P}}_k = (\mathbf{X}\mathbf{M}_{d,d-1})_k$, $\ddot{\mathbf{P}}_k = (\mathbf{X}\mathbf{M}_{d,d-2})_k$, $\dddot{\mathbf{P}}_k = (\mathbf{X}\mathbf{M}_{d,d-3})_k$. Using the derivative control points and the convexity property, we obtain the velocity penalty in (11c).

iv) Roll and pitch constraints: (8d) imposes that the modulus of the roll and pitch angles stay inferior to ϵ. The constraints expressed in terms of control points are obtained in Appendix I and the sufficient condition (27) is converted to the penalty term (11d).

v) Thrust constraints: (8e) imposes the thrust to stay in the interval $[T_{\min}, T_{\max}]$ during the whole trajectory. The expression in function of the control points is obtained in [6] and its associated penalty term is given in (11e).

vi) Angular velocity constraints: (8f) forces the angular velocities' magnitudes to stay inferior to $\omega_{\max}$. The constraints expressed in terms of control points are obtained in Appendix II and the sufficient condition (37) is converted to the penalty term (11f).

vii) Waypoints conditions: (8g) imposes that the position at time t_ℓ is inside the sphere centered in the waypoint $\mathbf{p}_\ell^{wp}$ and of radius d_{wp} for each ℓ. The corresponding penalty term is given in (11g).

viii) Initial and final conditions: (8h) and (8i) are specifying the desired initial and final components for the position ($\mathbf{p}_0$ and $\mathbf{p}_f$), velocity ($\mathbf{v}_0$ and $\mathbf{v}_f$) and acceleration ($\mathbf{a}_0$ and $\mathbf{a}_f$). The initial and final position are the first and last control points (we assume a clamped B-spline curve):

$$\mathbf{P}_0 = \mathbf{p}_0, \quad \mathbf{P}_{n-1} = \mathbf{p}_f. \tag{14}$$

For the initial velocity, $\dot{\mathbf{z}}(t=0) = \dot{\mathbf{P}}_0 = \mathbf{v}_0$, reminding the relation between $\dot{\mathbf{P}}_0$ and $\mathbf{P}$, we obtain:

$$\frac{d}{\tau_{d+1}}(\mathbf{P}_1 - \mathbf{P}_0) = \mathbf{v}_0, \tag{15}$$

which imposes the second control point to be:

$$\mathbf{P}_1 = \mathbf{P}_0 + \frac{\tau_{d+1}}{d}\mathbf{v}_0 = \mathbf{p}_0 + \frac{\tau_{d+1}}{d}\mathbf{v}_0. \tag{16}$$

A similar calculus starting from $\dot{\mathbf{z}}(t=T_f) = \dot{\mathbf{P}}_{n-2} = \mathbf{v}_f$ leads to

$$\mathbf{P}_{n-2} = \mathbf{P}_{n-1} - \frac{T_f - \tau_{n-1}}{d}\mathbf{v}_f = \mathbf{p}_f - \frac{T_f - \tau_{n-1}}{d}\mathbf{v}_f. \tag{17}$$

The same principle applied twice for the initial $\ddot{\mathbf{z}}(t=0) = \ddot{\mathbf{P}}_0 = \mathbf{a}_0$ and final $\ddot{\mathbf{z}}(t=T_f) = \ddot{\mathbf{P}}_{n-3} = \mathbf{a}_f$ acceleration is applied and results in:

$$\mathbf{P}_2 = \mathbf{p}_0 + \frac{\tau_{d+1}+\tau_{d+2}}{d}\mathbf{v}_0 + \frac{\tau_{d+2}\tau_{d+1}}{d(d-1)}\mathbf{a}_0, \tag{18a}$$

$$\mathbf{P}_{n-3} = \mathbf{p}_f - \frac{2T_f - \tau_{n-2} - \tau_{n-1}}{d}\mathbf{v}_f + \frac{(T_f - \tau_{n-2})(T_f - \tau_{n-1})}{d(d-1)}\mathbf{a}_f. \tag{18b}$$

As it is meaningless to compute trajectories that are not satisfying the initial and final conditions, we initialize all the particles with the 3 first control points and 3 last control points respecting (14), (16), (17) and (18). These control points are fixed at all the iterations for all the particles and the algorithm is optimizing only the remaining $n-6$ control points of $\mathbf{X}$.

Our PSO-spline approach is detailed in Algorithm 1. Parameters r_1 and r_2 refer to random values between 0 and 1. $G_b = F_{tot}(\mathbf{X}_b)$ is the cost associated to the current best particle $\mathbf{X}_b$. The code implementing this algorithm was run in Matlab 2023a using its codegen option to make a C-MEX variant of the Gilbert-Johnson-Keerthi function [15] testing polyhedron intersections and CasADi [16] to generate the B-splines basis functions.

IV. Simulation and experiment

The aim of this section is to describe the scenario, provide the numerical values used in the simulation and experiment, and analyse the results.

A. Simulation

The scenario is taken as in [6]: the planned trajectory needs to pass in the neighborhood (defined as spheres of radii d_{wp}) of eight waypoints at the specified times detailed in Table I. A maximum distance between the center of the nanodrone and the waypoints of 0.3m is allowed in experiment. Taking into account the bound on the tracking error ($\delta = 0.1$m) and the half size of our nanodrone (0.05m), the proposed value in [6] of $d_{wp} = 0.05$m is relevant. During all its trajectory (from time 0s to time $T_f = 30$s), the UAV must respect all the physical constraints detailed in the previous section. The numerical values of the associated bounds are taken the same as [6] and are listed in the first line of Table II. Note that the values used in the trajectory planning are tightened to account for the contribution of the tracking error which may lead to a violation of the physical bounds in the experiment otherwise. The physical bounds for the experiment are the ones usually considered in the state of the art.

The gravitational acceleration g is 9.81 m/s^2. The position of the UAV must stay inside the box defined by $\mathbf{x}_{\min} = \begin{bmatrix}-1.5 & -1 & 0\end{bmatrix}^\top$ and $\mathbf{x}_{\max} = \begin{bmatrix}1.5 & 1 & 1.5\end{bmatrix}^\top$.

The number of particles n_{part}, iterations n_{iter} and weights γ_i for the objective function are detailed in Table III and IV. The specific parameters of our PSO-spline approach are the damping weight which is set to 1 and the learning coefficients that are set to $c_1 = 1.2$ and

Algorithm 1: Solving a motion planning problem with our PSO-spline approach.

Input: list of parameters in (III) $n, c_1, c_2, n_{part}, n_{iter}$;
Output: $\mathbf{X}_b$ minimizing the fitness function F_{tot};

Create n_{part} random matrices of n control points respecting the initial and final conditions: $\mathbf{X}_1, ..., \mathbf{X}_{n_{part}}$
Create n_{part} random matrices of the same size corresponding to the associated velocity: $\mathbf{V}_1, ..., \mathbf{V}_{n_{part}}$
for k=1:n_{part} do
  $\mathbf{X}_k^* \leftarrow \mathbf{X}_k$;
  Compute $F_{tot}(\mathbf{X}_k^*)$ in (10);
  Store $\mathbf{X}_k^*$ and $F_{tot}(\mathbf{X}_k^*)$;
  if $F_{tot}(\mathbf{X}_k^*) < G_b$ do
    $G_b \leftarrow F_{tot}(\mathbf{X}_k^*)$;
    $\mathbf{X}_b \leftarrow \mathbf{X}_k^*$;
  end
end
for i=1:n_{iter} do
  for k=1:n_{part} do
    $\mathbf{V}_k \leftarrow \mathbf{V}_k + c_1 r_1 (\mathbf{X}_k^* - \mathbf{X}_k) + c_2 r_2 (\mathbf{X}_b - \mathbf{X}_k)$;
    $\mathbf{X}_k \leftarrow \mathbf{X}_k + \mathbf{V}_k$;
    Compute $F_{tot}(\mathbf{X}_k)$ in (10);
    if $F_{tot}(\mathbf{X}_k) < F_{tot}(\mathbf{X}_k^*)$ do
      $\mathbf{X}_k^* \leftarrow \mathbf{X}_k$;
      Upload $F_{tot}(\mathbf{X}_k^*)$;
      if $F_{tot}(\mathbf{X}_k^*) < G_b$ do
        $G_b \leftarrow F_{tot}(\mathbf{X}_k^*)$;
        $\mathbf{X}_b \leftarrow \mathbf{X}_k^*$;
      end
    end
  end
end
Return $\mathbf{X}_b$

TABLE I
Numerical values of the waypoints.

ℓ	1	2	3	4	5	6	7	8
$\mathbf{p}_\ell^{wp}$	-0.75	0.65	0.40	-0.15	0.65	-0.50	-0.60	0.25
	0.60	0.50	-0.40	0.25	-0.65	0.50	-0.60	0.25
	0.50	0.25	.40	0.25	0.25	0.75	0.50	0.25
t_ℓ[s]	7.8	15.3	24	4.5	12.6	18	21	27

TABLE II
Numerical values of the bounds.

Trajectory	$v_{\max}$	$T_{\min}$	$T_{\max}$	ϵ	$\omega_{\max}$
Planning	$0.5m/s$	$9.7m/s^2$	$9.9m/s^2$	$1.75°$	$1.5°/s$
Tracking	$1m/s$	$0m/s^2$	$10.5m/s^2$	$7°$	$30°/s$

$c_2 = 1.5$. To solve our optimization problem, we only changed two values compared to [6]: the degree of the B-spline function (4 instead of 5) and the number of control points (20 instead of 41). Indeed, those modifications are also providing a feasible trajectory while leading to a faster convergence time because it extends the influence of each control point.

TABLE III
Numerical values of the weights for the cost functions.

γ_0	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6
1	1	4×10^4	40	8×10^4	5×10^3	5×10^4

As there are random components in our algorithm, the trajectories may slightly differ between successive runs. As a consequence, we run our algorithm 100 times and average the outputs of interest (e.g., computational time, trajectory cost, maximal distance between waypoints and the position at the desired time) in Table IV. The results corresponding to one of the 100 simulations are shown in Fig. 1 for the resulted trajectory and Fig. 2 for the associated constraints. The average over 100 simulations of the maximal distance between waypoints and the position at the desired time is 0.28m with a standard deviation of 0.07m.

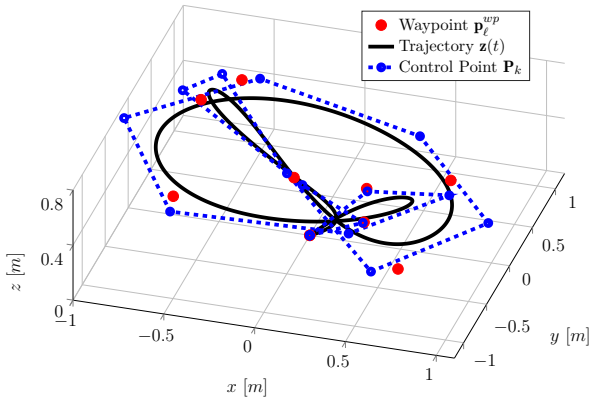

Fig. 1. Trajectories generated with the parameters in Table I. We show the planned trajectory (continuous black) along with the waypoints (red spheres) and the control points (dashed connected, blue circles).

This average value may be decreased by increasing the number of particles and iterations but the computation time would increase as well, or by increasing the waypoint criteria's weight γ_6 associated to (11g), but it would give less relative importance to the other constraints. Although the degree and the number of control points are fewer in our implementation, notice that the velocity, thrust, angle and angular velocity profiles are very similar to the ones obtained by [6] (see Fig. 4, Fig. 6). Moreover, our B-spline approach is giving the opportunity to formulate additional constraints of any type (in particular non convex constraints for obstacle avoidance).

B. Experiment

The trajectory obtained with our PSO-spline approach is followed by a nanodrone in our Esisarium platform. To keep the comparison with [6], we used the same

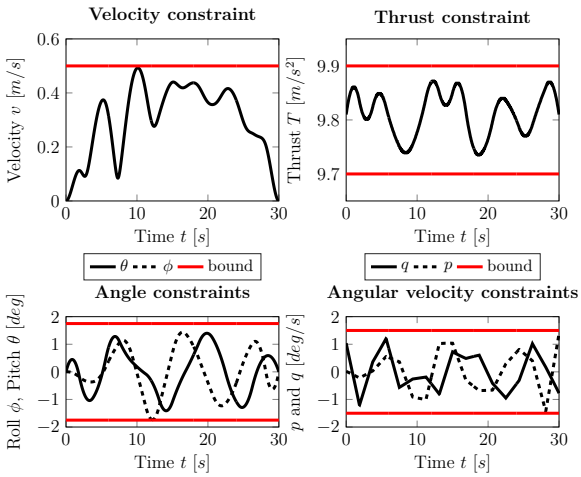

Fig. 2. Constraints satisfaction for the planned trajectory.

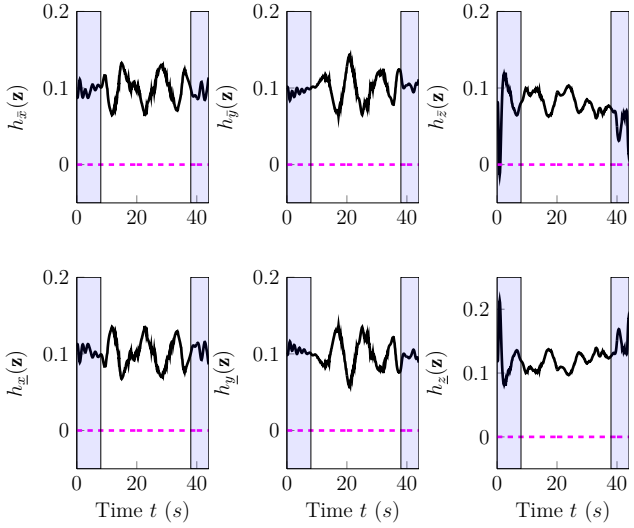

Fig. 3. Experiment: Tracking performance on each axis.

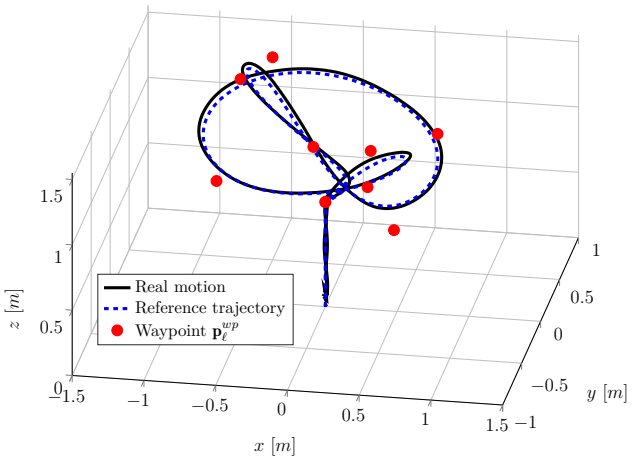

Fig. 4. Experiment: Trajectory tracking.

CBF-QP (Control Barrier Functions based Quadratic Program) with $\delta = 0.1$, $a_1 = 6$ and $a_2 = 8$, associated to a LQR (Linear Quadratic Regulator) with $Q = \text{diag}(500I_3, 250I_3)$ and $R = 10I_3$) to track the trajectory. The following quadratic program (QP) is solved online every 0.1s in order to restrict the tracking error on each axis $q \in \{x, y, z\}$ to $[-\delta, +\delta]$:

$$\min_{\boldsymbol{\mu}} \quad \|\Psi(\pi(\mathbf{z})) - \boldsymbol{\mu}\|_2^2 \tag{19}$$

$$\text{s.t.} \left|\mu_q - \ddot{q}^{ref} - a_1(\dot{q}^{ref} - \dot{q}) - a_2(q^{ref} - q)\right| \leq a_2\delta \tag{20}$$

A video of the experiment is available at https://tinyurl.com/icuas2024lcis. To avoid the observed ground effect, we added an offset of 0.8m in the z-axis. To do so, the trajectory that we obtained with our PSO-spline approach (lasting 30s) is starting after a period of take off (5s) and hovering (3s) and ending before a period of hovering (3s) and landing (3s). The results during the whole experiment are presented in the Fig. 3 to 7. We marked in blue areas the added parts that are not the focus of this paper and in red dashed lines the allowed bounds for the experiments, listed in Table II. As all the curves in Fig. 3 are positive between 8 and 38s, the actual trajectory of our nanodrone is staying inside the corridor of lower and upper margin $\delta = 0.1$m centered in the planned trajectory. We clearly see that the constraints imposed on the position, thrust, roll angle, pitch angle, velocity and angular velocities are satisfied in the Fig. 5, 6 and 7. The average of the distance between the actual trajectory and the waypoints is 0.07m and the maximum value is 0.17m. This result is satisfying the waypoint constraint as all the distances are inferior to the limit imposed of 0.3m. To conclude on the experiment, the nanodrone has followed its trajectory while satisfying all the constraints of the scenario.

V. Conclusion

This paper presented a PSO-spline approach to solve a constrained optimisation problem for a quadcopter trajectory generation. Using the benefits of B-spline properties, various constraints such as position, velocity, angles, angular velocities, thrust and waypoint passing are expressed and converted into penalties added to the cost function corresponding to the input effort. The capability of the provided algorithm compared to traditional solvers lies in the fact that there is no restriction to formulate the constraints and that the user is able to tune the parameters to give more importance to a given criteria by refining the weights associated to each penalty. Furthermore, the user is able to tilt the balance between fast computational time and optimality

TABLE IV
Simulations parameters and computational analysis associated (mean over 100 simulations).

n_{iter}	n_{part}	n	Penalty	Time $[s]$
200	500	20	3.81×10^4	28.0

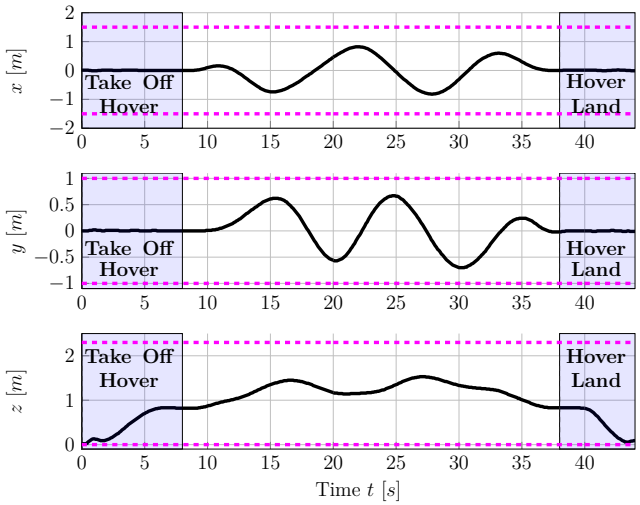

Fig. 5. Experiment: Position per axis.

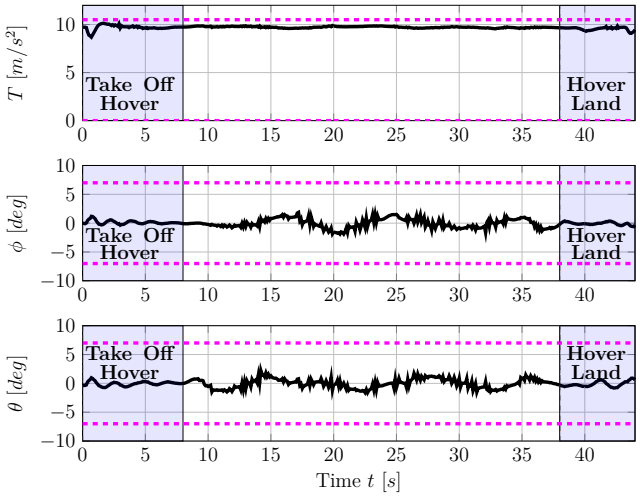

Fig. 6. Experiment: Thrust, Roll angle and Pitch angle.

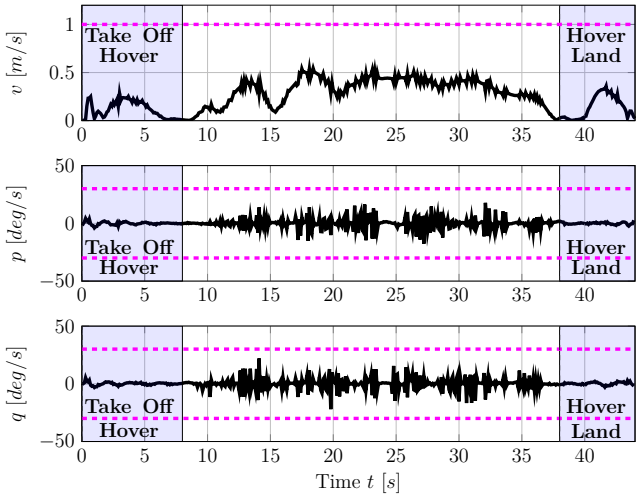

Fig. 7. Experiment: Velocity and angular velocities.

of the solution by playing with the number of particles and iterations. The paper also provided experimental results with a video to confirm that the scenario has been satisfied with a nanodrone tracking the planned trajectory in our indoor platform.

Future work concentrates on adding obstacle avoidance constraints, as well as generating trajectories for a group of quadcopters taking into account communication constraints while ensuring a certain formation configuration.

Appendix I
A less conservative expression for the angular constraint

These constraints

$$\|\mathbf{A}_\epsilon \ddot{\mathbf{r}}(t)\|_2 \leq \mathbf{z}_w^\top \ddot{\mathbf{r}}(t) + g \tag{21}$$

are expressed in terms of control points in [6] as

$$\left\|\mathbf{A}_\epsilon \ddot{\mathbf{P}}_k\right\|_2 \leq \mathbf{z}_w^\top \ddot{\mathbf{P}}_k + g \quad \forall k = 0, \ldots, n-3, \tag{22}$$

where[1] $\mathbf{A}_\epsilon = \text{diag}(\cot\epsilon, \cot\epsilon, 0)$ and g is the gravitational acceleration. A less conservative expression is obtained if we replace in (21) the term $\ddot{\mathbf{r}}(t)$ with its corresponding B-spline curve representation

$$\left\|\mathbf{A}_\epsilon \sum_{k=0}^{n-3} \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t)\right\|_2 \leq \mathbf{z}_w^\top \sum_{k=0}^{n-3} \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t) + g, \tag{23}$$

which, after squaring and regrouping the lhs and rhs terms, leads to

$$\begin{aligned}
&\cot^2\epsilon \sum_{k=0}^{n-3}\sum_{i=0}^{n-3} \ddot{\mathbf{P}}_i^\top \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t) B_{i,d-2,\boldsymbol{\xi}}(t) \\
&\leq (1+\cot^2\epsilon) \sum_{k=0}^{n-3}\sum_{i=0}^{n-3} \ddot{\mathbf{P}}_i^\top \mathbf{z}_w \mathbf{z}_w^\top \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t) B_{i,d-2,\boldsymbol{\xi}}(t) \\
&\quad + 2g\mathbf{z}_w^\top \sum_{k=0}^{n-3} \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t) + g^2.
\end{aligned} \tag{24}$$

Next, using the unit partitioning property we may group the rhs of (24) to arrive at

$$\begin{aligned}
&\cot^2\epsilon \sum_{k=0}^{n-3}\sum_{i=0}^{n-3} \ddot{\mathbf{P}}_i^\top \ddot{\mathbf{P}}_k B_{k,d-2,\boldsymbol{\xi}}(t) B_{i,d-2,\boldsymbol{\xi}}(t) \\
&\quad \leq \sum_{k=0}^{n-3}\sum_{i=0}^{n-3} \Big[(1+\cot^2\epsilon) \ddot{\mathbf{P}}_i^\top \mathbf{z}_w \mathbf{z}_w^\top \ddot{\mathbf{P}}_k \\
&\qquad + 2g\mathbf{z}_w^\top \ddot{\mathbf{P}}_k + g^2\Big] B_{k,d-2,\boldsymbol{\xi}}(t) B_{i,d-2,\boldsymbol{\xi}}(t)
\end{aligned} \tag{25}$$

which, with notation (12), may be written compactly as

$$\sum_{k=0}^{n-3}\sum_{i=0}^{n-3} G(\epsilon, i, k, \mathbf{P}) B_{k,d-2,\boldsymbol{\xi}}(t) B_{i,d-2,\boldsymbol{\xi}}(t) \leq 0. \tag{26}$$

Since the B-spline basis functions are positive (as per Property P2)), a sufficient condition for (21) to hold is that

$$G(\epsilon, i, k, \mathbf{P}) \leq 0, \quad \forall i, k = 0, \ldots, n-3. \tag{27}$$

[1] cot is the notation for the cotangent

Appendix II
A less conservative expression for the angular velocity constraint

The angular velocity constraints are expressed in terms of control points in [6] as

$$\mathbf{z}_w^\top \ddot{\mathbf{P}}_k \geq \zeta_{\ell-d+1} - g, \quad \forall k = \ell - d + 2, \ldots, \ell, \quad (28a)$$

$$\left\| \dddot{\mathbf{P}}_k \right\|_2 \leq \zeta_{\ell-d+1}\omega_{\max}, \quad \forall k = \ell - d + 3, \ldots, \ell. \quad (28b)$$

with $\zeta_1, \ldots, \zeta_{n-d+1}$ positive constants. A less conservative expression is obtained with the following steps:

Recall the initial inequality from [6]

$$\|\mathbf{h}_\omega\|_2 \leq \frac{\|\dddot{\mathbf{r}}(t)\|_2}{T(t)} = \frac{\|\dddot{\mathbf{r}}(t)\|_2}{\|\ddot{\mathbf{r}}(t) + \mathbf{z}_w g\|_2} \leq \omega_{\max}, \quad (29)$$

where, since all terms are positive, we may put it into form

$$\|\dddot{\mathbf{r}}(t)\|_2^2 \leq \omega_{\max}^2 \|\ddot{\mathbf{r}}(t) + \mathbf{z}_w g\|_2^2. \quad (30)$$

Replacing with the B-spline curves associated with $\ddot{\mathbf{r}}(t)$, $\dddot{\mathbf{r}}(t)$ we arrive at

$$\left\| \dddot{\mathbf{P}} \mathbf{B}_{d-3,\boldsymbol{\xi}}(t) \right\|_2^2 \leq \omega_{\max}^2 \left\| \ddot{\mathbf{P}} \mathbf{B}_{d-2,\boldsymbol{\xi}}(t) + \mathbf{z}_w g \right\|_2^2. \quad (31)$$

Further applying the partition of unity property P2), we have

$$\left\| \dddot{\mathbf{P}} \mathbf{B}_{d-3,\boldsymbol{\xi}}(t) \right\|_2^2 \leq \omega_{\max}^2 \left\| \left(\ddot{\mathbf{P}} + \mathbf{z}_w g \mathbb{1}_{1\times(n-1)} \right) \mathbf{B}_{d-2,\boldsymbol{\xi}}(t) \right\|_2^2 \quad (32)$$

Since the lhs and lrs of (32) have different orders of the B-splines ($d-3$ and, respectively, $d-2$), we analyze the inequality over each knot vector sub-interval and make use of property P5) which allows to express any $d-3$ - order B-spline curve in terms of $d-2$ - order B-spline functions:

$$\dddot{\mathbf{P}} \mathbf{B}_{d-3,\boldsymbol{\xi}}(t) = \dddot{\mathbf{P}} \mathbf{D}_{d-3,d-2}^{\ell} \mathbf{B}_{d-2,\boldsymbol{\xi}}(t). \quad (33)$$

Thus, on each knot sub-interval ℓ, the following relation holds

$$\left\| \dddot{\mathbf{P}} \mathbf{D}_{d-3,d-2}^{\ell} \mathbf{B}_{d-2,\boldsymbol{\xi}}(t) \right\|_2^2 \leq \omega_{\max}^2 \left\| \left(\ddot{\mathbf{P}} + \mathbf{z}_w g \mathbb{1}_{1\times(n-1)} \right) \mathbf{B}_{d-2,\boldsymbol{\xi}}(t) \right\|_2^2. \quad (34)$$

Via the local support property P1), (34) simplifies to involve only the non-zero (on the ℓ-th sub-interval basis functions):

$$\left\| \sum_{k=\ell+d-2}^{\ell+d} (\dddot{\mathbf{P}} \mathbf{D}_{d-3,d-2}^{\ell})_k B_{k,d-2,\boldsymbol{\xi}}(t) \right\|_2^2 \leq \omega_{\max}^2 \left\| \sum_{k=\ell+d-2}^{\ell+d} \left(\ddot{\mathbf{P}}_k + \mathbf{z}_w g \right) B_{k,d-2,\boldsymbol{\xi}} \right\|_2^2. \quad (35)$$

With notation (13), (35) becomes

$$\sum_{k=\ell+d-2}^{\ell+d} \sum_{i=\ell+d-2}^{\ell+d} S(\ell, i, k, \mathbf{P}) B_{i,d-2,\boldsymbol{\xi}}^\top(t) B_{k,d-2,\boldsymbol{\xi}}(t) \leq 0. \quad (36)$$

Since the B-spline basis functions are positive (as per property P2)), a sufficient condition for (29) to hold on the sub-interval $[\tau_\ell, \tau_{\ell+1})$ is that

$$S(\ell, i, k, \mathbf{P}) \leq 0, \quad \forall i, k = \ell + d - 2, \ldots, \ell + d. \quad (37)$$

References

[1] A. Farooq, A. Anastasiou, N. Souli, C. Laoudias, P. S. Kolios, and T. Theocharides, "Uav autonomous indoor exploration and mapping for sar missions: Reflections from the icuas 2022 competition," in 2022 19th International Conference on Ubiquitous Robots (UR). IEEE, 2022, pp. 621–626.

[2] L. Markovic, F. Petric, A. Ivanovic, J. Goricanec, M. Car, M. Orsag, and S. Bogdan, "Towards a standardized aerial platform: ICUAS'22 firefighting competition," Journal of Intelligent &amp Robotic Systems, vol. 108, no. 3, July 2023. [Online]. Available: https://doi.org/10.1007/s10846-023-01909-z

[3] X. Li, Y. Zhao, J. Zhang, and Y. Dong, "A hybrid pso algorithm based flight path optimization for multiple agricultural uavs," in 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), 2016, pp. 691–697.

[4] S. G. Manyam, D. W. Casbeer, I. E. Weintraub, and C. Taylor, "Trajectory optimization for rendezvous planning using quadratic bézier curves," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, pp. 1405–1412.

[5] R. T. Rodrigues, N. Tsiogkas, A. Pascoal, and A. P. Aguiar, "Online range-based slam using b-spline surfaces," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1958–1965, 2021.

[6] V. Freire and X. Xu, "Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees," IEEE Transactions on Control Systems Technology, vol. 31, no. 6, pp. 2319–2334, 2023.

[7] B. Salamat and A. M. Tonello, "Stochastic trajectory generation using particle swarm optimization for quadrotor unmanned aerial vehicles (uavs)," Aerospace, vol. 4, no. 2, p. 27, 2017.

[8] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[9] M. ApS, The MOSEK optimization toolbox for MATLAB manual. Version 9.0., 2019. [Online]. Available: http://docs.mosek.com/9.0/toolbox/index.html

[10] S. Lai, M. Lan, and B. M. Chen, "Model predictive local motion planning with boundary state constrained primitives," IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3577–3584, 2019.

[11] L. Xi, X. Wang, L. Jiao, S. Lai, Z. Peng, and B. M. Chen, "Gto-mpc-based target chasing using a quadrotor in cluttered environments," IEEE Transactions on Industrial Electronics, vol. 69, no. 6, pp. 6026–6035, 2021.

[12] N. Naidja, S. Font, M. Revilloud, and G. Sandou, "An interactive game theory-pso based comprehensive framework for autonomous vehicle decision making and trajectory planning," in IFAC World Congress-22nd WC 2023, 2023.

[13] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve uav path planning in disaster scenarios," Knowledge-Based Systems, vol. 204, p. 106209, 2020.

[14] F. Stoican, I. Prodan, D. Popescu, and L. Ichim, "Constrained trajectory generation for uav systems using a b-spline parametrization," in 2017 25th Mediterranean Conference on Control and Automation (MED), 2017, pp. 613–618.

[15] M. Sheen. (2023) Fast 3d collision detection – gjk algorithm. GitHub project. [Online]. Available: https://github.com/mws262/MATLAB-GJK-Collision-Detection

[16] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, 2019.