On the application of the Schoenberg quasi-interpolant for complexity reduction in trajectory generation

Vincent Marguet, Florin Stoican, Ionela Prodan

On the application of the Schoenberg quasi-interpolant for complexity reduction in trajectory generation

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

***Abstract*— The paper extends previous work on trajectory generation for UAV (Unmanned Aerial Vehicles) using B-spline curves to parameterize an associated flat output. Typical constraints and costs (such as those involving input bounds and trajectory length) lead to nonlinear formulations in terms of the control points weighting the B-spline curve. This complexity adversely affects the computation time and conservatism of the result. To mitigate these effects we use the Schoenberg operator to provide a quasi-interpolant of the original nonlinear functions. These improvements come at the price of an approximation error which requires a tightening of the original constraints (either from a theoretical bound or via an iterative procedure). The obtained results are exemplified over a fixed wing UAV model and they can be applied for any optimization-based trajectory planning problem.**

***Index Terms*– Trajectory generation, B-spline curve, Schoenberg operator, approximation, optimization problem, UAV (Unmanned Aerial Vehicles), Fixed wing aircraft application.**

I. Introduction

Trajectory generation is a topic much studied in the literature, especially in the context of motion planning for unmanned aerial vehicles (UAVs). One popular approach is to solve an optimization problem: minimize a cost penalizing the trajectory's energy, time or length while also respecting various constraints (such as having to pass through pre-defined way-points) [1], [2]. Unfortunately, many times, the resulting trajectory cannot be accurately tracked when the robotic platform has non-negligible dynamics. E.g., in [3], a hybrid PSO algorithm is used to compute paths as straight segments linking farm blocks which have to be inspected by agricultural UAVs. This kind of trajectory may prove adequate for over-actuated platforms (such as quadcopters) but not for fixed-wing UAVs which have stringent constraints on the banking angle and the stall velocity.

One popular way to tackle the feasibility issue is to impose smoothness in the design stage by parameterizing the trajectory as weighted sum of basis functions which have the desired properties. [4] and [5] choose Bezier functions. It allows to define the trajectory as a combination of control points (the weights) and Bernstein polynomials. B-spline functions, an even more generic class, are also quite popular for trajectory generation like in [6], [7] or to generate a map of the environment as in [8]. The idea is to construct an optimization problem, via various manipulations and simplifications, such that the decision variables are the control points [9].

While solving a constrained optimization problem may lead to a provably feasible trajectory, the complexity of the problem depends strongly on the underlying dynamics, cost and constraints. Quite often these are described by nonlinear relations involving derivatives. Hence, the solution takes a long time to compute, is convoluted and conservative (the nonlinear formulations involving control point lead to spline curves far from their bounds).

The goal of this paper is to decouple the computational difficulties from the specifics of the problem at hand (i.e., how 'hard' the problem is, should be a decision taken in the design stages by refining various design parameters and not be decided implicitly by the platform's limitations). Hence, the contributions of this paper are:

i) employ the Schoenberg operator to approximate nonlinear functions (which stem from cost and/or constraints representation in an optimisation-based trajectory generation problem);
ii) present a motion planning algorithm which efficiently provides feasible trajectories via Schoenberg-based approximations.

Although the trajectory generation problem has been studied for decades, to the best of our knowledge, there is no paper using the Schoenberg operator in this context.

Section II recapitulates relevant definitions and properties of the B-spline curve, introduces the Schoenberg operator and presents the model of the fixed wing aircraft used in this paper to illustrate the results hereinafter. Section III states the main goal and provides an algorithm to solve a motion planning problem via Schoenberg-based approximations. Section IV illustrates the benefits of this approach through simulations. Section V draws the conclusions. The following notations are used throughout the paper.

II. Prerequisites

This section recalls some basic notions of B-spline basis parametrizations, introduces the Schoenberg operator [10] and the UAV model used for exemplification [11].

* This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissements d'avenir.

[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

Notation	Meaning
t	Time instant
T	Final time
n	Number of control points
p	Order of the B-spline curves
ξ	Knot vector used to define the B-spline basis functions
$\mathbf{P}$	Matrix gathering the control points defining a B-spline curve
$\mathbf{B}_{p,\xi}(t)$	B-spline basis function of order p and knot vector ξ
P_k	Column with the components of the $(k+1)$-th control point
$B_{k,p,\xi}$	$(k+1)$-th B-spline basis function of order p and knot vector ξ (ξ can be omitted to be less heavy)
τ_k	$(k+1)$-th time instant of the knot vector ξ
m	Number of components in the knot vector ξ -1
z	B-spline trajectory
$\dot{z}$	First derivative of z
$\ddot{z}$	Second derivative of z
Mat^T	Transpose matrix of Mat
$M_{p,p-r}$	Matrix used to calculate the derivatives of the B-splines until order r
$\tilde{t}_k$	$(k+1)$-th Greville point
$\tilde{f}$	Schoenberg operator applied to the function f (i.e. approximation of f)
$\|\Delta_n\|_\infty$	Maximal knot interval
$x(t)$	Position on x-axis at time t
$y(t)$	Position on y-axis at time t
$\psi(t)$	Yaw angle at time t
$\phi(t)$	Roll angle at time t
$V_a(t)$	Airspeed velocity at time t
g	Gravitational acceleration
p_j	Position of the j-th way-point

A. B-spline curve

We consider the parametrization of a trajectory $z(t)$ with B-spline curves as a linear combination of control points and B-spline basis functions:

$$z(t) = \sum_{k=0}^{n-1} P_k B_{k,p,\xi}(t) = \mathbf{P}\mathbf{B}_{p,\xi}(t), \forall t \in [0,T], \quad (1)$$

with $\mathbf{P} = \begin{bmatrix} P_0 & \dots & P_{n-1} \end{bmatrix}$ the matrix which gathers the control points, $\mathbf{B}_{p,\xi}(t) = \begin{bmatrix} B_{0,p,\xi}(t) & \dots & B_{n-1,p,\xi}(t) \end{bmatrix}^\top$ the basis vector, $\xi = \{\tau_0 \leq \tau_1 \leq ... \leq \tau_m\}$ a knot sequence starting at 0 and ending at T. If $m \geq p+2$, we can define B-splines of order up to p over the knot sequence.

The k-th B-spline basis function of order p is defined recursively by:

$$B_{k,1,\xi}(t) = \begin{cases} 1, & t \in [\tau_k; \tau_{k+1}[\\ 0, & \text{otherwise,} \end{cases} \quad (2a)$$

$$B_{k,p,\xi}(t) = \frac{t-\tau_k}{\tau_{k+p}-\tau_k} B_{k,p-1,\xi}(t) + \\ + \frac{\tau_{k+p+1}-t}{\tau_{k+p+1}-\tau_{k+1}} B_{k+1,p-1,\xi}(t). \quad (2b)$$

This particular function has many properties [12]:

P1) Each B-spline basis function has a local support:

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

P2) The B-spline functions partition the unity:

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

and

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

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

$$\mathbf{B_{p,\xi}}^{(r)}(t) = M_{p,p-r}\mathbf{B_{p-r,\xi}}(t) \quad (5)$$

P4) $B_{k,p,\xi}(\tau_l) \in \mathcal{C}^{p-\mu_l}$ at $\tau_l \in \xi$ with multiplicity μ_l and $\mathcal{C}^\infty$ otherwise.

P5) Convexity property: The B-spline curve lies within the union of all convex hulls defined by subsets of p consecutive control points.

Remark 1. *The property **P5)** implies that taking the p first values of the knot vector equal to 0, the first control point will be the starting point of the curve. For the same reason, taking the p last values of the knot vector equal to T will have the consequence of setting the final point of the curve at the last control point. This is why the knot vector used in this paper will be defined as*

$$\xi = \Big\{ \tau_0 = \dots = \tau_{p-1} < \tau_p < \\ \dots < \tau_{m-p} < \tau_{m-p+1} = \dots = \tau_m \Big\}. \quad (6)$$

For further use, note that we take $n = m - p + 1$.

B. Schoenberg operator

The Schoenberg operator [10], [13], [14] approximates a function (call it $f(\cdot)$) by interpolating it as weighted sum of B-splines basis functions and the function's values in the Greville points of the B-spline basis:

$$\tilde{f}(t) = \sum_{k=0}^{\tilde{n}-1} f(\tilde{t}_k) \tilde{B}_{k,\tilde{p},\tilde{\xi}}(t), \quad (7)$$

where the Greville points are:

$$\tilde{t}_k = \frac{\tilde{\tau}_{k+1} + ... + \tilde{\tau}_{k+\tilde{p}-1}}{\tilde{p}-1}. \quad (8)$$

Remark 2. *Note the presence of the 'tilde' symbol in (7): even if the function $f(\cdot)$ to be approximated is a combination of B-splines, it is not necessary that the same family of B-splines is used in (7), hence, $\tilde{p}, \tilde{\xi}$ may differ.*

C. Fixed wing aircraft model

Let us consider a 2D 3-DOF model of a fixed wing UAV, [12]. The position ($x(t)$, $y(t)$) and the heading (yaw) angle $\psi(t) \in [0; 2\pi]$ rad are the state variables. Their dynamics are described by the following equations,

where $V_a(t)$ and $\phi(t)$ are the inputs, the airspeed velocity and the roll angle, respectively:

$$\dot{x}(t) = V_a(t)\cos\psi(t), \tag{9a}$$
$$\dot{y}(t) = V_a(t)\sin\psi(t), \tag{9b}$$
$$\dot{\psi}(t) = \frac{g\,\tan\phi(t)}{V_a(t)}. \tag{9c}$$

We take as flat output (see [15], [11] for more details on flatness and this UAV's flat representation) the position of the aircraft $z(t) = \begin{bmatrix} z_1(t) & z_2(t) \end{bmatrix}^\top = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^\top$. Thus, we are able to express all the state variables and inputs via the flat output and its first and second derivatives:

$$x(t) = z_1(t), \tag{10a}$$
$$y(t) = z_2(t), \tag{10b}$$
$$\psi(t) = \arctan\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right), \tag{10c}$$
$$V_a(t) = \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}, \tag{10d}$$
$$\phi(t) = \arctan\left(\frac{1}{g}\frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}}\right). \tag{10e}$$

This will allow to reformulate the various costs/constraints in terms of the flat output $z(t)$ which will be further parameterized as in (1).

III. Schoenberg-based approximation

Motion planning often reduce to solving a constrained optimization problem of the form

$$\min_{z(t)} \quad \int_0^T \mathcal{C}(z(t))\mathrm{d}t \tag{11a}$$
$$\text{s.t.} \quad g_i(z(t)) \leq 0,\ 0 \leq t \leq T, \tag{11b}$$
$$h_j(z(t)) = 0,\ 0 \leq t \leq T. \tag{11c}$$

The goal is to find a trajectory $z(t)$ which minimizes the cost $\mathcal{C}(\cdot)$ from (11a) while simultaneously respecting (11b) and (11c) over the time interval $[0, T]$.

To better illustrate the subsequent ideas let us particularize (11) by taking $z(t)$ as in (10). Then, we have:

- the cost $\mathcal{C}(\cdot)$ minimizes the trajectory length along the time horizon $[0, T]$;
- the bounds on the velocity's magnitude are denoted by $\underline{v}$ and $\overline{v}$, respectively;
- the bounds on the roll angle are denoted by $\underline{\phi}$ and $\overline{\phi}$, respectively;
- the way-points passing (through p_j at t_j) conditions.

Thus, (11) is instantiated to the particular form:

$$\min_{z(t)} \quad \int_0^T \|\dot{z}(t)\|\mathrm{d}t \tag{12a}$$
$$\text{s.t.} \quad \underline{v} \leq \|\dot{z}(t)\| \leq \overline{v},\ \forall t \in [0, T] \tag{12b}$$
$$\underline{\phi} \leq \phi(t) \leq \overline{\phi},\ \forall t \in [0, T] \tag{12c}$$
$$z(t_j) = p_j, \forall j. \tag{12d}$$

Except very specific cases (e.g., computing Dubins trajectories for the unicycle model) it is not possible to solve (11), or (12), for an arbitrary $z(t)$. The usual approach is to project $z(t)$ onto a basis of functions and reformulate (11), or (12), into a more manageable discrete optimization problem. Specifically, parameterizing $z(t)$ as in (1) allows to reformulate[1] (12) into

$$\min_{P_k, B_k(t)} \quad \int_0^T f_c(t)\mathrm{d}t, \tag{13a}$$
$$\text{s.t.} \quad \underline{v}^2 \leq f_v(t) \leq \overline{v}^2, \tag{13b}$$
$$\tan(\underline{\phi}) \leq f_\phi(t) \leq \tan(\overline{\phi}), \tag{13c}$$
$$\sum_{k=0}^{n-1} P_k B_{k,p}(t_j) = p_j, \forall j, \tag{13d}$$

with the notations

$$f_c(t) = \sqrt{\sum_{k_1,k_2=0}^{n-2} \left(P'_{k_1}\right)^\top P'_{k_2} B_{k_1,p-1}(t) B_{k_2,p-1}(t)}, \tag{14a}$$
$$f_v(t) = \sum_{k_1,k_2=0}^{n-2} \left(P'_{k_1}\right)^\top P'_{k_2} B_{k_1,p-1}(t) B_{k_2,p-1}(t), \tag{14b}$$
$$f_\phi(t) = \frac{\sum_{k_1=0}^{n-3}\sum_{k_2=0}^{n-2} \left(P''_{k_1}\right)^\top Q P'_{k_2} B_{k_1,p-2}(t) B_{k_2,p-1}(t)}{g\sqrt{\sum_{k_1,k_2=0}^{n-2} \left(P'_{k_1}\right)^\top P'_{k_2} B_{k_1,p-1}(t) B_{k_2,p-1}(t)}}, \tag{14c}$$

where $Q = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and P'_k, P''_k are used as short-hands for the control points associated to the first and second order derivatives of the B-spline curve (as per Property **P3)**, the dependencies are linear).

Problem (13) still explicitly considers time as a variable. To bring it to a discrete time variant we exploit B-spline properties and provide sufficient conditions involving the associated control points.

Condition (13b) is a good example of this. Starting from (1) and applying Property **P3)** we arrive at $\dot{z}(t) = \sum_{k=0}^{n-2} P'_k B_{k,p-1}(t)$ which in turn means that $\|\dot{z}(t)\|^2 = \sum_{k_1,k_2=0}^{n-2} \left(P'_{k_1}\right)^\top P'_{k_2} B_{k_1,p-1}(t) B_{k_2,p-1}(t)$. The $(n-1)^2$ terms $B_{k_1,p-1}(t)B_{k_2,p-1}(t)$, based on Property **P2)**, also partition the unity which means that $\|\dot{z}(t)\|^2$ lies within the convex hull of all combinations $\left(P'_{k_1}\right)^\top P'_{k_2}$, hence

$$\underline{v}^2 \leq \left(P'_{k_1}\right)^\top P'_{k_2} \leq \overline{v}^2, \quad 0 \leq k_{\{1,2\}} \leq n-2, \tag{15}$$

is a sufficient condition for (12b). With varying degrees of success, the same reasoning may be applied to any term from (13).

We notice that the problem depends now on a finite number of variables (the control points P_k) meaning that

[1]Whenever clear from context, we simplify the notation and do not write the knot vector, hence, $B_{k,p}(t)$ instead of $B_{k,p,\xi}(t)$.

it can be posed and solved as a standard constrained optimization problem. This comes with several drawbacks:

- the number of control points, the order and knot vector of the B-splines restrict the shape of $z(t)$;
- some sufficient formulations of the constraints are conservative (see, e.g., (15) wrt (13b));
- some terms (specifically those appearing in (13a)–(13c)) are nonlinear.

Remark 3. *Already in* (13) *we observe a relatively complex formulation for requirements which are in essence quite simple. Even worse, the difficulty of the problem is not entirely 'controllable' at the design stage: while we may choose the basis function parameters (their number n and order p) we cannot control the complexity of the functions which describe the constraints/cost (since they are directly influenced by the model).* ♦

Remark 4. *Two shortcomings, induced by weighting* (1)*, are best illustrated by* (13b)*. First is that the number of inequalities has increased significantly ($2(n-1)^2$ in* (15)*). Second, $\|\dot{z}(t)\|^2$ will be 'far' from its bounds, $\left(P'_{k_1}\right)^\top P'_{k_2}$, thus making* (15) *conservative with respect to* (13b)*. This is even worse for* (13c)*, where, by the same reasoning, there would be $2(n-2)^3(n-3)$ conditions to test.* ♦

To handle the issues raised in Remarks 3 and 4 we make use of the Schoenberg operator introduced in Section II-B to approximate terms appearing in cost and/or constraints. Using again the illustrative example from (13) we apply (7) thrice, once to approximate the nonlinear cost (13a), once to approximate the velocity in (13b) and once to approximate the roll angle in (13c). Correspondingly, the approximations are

$$\tilde{f}_{\{c,v,\phi\}}(t) = \sum_{k=0}^{\tilde{n}-1} f_{\{c,v,\phi\}}(\tilde{t}_k) B_{k,\tilde{p},\tilde{\xi}}(t), \tag{16}$$

which allow to:

i) approximate the cost $\int_0^T f_c(t)\mathrm{d}t$ from (13a) as

$$\begin{aligned}\int_0^T \tilde{f}_c(t)\mathrm{d}t &= \int_0^T \sum_{k=0}^{\tilde{n}-1} f_c(\tilde{t}_k) B_{k,\tilde{p},\tilde{\xi}}(t) \\ &= \sum_{k=0}^{\tilde{n}-1} \left[f_c(\tilde{t}_k) \int_0^T B_{k,\tilde{p},\tilde{\xi}}(t)\mathrm{d}t \right], \end{aligned} \tag{17}$$

ii) approximate the velocity inequalities $\underline{v}^2 \leq f_c(t) \leq \overline{v}^2$ from (13b) by

$$\underline{v}^2 \leq \sum_{k=0}^{\tilde{n}-1} f_v(\tilde{t}_k) B_{k,\tilde{p},\tilde{\xi}}(t) \leq \overline{v}^2, \tag{18}$$

iii) approximate the roll angle inequalities $\tan(\underline{\phi}) \leq f_\phi(t) \leq \tan(\overline{\phi})$ from (13c) by

$$\tan(\underline{\phi}) \leq \sum_{k=0}^{\tilde{n}-1} f_\phi(\tilde{t}_k) B_{k,\tilde{p},\tilde{\xi}}(t) \leq \tan(\overline{\phi}). \tag{19}$$

Introducing approximations (17)–(19) and keeping the already linear equality (13d) leads to the approximated constrained-optimization problem

$$\min_{P_k} \quad \epsilon^2 \tag{20a}$$

$$-\epsilon \leq \sum_{k=0}^{\tilde{n}-1} \left[f_c(\tilde{t}_k) \int_0^T B_{k,\tilde{p}}(t)\mathrm{d}t \right] \leq \epsilon, \tag{20b}$$

$$\text{s.t.} \quad \underline{v}^2 \leq f_v(\tilde{t}_k) \leq \overline{v}^2,\ 0 \leq k \leq \tilde{n}-1, \tag{20c}$$

$$\tan(\underline{\phi}) \leq f_\phi(\tilde{t}_k) \leq \tan(\overline{\phi}),\ 0 \leq k \leq \tilde{n}-1, \tag{20d}$$

$$p_j = \sum_{k=0}^{n-1} P_k B_{k,p}(t_j), \forall j. \tag{20e}$$

Note that (20) neatly tackles the shortcomings identified earlier: the cost's non-linearity is pushed into the constraints via the addition of slack variable ϵ and the constraints' number is reduced from $2(n-1)^2 + 2(n-2)(n-1)^3$ to $4\tilde{n}$ when comparing (20c)–(20d) with (13b)–(13c). To sum up, the initial problem (12) is restated into the B-spline parameterized problem (13) via mapping (1), next put into a sufficient formulation as a discrete constrained optimization via (15) and the like, which is then brought to an approximated but simplified form in (20) via the application of (7) in (17)–(19).

Building on the particularities of (9), we highlight that the reasoning holds for the generic motion problem (11), as described in Algorithm 1.

Algorithm 1: Solving a motion planning problem via Schoenberg-based approximations.

Input: list of functions defining problem (11); parameters p, n, ξ appearing in (1);
Output: feasible trajectory $z(t)$;

1 Construct an initial B-spline basis as in (2);
2 Express the terms appearing in (11) as (possibly) nonlinear combinations of control points and splines (as is done in (14) for (13));
3 Provide approximations based on the Schoenberg operator (7), as is done in (16) for (14);
4 Introduce these approximations into the initial problem and solve it (i.e., obtain the control points P_k), similar to (13) versus (20);
5 Introduce the control points in (1) to obtain the feasible trajectory $z(t)$. Use it and its derivatives to provide the dynamics' inputs.

Hidden under the rug is the effect of the approximation error: we solve (20) for simplified functions $\tilde{f}$ but the original cost and constraints apply to the 'real' functions f. This may be handled by increasing the number of elements in the approximation basis $\{B_{0,\tilde{p},\tilde{\xi}}, \dots B_{\tilde{n}-1,\tilde{p},\tilde{\xi}}\}$ (the fact that we used the same $\tilde{p}, \tilde{n}, \tilde{\xi}$ in (14) was for convenience; their values may have changed from (14a) to (14b)) but this quickly runs into numerical issues.

TABLE I

NUMERICAL VALUES OF THE SIMULATIONS.

Simulation data	Numerical Value
Gravitational acceleration g	9.81 m/s^2
Velocity control input	$V_a \in$ [18 25] m/s
Bank control input	$\phi \in [-0.43\ 0.43]$ rad
Altitude	150 m
Simulation time T	150 s
Way-point list	$\mathbb{P}$={(1000,1000,150),(1100,250,150), (400,0,150),(-250,250,150), (400,500,150)}
Number of control points n	10
Order of the B-spline curves p	4
Number of greville points $\tilde{n}$	20

IV. ILLUSTRATIVE EXAMPLE FOR AN UAV SYSTEM

The ideas discussed in Section III are illustrated by the UAV model described in (II-C). We remind that the initial problem (13) cannot be solved because of the high number of constraints and their complexity. Hence, we underline that the interest of applying the Schoenberg operator to approximate the constraints and the cost function is to solve the approximated optimization problem and obtain a feasible solution. Due to the approximation error, some constraints may have to be tightened, that is why we need to confirm that the control points obtained in (20) are a solution of the initial problem (13). The numerical values for the constraints in the optimization problem (20) are taken from the experimental results proposed in [11].

Every 37.5 seconds of the simulation, the UAV has to be at the next way-point. As delineated in Table I, the roll angle limits are $\underline{\phi} = -\overline{\phi} = -0.43$ rad, and tan is an odd and increasing function in the interval [-0.43;+0.43]. Hence, the constraint (20d) can be rewritten as

$$0 \leq |f_\phi(\tilde{t}_k)| \leq \tan(\overline{\phi}), \quad 0 \leq k \leq \tilde{n} - 1, \tag{21}$$

Considering the square of the above constraint we have:

$$f_\phi^2(\tilde{t}_k) \leq \tan^2(\overline{\phi}), \quad 0 \leq k \leq \tilde{n} - 1, \tag{22}$$

To resume, the optimization problem (20) is composed of $3\tilde{n}$ inequality constraints ($\tilde{n}$ inequality constraints for the minimum velocity, $\tilde{n}$ inequality constraints for the maximum velocity, $\tilde{n}$ inequality constraints for the roll angle) and the constraints for the way-points passing.

The optimization problem (20) with the numerical values from Table I was run in Matlab 2021a using the IPOPT solver. The velocity constraint is represented in Fig 1: the black stars representing $\sqrt{f_v(\tilde{t}_k)}$ were used as constraints: they have to stay between v_{min} and v_{max} and it guarantees that the approximated function $\sqrt{\tilde{f}_v(t)}$ (in blue) lies within these bounds. It means that the approximated function respects the constraint.

However, the objective is that the initial function (in black) respects the constraints. We see, that it is almost

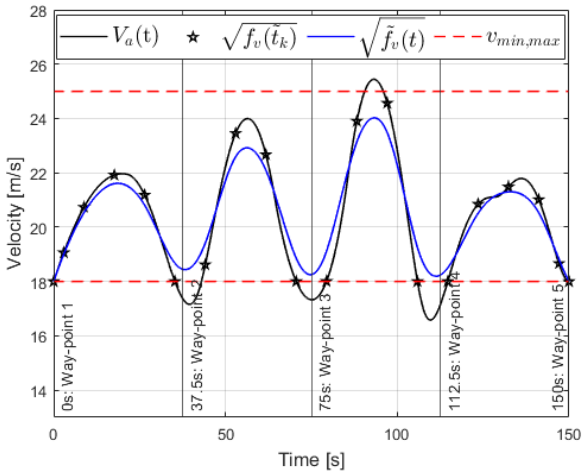

Fig. 1. Evolution of the airspeed velocity V_a when the velocity constraint is not tightened.

the case. To respect the constraint during the whole simulation, we have to tighten the constraints applied in the optimization problem. Indeed, as explained in the previous section there is an approximation error caused by the Schoenberg operator. A study of the bounds based on the control points to describe this approximation error would be of interest to offer feasibility guarantees, i.e., the control points obtained from the simplified problem (such as (20)) lead to a feasible solution in the initial problem (13).

Thus, we replace the velocity interval $[v_{min}; v_{max}]$ by $\mathbb{I}_v = [v_{min} + 0.2(v_{max} - v_{min}); v_{max} - 0.2(v_{max} - v_{min})]$. The results taking into account this tightening on the velocity bound are delineated in the next illustrations: Fig 2, 3 and 4 show that the way-points, velocity and the roll angle constraints are satisfied by the approximations and by the initial curves. The computational time analysis results are presented in Table II. This approximation method clearly reduced the computational time to solve the optimization problem by reducing the number of constraints.

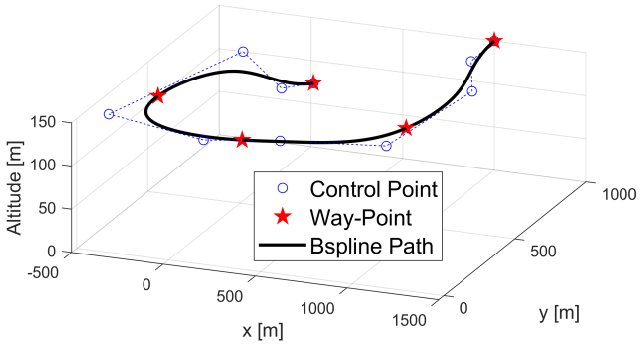

Fig. 2. Path of the fixed wing aircraft obtained when the velocity constraint is tightened to $\mathbb{I}_v$.

We have considered the iterative procedure from Algo-

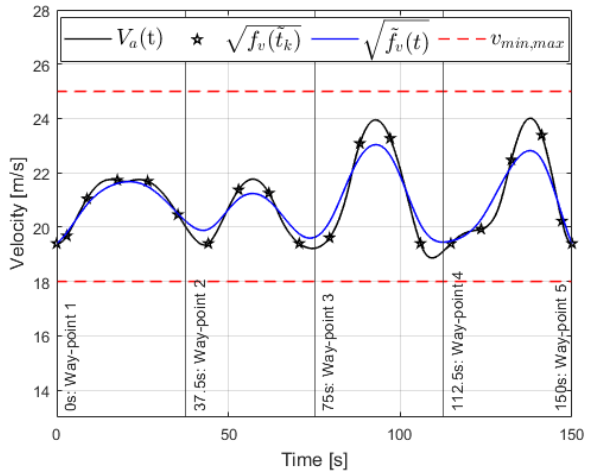

Fig. 3. Evolution of the airspeed velocity V_a when the velocity constraint is tightened to $\mathbb{I}_v$.

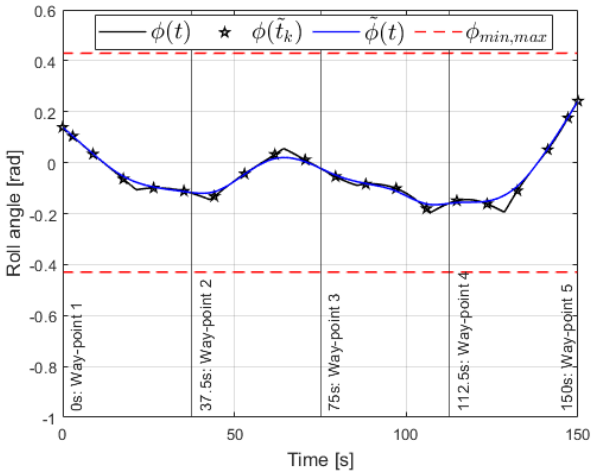

Fig. 4. Evolution of the roll angle ϕ when the velocity constraint is tightened to $\mathbb{I}_v$.

TABLE II

Computational time analysis for the approximated curves.

Simulation Results	Numerical Value
Number of inequality constraints of (13)	∞
Computational time of (13)	–
Number of inequality constraints of (20)	60
Computational time of (20)	67 s
Number of iterations to solve (20)	176

rithm 1 where the original constraints are progressively tightened until the approximated problem provides a feasible solution for the original one.

V. Conclusion

This paper presents an algorithm to approximate the functions representing the cost function and the constraints of an optimization problem with a better computational time. Reducing the computational time is even more relevant, when the trajectory needs to be computed on-line. The Schoenberg quasi-interpolant is applied to reduce the complexity of the optimal B-spline curve generated for a fixed-wing aircraft. Furthermore, this algorithm can be applied to every system which admits a flat representation. Future work will concern the approximation error caused by the Schoenberg operator and the extension of this algorithm to multiple UAVs, taking into account "inter-UAV" constraints like communication range among them and collision avoidance.

References

[1] D. Saccani and L. Fagiano, "Autonomous uav navigation in an unknown environment via multi-trajectory model predictive control," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 1577–1582.

[2] X. Li, X. Gao, W. Zhang, and L. Hao, "Smooth and collision-free trajectory generation in cluttered environments using cubic b-spline form," *Mechanism and Machine Theory*, vol. 169, p. 104606, 2022.

[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)*. IEEE, 2016, pp. 691–697.

[4] G. J. Yang and B. W. Choi, "Smooth trajectory planning along bezier curve for mobile robots with velocity constraints," *International Journal of Control and Automation*, vol. 6, no. 2, pp. 225–234, 2013.

[5] B. Sabetghadam, R. Cunha, and A. Pascoal, "Real-time trajectory generation for multiple drones using bézier curves," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9276–9281, 2020.

[6] N. T. Nguyen, L. Schilling, M. S. Angern, H. Hamann, F. Ernst, and G. Schildbach, "B-spline path planner for safe navigation of mobile robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 339–345.

[7] Y. Hervagault, I. Prodan, and L. Lefevre, "Trajectory generation with communication-induced constraints for surface vehicles," in *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2017, pp. 482–487.

[8] 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.

[9] F. Stoican, A. Postolache, and I. Prodan, "Nurbs-based trajectory design for motion planning in a multi-obstacle environment," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 2014–2019.

[10] T. Lyche, C. Manni, and H. Speleers, "Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement," in *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*. Springer, 2018, pp. 1–76.

[11] I. Prodan, S. Olaru, R. Bencatel, J. B. de Sousa, C. Stoica, and S.-I. Niculescu, "Receding horizon flight control for trajectory tracking of autonomous aerial vehicles," *Control Engineering Practice*, vol. 21, no. 10, pp. 1334–1349, 2013.

[12] 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)*. IEEE, 2017, pp. 613–618.

[13] L. Beutel, H. Gonska, D. Kacsó, and G. Tachev, "On variation-diminishing schoenberg operators: new quantitative statements," *Multivariate Approximation and Interpoltaion with Applications (ed. by M. Gasca), Monogr. Academia Ciencas de Zaragoza*, vol. 20, pp. 9–58, 2002.

[14] T. Zapryanova and G. Tachev, "Generalized inverse theorem for schoenberg operator," *Journal of Modern Mathematics Frontier*, vol. 1, no. 2, pp. 11–16, 2012.

[15] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International journal of control*, vol. 61, no. 6, pp. 1327–1361, 1995.