A General Approach for Planning a Smooth Planar Path Within a Channel Using NURBS

Marco Riboli, Elisabetta Manconi, Rinaldo Garziera, and Alessandra Aimi

***Abstract*—This letter presents a new and comprehensive approach to planning a smooth and optimal path in a planar channel. The algorithm generates a NURBS that traverses the channel, in turn defined by two bounding rational splines, and stays inside. A nonlinearly constrained problem is solved to model the shape of the two-sided bounded path to optimise the objective function. The user can choose the degree of the curve that defines the resulting path and optionally add the interpolation of waypoints and orientations. The presented method expands the domain of feasible solutions compared to previous approaches based on nonrational splines, leading to improvements in terms of the objective function. Various examples are provided to illustrate and clarify the method.**

***Index Terms*—Path planning, collision avoidance, NURBS, channel problem, nonlinear programming.**

I. Introduction

THE generation of smooth curves confined in a channel is a recurring problem in various engineering applications. Some of these, which we consider current and relevant in their respective fields, are: collision-free path planning for autonomous systems [1], [2], [3], [4], [5], curvature-optimised path generation for machining [6], cartographic generalization of isobathymetric lines [7], and isogeometric analysis [8]. Considering the first of the mentioned applications, an interesting example is proposed in [9] in the field of endovascular navigation. In that work, a smooth and energy-optimal trajectory for a microrobot is planned using a GA (Genetic Algorithm) based on nonrational splines, taking into account the boundary constraints of the vessel network. In this letter, we focus on collision-free path planning applications of the presented algorithm, with the understanding that the study can be adapted and applied to other application fields.

Strategies for path planning in environments with obstacles can be broadly classified into sampling-based and optimisation-based path planners [10]. Focusing on optimisation-based path planners, the problem is addressed by defining an optimisation scheme to plan a feasible, locally or globally optimal path. Optimisation metrics depend on the application case, e.g., time [11], energy consumption [12], time and jerk [13], and a combination of multiple criteria [14]. Although this approach potentially offers a high quality of motion, the effective definition of the model is often challenging. This typically involves employing a two-step approach, where a distance-minimising path-finding algorithm, such as Dijkstra, A* or their variants, precedes the optimisation step to select a simply-connected portion of the environment map. In this feasible region, the path can be optimised without the risk of collisions and is often defined by two non-intersecting curves defining a channel.

The contribution of this letter focuses on this second step, providing a general optimisation-based procedure for defining the optimal path within a planar channel. Compared to previous works, the path parameterisation is provided in NURBS (Non-Uniform Rational B-Spline) format, and the channel geometry is generally delimited by nonlinear boundaries. The solution proposed in this letter extends established methodologies reported in the literature, which are thoroughly detailed in the following sections. This extension leads to improved motion performance and simplifies the automation of CAD/CAM geometry loading.

The remainder of this letter is organised as follows. Section II offers an overview of the main CAD techniques employed in channel problem solution. Section III explains the proposed method by detailing the mathematical formulation of the problem. Section IV proposes various examples of our method. Section V summarises the work and results.

The mathematical notations used in the remainder of the letter, not specified in the text, are summarized below.

Notation
$\triangleq$ is equal by definition.
$\mathrm{pow}(x, n)$ is x raised to the power n.
$a \mathrm{mod} b$ is the remainder after a is divided by b.
$\boldsymbol{x}$, $[x^i]_{i=1}^n$, and $[x^1 \dots x^n]$ are the same row n-vector.
$\{\boldsymbol{x} \mid condition\}$, with $\boldsymbol{x} \in \mathbb{R}^n$, is a subset of $\mathbb{R}^n$.
$\boldsymbol{ab}$, with $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, is the n-vector $\boldsymbol{b} - \boldsymbol{a}$.
$\det \boldsymbol{M}$ is the determinant of the matrix $\boldsymbol{M}$.
$\mathbf{1}$ is the unitary vector $[1 \dots 1]$.
$\wedge$ is logical and.
$\| \cdot \|$ is Euclidean norm.
$p_r(\tau) : \mathbb{R} \to \mathbb{R}$ is the r^{th} dimensional component of the function $\boldsymbol{p}(\tau) : \mathbb{R} \to \mathbb{R}^n$, with $r = 1, \dots, n$.
e^k is the k^{th} breakpoint of the polyline $\boldsymbol{e}(\tau)$.

Manuscript received 10 February 2024; accepted 28 May 2024. Date of publication 11 June 2024; date of current version 17 June 2024. This letter was recommended for publication by Associate Editor T. Do and Editor A. Bera upon evaluation of the reviewers' comments. This work was supported by the University of Parma through the action Bando di Ateneo 2021 per la ricerca co-funded by MUR-Italian Ministry of Universities and Research - D.M. 737/2021 - PNR - PNRR - NextGenerationEU. *(Corresponding author: Marco Riboli.)*

Marco Riboli, Elisabetta Manconi, and Rinaldo Garziera are with the Department of Industrial Engineering Systems and Technologies, University of Parma, 43124 Parma, Italy (e-mail: marco.riboli@unipr.it; elisabetta.manconi@unipr.it; rinaldo.garziera@unipr.it).

Alessandra Aimi is with the Department of Mathematical, Physical and Computer Sciences, University of Parma, 43124 Parma, Italy (e-mail: alessandra.aimi@unipr.it).

Digital Object Identifier 10.1109/LRA.2024.3412636

II. Related Work

In the field of path planning, B-splines are frequently used in constrained optimisation problems. In addition to allowing smooth movements, this parameterisation ensures that the spline is closely confined within the convex hull of its control polygon [15]. Consequently, the path can be easily approximated by it (or formulations derived from it), and the quality of this approximation can be improved by refining the control polygon with knot insertion techniques. This property greatly simplifies the definition of constraints and the objective function, which can often be modelled linearly.

Lutterkort and Peters [16], [17] propose a piecewise linear envelope with few pieces based on the convex hull property, which has found many applications in the field of collision-free path planning. This algorithm is the basis of the channel problem, so called again by Lutterkort and Peters in [18], where a first highly efficient 1-D solution is presented in terms of linearly constrained optimisation. By channel problem, we refer to the search for a function within a simply connected domain, defined by explicit geometric representations of two non-intersecting curves, to optimise an objective function while satisfying specific requirements. In [18], a piecewise linear enclosure for a candidate curve is defined with the previously mentioned envelope algorithm, and the problem is solved by fitting the enclosure into the channel imposing linear inequality constraints. This approach has recently found application in an industrial context in the field of sorting systems for laser cutting machines [19]. In that work, the solution to the channel problem defines deviation splines to be added to the pre-computed paths of the two arms of the machine to avoid collisions.

A first extension to a multidimensional domain of the previously analysed envelope algorithm was also provided by Lutterkort and Peters [20]. In that work, multidimensional bounding boxes are defined by applying the 1-D envelope algorithms for each dimensional component of the spline. This approach allows the definition of a n-D enclosure [20], indicated with the term SLEFE (Subdividable Linear Efficient Function Enclosure) in the literature [21]. SLEFEs have found several engineering applications in the collision-free motion planning of autonomous vehicles [1], [2], [5], providing some generalisations of the channel problem to a planar domain. For example, Berglund et al. [2] propose an automatic generator of minimal-curvature planar paths for autonomous mining vehicles based on 2-D channel solution. Another interesting application is provided in [5], where a 2-D smooth path generator for UAVs (Unmanned Aerial Vehicles) in a confined environment was presented. The domain is progressively defined as a sequence of collision-free cells obtained from a high-level discrete planner based on a D*-lite algorithm. This solution allows a step-by-step construction of the channel, leading to solve a small constrained optimisation problem at each step and ensuring adequate performance for real-time applications [3], [4].

One of the most significant contributions in this research direction is the work by Myles and Peters [22]. That work proposes a method for determining a 3-D nonrational spline with few pieces that traverse a spatial channel and stay within it, solving an LP (Linear Programming) problem. The optimisation metric is defined by a linear objective function that penalises fluctuations and loops, while the non-interference constraints are described linearly, setting a strict space-invariant correspondence between the curve SLEFE and the channel. While the work only suggests potential practical applications in the field of collision-free motion planning, a few years later, Selinger and Linsen apply this theory to generate G^2-continuous paths with a guaranteed error bound for 3-axis machining [23], thus confirming the method's effectiveness.

Another class of studies is conducted by Kano and Fujioka [24], [25], [26]. Their most recent work focuses on planning planar paths confined in a polygonal channel, parameterising the result with a spline that uses normalised uniform B-splines as the basis functions. The resulting path is obtained by solving a LCQP (Linearly Constrained Quadratic Programming) problem. The objective function allows the channel's centerline to be approximated by a smoothness controlled by a user-settable parameter. Non-interference constraints with channel boundaries can be added with linear inequalities as a function of the control polygon without using a SLEFE definition. This offers the advantage of not using slack variables, which would otherwise be necessary to linearise the min-max selections in the enclosure definition, thereby reducing the size of the optimisation problem. It is also discussed in [26] how that method can be applied to channels with nonlinear boundaries. In this regard, an example of a circular channel approximated by an n-sided polygon is provided. However, this approach is designed for that specific type of route and is not a general methodology valid for other types of channels.

Despite considerable progress in this research direction, the current state of the art lacks an assessment of the advantages of using rational splines in the channel problem formulation. Although polynomial splines offer numerous advantages, they are limited in accurately representing some families of curves, such as circles or ellipses, except in an approximate way [28]. These constraints delineate a potential space for enhancement in the optimization process of the aforementioned methodologies, defined by a set of feasible solutions that cannot be represented by nonrational splines. We summarize our contributions as follows:

1) We propose a solution to the channel problem based on NURBS, and show through several examples its significant advantage over using nonrational splines in terms of objective function minimization.
2) We present a comprehensive algorithm for generating a polygonal channel from nonlinear boundaries parameterised by rational splines.
3) We introduce a new algorithm for enveloping NURBS curves with a few linear pieces.

III. Method

A. Theoretical Background

A n-D spline of degree d is a piecewise polynomial function $\boldsymbol{p} : \mathbb{R} \to \mathbb{R}^n$ in the parametric variable $\tau \in \mathbb{R}$. In its B-form, it is defined by the linear combination:

$$\boldsymbol{p}(\tau) = \sum_{k=0}^{m} \boldsymbol{b}^k N^k_{d,\boldsymbol{u}}(\tau) \quad u^0 \le \tau \le u^{m+d+1} \tag{1}$$

where m is the maximum index of the control points vector $\boldsymbol{b} = [\boldsymbol{b}^0 \dots \boldsymbol{b}^m]$, with $\boldsymbol{b}^k \in \mathbb{R}^n$, and $N^k_{d,\boldsymbol{u}}(\tau)$ are the basis functions defined by [27]:

$$\begin{cases} N^k_{0,\boldsymbol{u}}(\tau) = \begin{cases} 1, & \text{if } \ u^i \le \tau < u^{i+1} \\ 0, & \text{otherwise} \end{cases} \\ N^k_{d,\boldsymbol{u}}(\tau) = \frac{\tau - u^k}{u^{k+d} - u^k} N^k_{d-1,\boldsymbol{u}}(\tau) \\ \quad + \frac{u^{k+d+1} - \tau}{u^{k+d+1} - u^{k+1}} N^{k+1}_{d-1,\boldsymbol{u}}(\tau) \end{cases},$$

based on a non-decreasing knot sequence $\boldsymbol{u} = [u^0 \dots u^{m+d+1}]$. In this work, we only use non-periodic and non-uniform knots, therefore subject to the conditions $u^0 = \cdots = u^d$, $u^m = \cdots = u^{m+d+1}$.

When more flexibility in graphical representation is necessary, rational variants of B-splines, known as NURBS, are frequently employed. These functions are obtained from (1) by substituting the basis functions $N^k_{d,\boldsymbol{u}}(\tau)$ with the rational basis functions $R^k_{d,\boldsymbol{u}}(\tau)$, defined by:

$$R^k_{d,\boldsymbol{u}}(\tau) = \frac{\omega^k N^k_{d,\boldsymbol{u}}(\tau)}{\sum_{i=0}^{m} \omega^i N^i_{d,\boldsymbol{u}}(\tau)}$$

where $\boldsymbol{\omega} = [\omega^0 \dots \omega^m]$ are scalars called weights. We should note that, by exploiting the unit partition property of the basis functions [28], it can be easily demonstrated that the condition $\boldsymbol{\omega} = \mathbf{1}$ leads to $R^k_{d,\boldsymbol{u}}(\tau) = N^k_{d,\boldsymbol{u}}(\tau)$. Consequently, a rational spline with unit weights describes the same function as the nonrational spline defined by the same knots and control points.

Let $p_r(\tau)$ be the r^{th} dimensional component of the spline $\boldsymbol{p}(\tau)$, defined by (1). With reference to $p_r(\tau)$, we define the second centered differences of the control points as:

$$\Delta_2 b^k_r = \frac{b^{k+1}_r - b^k_r}{g_{k+1} - g_k} - \frac{b^k_r - b^{k-1}_r}{g_k - g_{k-1}}$$

with $\boldsymbol{g} = [g^0 \dots g^m]$ the Greville abscissae, calculated by the following averaging formula [27]:

$$g^k = \sum_{i=k+1}^{k+d} u^i/d. \tag{2}$$

According to the algorithm proposed in [16], [17], the envelope of $p_r(\tau)$ is delineated by the two polylines such that $\underline{e}_r(\tau) \le p_r(\tau) \le \overline{e}_r(\tau)$. Given $g^k \in [u_l, u_{l+1}]$, the breakpoints defining these polylines are determined by the relations:

$$\underline{e}^k_r = b^k_r + \sum_{i=l-d}^{l} \min\left(0, \Delta_2 b^i_r\right) \gamma^{k,i}(g^k)$$

$$\overline{e}^k_r = b^k_r + \sum_{i=l-d}^{l} \max\left(0, \Delta_2 b^i_r\right) \gamma^{k,i}(g^k)$$

$$\gamma^{k,i}(g^k) \triangleq \begin{cases} \sum_{j=l-d}^{i} (g^i - g^j) N^j_{d,\boldsymbol{u}}(g^k), & i \le k \\ \sum_{j=i}^{l} (g^j - g^i) N^j_{d,\boldsymbol{u}}(g^k), & i \ge k+1 \end{cases}, \tag{3}$$

with the function $\gamma^{k,i}$ non-negative and convex on the l^{th} knot span, and $\gamma^{k,i}(g^k) > 0$ if and only if $i \in [l-d, l]$.

The application of (3) to the n components of $\boldsymbol{p}(\tau)$ acquires the geometric meaning of bounding boxes mentioned in Section II, of which we can provide the following formal definition:

$$\mathcal{S}^k = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid \underline{e}^k_r \le x_r - b^k_r \le \overline{e}^k_r, \ \forall r = 1, \dots, n \right\}. \tag{4}$$

B. NURBS Envelope With Few Pieces

The methodology we propose is based on a new definition of the polygonal envelope of NURBS, which is explained here.

It is easy to confirm that, in the case of rational splines, equations based only on the control polygon do not ensure that the curve lies within the calculated envelope. An example of this is illustrated in Fig. 1(b) (grey area). The main idea for extending these formulations to NURBS is to apply (3) to the two components of the nonrational spline $\boldsymbol{p}_\omega(\tau)$, represented in homogeneous coordinates. $\boldsymbol{p}_\omega(\tau)$ is obtained by applying the inverse perspective map $\mathcal{H}^{-1}$ to the control points and weights of $\boldsymbol{p}(\tau)$. In our case, it is always $w^k > 0$, so the perspective map function can be defined simply as [28]:

$$\mathcal{H}(\boldsymbol{b_\omega}^k) \triangleq \boldsymbol{b}^k, \quad \text{with} \begin{cases} \boldsymbol{b}^k = [b^k_1 \dots b^k_n] \\ \boldsymbol{b_\omega}^k = [\omega^k b^k_1 \dots \omega^k b^k_n \omega^k] \end{cases},$$

In this way, we can define bounding rectangles represented in the homogeneous coordinates, to which the perspective map is applied to compute the breakpoints of the desired envelope. A 3-D graphical representation of this operation is reported in Fig. 1(a). Adopting this approach and denoting for simplicity the product $\omega^k b^k_r$ by $b^k_{\omega,r}$, the breakpoints of the upper and lower polylines delimiting the envelope can be expressed as:

$$\overline{e}^k_{\omega,r} = \begin{cases} \max(b^0_r, b^1_r), & k = 0 \\ \max(b^{m-1}_r, b^m_r), & k = m \\ \frac{b^k_{\omega,r} + \sum_{i=l-d}^{l} \max\left(0, \Delta_2 b^i_{\omega,r}\right) \gamma^{k,i}(g^k)}{\omega^k + \sum_{i=l-d}^{l} \min(0, \Delta_2 \omega^i) \gamma^{k,i}(g^k)}, & \text{otherwise} \end{cases}$$

$$\underline{e}^k_{\omega,r} = \begin{cases} \min(b^0_r, b^1_r), & k = 0 \\ \min(b^{m-1}_r, b^m_r), & k = m \\ \frac{b^k_{\omega,r} + \sum_{i=l-d}^{l} \min\left(0, \Delta_2 b^i_{\omega,r}\right) \gamma^{k,i}(g^k)}{\omega^k + \sum_{i=l-d}^{l} \max(0, \Delta_2 \omega^i) \gamma^{k,i}(g^k)}, & \text{otherwise} \end{cases} \tag{5}$$

Subsequently, the envelope defined by the upper and lower polyline $\overline{e}_{\omega,r}(\tau)$ and $\underline{e}_{\omega,r}(\tau)$, obtained by the breakpoints (5), can be easily thinned by refining the knots (e.g. by the bisection method). An example can be found in in Fig. 1(c).

It is important to note that while the proposed approach typically results in a wider envelope compared to the one based only on the control polygon, the resulting polylines do not guarantee the containment of $p_r(\tau)$ across the entire domain.

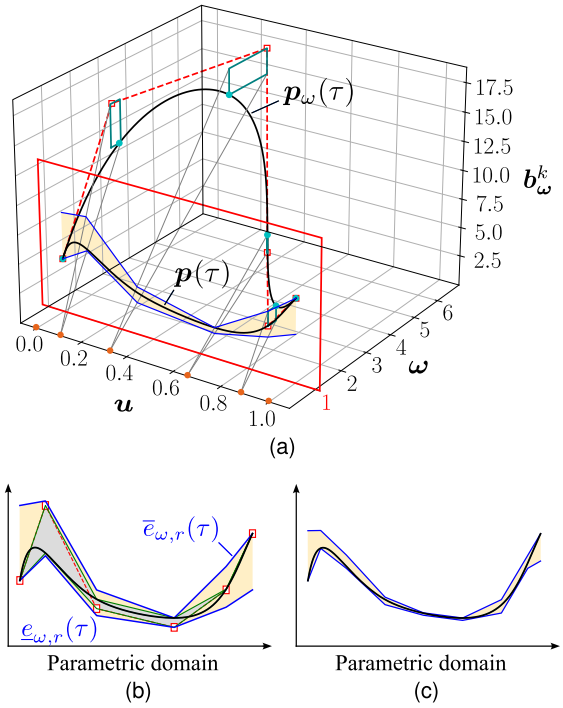

Fig. 1. NURBS envelope strategy. (a) Spatial representation of our approach. (b) Comparison between the proposed algorithm (yellow area) with one based only on the control polygon (grey area). The spline data are: $\boldsymbol{b} = [[4.5][8.5][3][2][4][7]]$, $\boldsymbol{w} = [126311]$, $\boldsymbol{u} = [00001/32/31111]$. (c) Envelope refinement by inserting new knots using the bisection method.

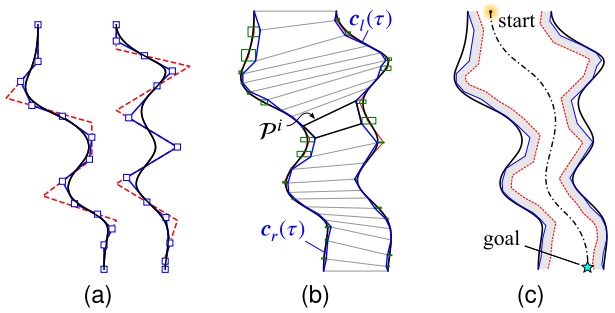

Fig. 2. Planar channel generation. (a) New control polygons after making the splines compatible. (b) Channel obtained after knot refinement, with new knots placed using the bisection method. (c) Channel narrowing.

Specifically, while the condition $\underline{e}_{\omega,r}(g^k) \le p_r(g^k) \le \overline{e}_{\omega,r}(g^k)$ is guaranteed $\forall k = 0, \dots, m$, the spline may violate the polylines between one Greville abscissa and the next. Potential violations can be evaluated numerically, for instance, by verifying the simple conditions:

$$\max_{i=0}^{N} \left(p_r(\tau^i) - \overline{e}_{\omega,r}(\tau^i) \right) \le 0$$
$$\max_{i=0}^{N} \left(\underline{e}_{\omega,r}(\tau^i) - p_r(\tau^i) \right) \le 0 \qquad (6)$$

with $\boldsymbol{\tau} = [\tau^0 \dots \tau^N]$ equispaced mesh in the parametric domain. In the case conditions (6) are not satisfied, one approach is to introduce a tolerance that is added and subtracted from breakpoints $\overline{e}^k_{\omega,r}$ and $\underline{e}^k_{\omega,r}$ respectively. This guarantees the absence of violations by iteratively refining the knots until convergence is attained.

C. Polygonal Channel Definition

Based on the previously detailed NURBS envelope, a general method for approximating channels defined by rational spline boundaries is provided here. According to Fig. 2, this approach can be summarised in the following three steps:

1) Starting from the two rational splines, they are made compatible with knot refinement (Fig. 2(a)). This operation is necessary to couple the NURBS control points to the same Greville abscissa.
2) The SLEFEs for the two splines are calculated using the relations (5). With reference to Fig. 2(b), the polylines defining the channel, $\boldsymbol{c}_l(\tau)$ and $\boldsymbol{c}_r(\tau)$ to the left and right of the channel direction, are the polylines that encloses the channel. Polygonal channel pieces $\mathcal{P}^i$, $i = 0, \dots, N$, are defined by connecting the coupled breakpoints.
3) Finally, with reference to Fig. 2(c), to ensure a safe distance from the channel we can apply an offset algorithm to the polylines $\boldsymbol{c}_l(\tau)$ and $\boldsymbol{c}_r(\tau)$.

It should be noted that the channel approximation can easily be improved by simultaneous knot refinement of the two boundary splines. A global approach for selecting new knots to be inserted is the bisection method; alternatively, focused knot insertions allow localised refinement. Progressive refinement also guarantees that the resulting polylines are non-intersecting and that the initial channel is approximated within the desired tolerance.

D. Initial State and Correspondence

The presented method is based on a strict correspondence of each of the Greville abscissae of the NURBS describing the geometric path with a specific channel piece. Each Greville abscissa g^k, computed from the knots $\boldsymbol{u}$ using relation (2), is related to a bounding rectangle $\mathcal{S}^k$, which is confined within the selected channel piece $\mathcal{P}^i$. We define this correspondence with $\mathcal{S}^k \subset \mathcal{P}^i$.

Solvers for nonlinear optimisation problems typically require a feasible initial solution. Searching for a non-optimal solution to the channel problem is generally not arduous, and several methods can be used for this purpose. However, it must be considered that this initial solution already establishes the aforementioned correspondence, significantly impacting the final result. For these reasons, although it is actually freely settable by the user, we provide an automated procedure for defining the initial state in Algorithm 1, where by initial state we are referring to a generic set of knots, control points, weights and correspondences that define a feasible and non-optimal solution to the problem.

Algorithm 1: Initial State definition.

$\boldsymbol{c}^k \leftarrow \boldsymbol{Mean}(\boldsymbol{c}_l^k\,,\ \boldsymbol{c}_r^k)\,, \quad \textbf{for} k = 0, \dots, N$
$\Delta l^k \leftarrow \|\boldsymbol{c}^k - \boldsymbol{c}^{k-1}\|\,, \quad \textbf{for} k = 1, \dots, N$
$\Delta l_{max} \leftarrow \boldsymbol{Max}(\Delta l^k | k = 1, \dots, N)$
$\boldsymbol{b} \leftarrow [\boldsymbol{c}^0]$
for $k = 1, \dots, N$ **do**
$n^k \leftarrow \boldsymbol{Max}\left(1, \boldsymbol{Min}(n_c, \boldsymbol{Ceil}(\Delta l^k / \Delta l_{max}) n_c + 1)\right)$ (7)
  for $j = 1, \dots, n^k$ **do**
    $\phi \leftarrow (j+1)/(n^k+1)$
    $\boldsymbol{b} \leftarrow \boldsymbol{Append}(\boldsymbol{b}, \boldsymbol{c}^{k-1} + \phi(\boldsymbol{c}^k - \boldsymbol{c}^{k-1}))$
  end for
end for
$\boldsymbol{b} \leftarrow \boldsymbol{Append}(\boldsymbol{b}, \boldsymbol{c}^N)$
$\boldsymbol{\omega} \leftarrow \mathbf{1}$
$\boldsymbol{u}^* \leftarrow \sum_{k=0}^{m-1} \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^k\|$
$\boldsymbol{u} \leftarrow \boldsymbol{Averaging}(d\,,\ \boldsymbol{u}^*)$ ▷from [28, p. 365]
return $\boldsymbol{b}\,,\ \boldsymbol{\omega}\,,\ \boldsymbol{u}$

The main idea of Algorithm 1 is to place the initial control points on the channel centerline. The integer parameter n_c allows the user to choose how many control points have to be used. Specifically, according to relation (7), a maximum of n_c equally spaced control points and a minimum of 1 are allocated for each channel piece. The weights are initially set all unitary, which is equivalent to start with a nonrational spline. The knots are finally determined by associating each control point with a parametric value using the chord length method and then averaging these values. To minimise the solution's cardinality, the parameter n_c should be as low as possible. To this end, Algorithm 1 can be iterated by increasing the value of n_c until a feasible solution is found. With reference to Fig. 3, this adaptive strategy is guaranteed to yield a feasible non-optimal solution since the SLEFE converges to the centerline under refinement of the knot sequence. This property is a direct consequence of the knots refinement of 1-D NURBS on its envelope (already observed in Fig. 1(c)), which results in smaller bounding rectangles $\mathcal{S}^k$.

E. Constraints Definition

1) Weights Positivity: To avoid divisions by zero, it is not particularly limiting to constrain the upper and lower envelope of the weights component of the optimal spline expressed in homogeneous coordinates to strictly positive values. This condition is equivalent to setting the following $m - 1$ inequality

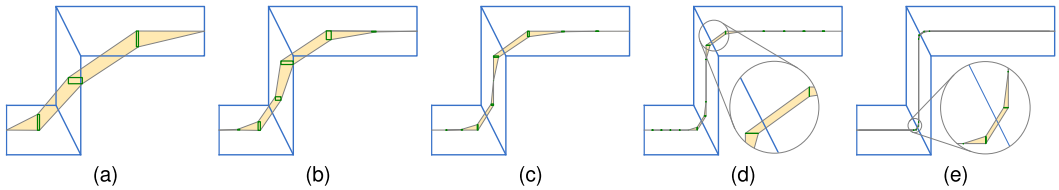

Fig. 3. Initial state identification of a simple channel using Algorithm 1 for different values of n_c. (a) $n_c = 1$. (b) $n_c = 2$. (c) $n_c = 3$. (d) $n_c = 6$. (e) $n_c = 24$.

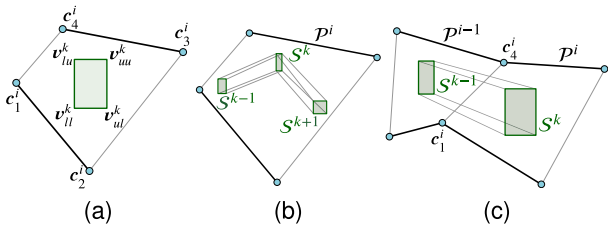

Fig. 4. Non-interference constraints. (a) Channel piece $\mathcal{P}^i$ and bounding rectangle $\mathcal{S}^k$. (b) Each bounding rectangle is confined within its associated channel piece. (c) The channel breakpoints are constrained outside the convex hull of the bounding rectangle between one channel piece and the next.

constraints:

$$\omega^k + \sum_{i=l-d}^{l} \min\left(0, \Delta_2\omega^i\right)\gamma^{k,i}(g^k) > 0 \;\; k = 1, \ldots, m-1.$$

2) Non-Interference Constraints: By non-interference constraints, we refer to conditions that ensure the optimal path stays within the polygonal channel. These relations can be defined as geometric constraints between the bounding rectangles $\mathcal{S}^k$ and the polylines $\boldsymbol{c}_l(\tau)$ and $\boldsymbol{c}_r(\tau)$. This assertion is a straightforward extension of what was stated in [20] for nonrational splines, indicating that each point of the spline calculated for a value in the k^{th} span of the Greville abscissae vector lies within the box defined by the convex combination of the k^{th} bounding rectangle and the next.

For simplicity, we refer to the vertices of the bounding rectangle $\mathcal{S}^k$ with $\boldsymbol{v}_*^k$, where the subscripts $* = ll, lu, ul, uu$ is defined as shown in Fig. 4(a). The vertices of the channel piece $\mathcal{P}^i$ are instead denoted by $\boldsymbol{c}_j^i$, with $j = 1, \ldots, 4$, according to counterclockwise order.

Firstly, with reference to Fig. 4(b), we force $\mathcal{S}^k$ within the associated channel piece $\mathcal{P}^i$. This conditions result in a $16(m+1)$ inequality constraints:

$$\det\begin{bmatrix}\boldsymbol{c}_j^i \boldsymbol{c}_{(j+1)\bmod 4}^i \\ \boldsymbol{c}_j^i \boldsymbol{v}_*^k\end{bmatrix}^T \geq 0 \qquad j = 1, \ldots, 4$$

$\forall k: \; \mathcal{S}^k \subset \mathcal{P}^i$. Similarly, the breakpoints of polylines $\boldsymbol{c}_l(\tau)$ and $\boldsymbol{c}_r(\tau)$ must lie outside the SLEFE of the optimal curve (Fig. 4(c)), resulting in the following $4(N-2)$ inequality constraints:

$$\text{pow}(-1, j)\det\begin{bmatrix}\boldsymbol{v}_*^{k-1}\boldsymbol{v}_*^k \\ \boldsymbol{v}_*^{k-1}\boldsymbol{c}_j^i\end{bmatrix}^T \geq 0 \quad j = 1, 4$$

$$\forall k: \; (\mathcal{S}^k \subset \mathcal{P}^i) \wedge (\mathcal{S}^{k-1} \subset \mathcal{P}^{i-1}).$$

3) Waypoints and Orientation Interpolations: The interpolation of start and target points can easily be forced by setting the first and last control points using bounds. In the case of different waypoints, the equality constraint can be set using the definition (1) once the parametric value corresponding to the interpolated point is defined. Similarly, orientation constraints can be set on the derivatives of (1). In this respect, refer to [28] for formulations in explicit form.

F. Minimum Variation Objective Function

Considering the unknown vector:

$$\boldsymbol{x} = [b_{\omega,x}^0 \ldots b_{\omega,x}^m b_{\omega,y}^0 \ldots b_{\omega,y}^m \omega^0 \ldots \omega^m] \in \mathbb{R}^{3m+3},$$

we adopt the objective function proposed in [5], extended to the representation of NURBS in homogeneous coordinates:

$$f(\boldsymbol{x}) = \sum_{k=1}^{m-1} (\Delta_2 \boldsymbol{b}_{\boldsymbol{\omega}}{}^k)^T (\Delta_2 \boldsymbol{b}_{\boldsymbol{\omega}}{}^k) = \boldsymbol{x}^T \bar{\boldsymbol{P}} \boldsymbol{x} \tag{8}$$

with $\bar{\boldsymbol{P}} = \begin{bmatrix}\boldsymbol{P} & & 0 \\ & \boldsymbol{P} & \\ 0 & & \boldsymbol{P}\end{bmatrix} \in \mathbb{R}^{(3m+3)\times(3m+3)}$ where submatrix $\boldsymbol{P} \in \mathbb{R}^{(m+1)\times(m+1)}$ is pentadiagonal, symmetrical and positive semidefinite. The proof and coefficients computation are summarised in Appendix.

Minimisation of the function (8) keeps the optimal spline as close as possible to a straight line. In general, there is a tendency to generate small segments with sudden changes of direction (high curvature) and almost straight path segments. However, Algorithm 1 defines a knots sequence that ensures low gradients in the Greville abscissa distribution. In [19], the authors show that this feature helps to avoid spikes in the second geometric derivative of the spline, allowing the curvature distribution to be adopted as a qualitative metric.

G. Problem Formulation

In the context of the previous sections, the NLP (NonLinear Programming) problem can be formulated as follows:

$$\begin{aligned} &\text{minimise} && \text{Equation } (8)\,, \\ &\text{subject to} && \boldsymbol{x_l} \leq \boldsymbol{x} \leq \boldsymbol{x_u}\,, && \triangleright \text{ Variable boundaries} \\ & && \boldsymbol{q}_l \leq \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{q}_u\,, && \triangleright \text{ Section III-E} \end{aligned}$$

where $\boldsymbol{g}(\boldsymbol{x})$ is a vector function summarising the constraint relations given in Section III-E. We refer to the optimal solution with $\boldsymbol{x}_{min} = \arg\min f(\boldsymbol{x})$. It follow that the minimum value of the objective function is then defined by $f(\boldsymbol{x}_{min})$. Note that when the weights are unitary, the optimal curve is defined in the domain of nonrational splines. In this case, we refer to the minimum value of the objective function with $f(\boldsymbol{x}_{min})|_{\boldsymbol{\omega}=\mathbf{1}}$. In view of this, we can express the percentage gain in terms of the objective function minimisation with the parameter

$$\eta = \frac{f(\boldsymbol{x}_{min})|_{\boldsymbol{\omega}=\mathbf{1}} - f(\boldsymbol{x}_{min})}{f(\boldsymbol{x}_{min})|_{\boldsymbol{\omega}=\mathbf{1}}} \;\%. \tag{9}$$

It is important to note that, owing to potential violations in the envelopes generated for the two spatial dimensions (as detailed in Section III-B), a check for non-violation of the boundaries

TABLE I
RESULTS OF NUMERICAL BENCHMARKS

Case	d	m	CPU time[a] (ms)	Curve length (m)	$f(\boldsymbol{x}_{min})$	η (%)	$\boldsymbol{b}_{\boldsymbol{\omega}}^{0}$	$\boldsymbol{b}_{\boldsymbol{\omega}}^{m}$	ψ_s	ψ_t	ψ_s'	ψ_t'
Fig. 5c	5	11	366	4.941	1.315	13.75	[0 0.5 1]	[4 2.5 1]	[1.5 0]	[1.5 0]	[0.5 0]	[0.5 0]
Fig. 7a	3	4	54	5.423	0.563	72.52	[0 3.1 1]	[3.1 0 1]	[1 0]	[0 −1]	–	–
Fig. 7b	3	6	116	5.031	0.3	25.35	[0 3.1 1]	[3.1 0 1]	[1 0]	[0 −1]	–	–
Fig. 7c	3	10	386	5.057	0.167	24.92	[0 3.1 1]	[3.1 0 1]	[1 0]	[0 −1]	–	–
Fig. 8b	2	17	1572	29.798	0.926	59.52	[0 12 1]	[17 9 1]	[1 0]	[1 0]	–	–
Fig. 8b	3	17	4148	30.753	0.956	64.12	[0 12 1]	[17 9 1]	[1 0]	[1 0]	–	–
Fig. 8b	4	17	1658	31.625	1.133	71.45	[0 12 1]	[17 9 1]	[1 0]	[1 0]	–	–
Fig. 8c	5	17	1694	31.85	1.317	67.17	[0 12 1]	[17 9 1]	[1 0]	[1 0]	–	–
Fig. 9	3	21	7180	98.30	3.064	89.20	[0 0 1]	[0 0 1]	[0.5 0]	[0.5 0]	–	–

[a] The operations have been computed on an Intel® Core™ i7-13700HX CPU, 5.000 GHz, and 32 GB of RAM. Implementation in Python and Cython languages, using SLSQP solver from SciPy library.

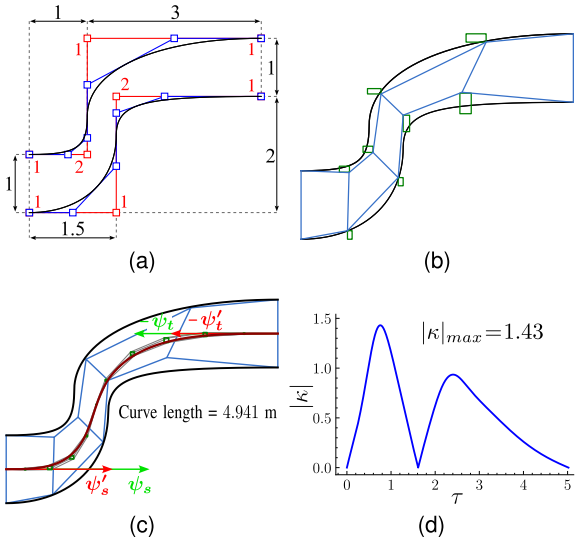

Fig. 5. A short channel numerical benchmark. (a) Rational splines delimiting the channel. Red lines: initial control polygons (weights reported in the label next to corresponding control point); blue lines: new control polygons obtained after refinement.(b) Polygonal channel obtained from algorithm detailed in Section III-C. (c) Solution obtained using quintic NURBS. (d) Curvature distribution over the solution of Fig. 5(c).

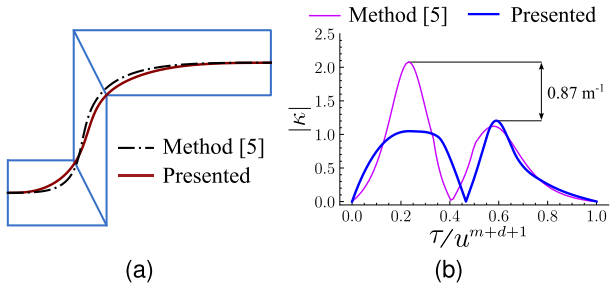

Fig. 6. A short polygonal channel benchmark, taken from [5]. The feasible non-optimal solution defining the initial state is obtained by the automatic approach of Algorithm 1, iterated up to $n_c = 3$. (a) Solution of the presented method compared with the approach proposed in [5]. (b) Curvature distribution over the curve for the two methods.

must be performed once the optimal solution is obtained. If conditions (6) are not satisfied, the algorithm can be integrated into an iterative loop using a tolerance parameter; however, in the numerous conducted simulations, a single iteration proved to be sufficient. Indeed, it was empirically observed that the minimization of the objective function (8) inherently tends to avoid such violations.

IV. NUMERICAL EXAMPLES

In this section we present some numerical tests of the proposed method. Results and input data of all numerical examples are summarised in Table I, where the gain obtained with respect to the solution based on nonrational splines is expressed in terms of the parameter η, defined by (9).

A. A Short Channel

This example is adapted from [5], where the boundary polylines are used as NURBS control points defining our channel. Channel boundaries, reported in Fig. 5(a), are refined by inserting new knots using the bisection method. The methodology detailed in Section III-C leads to the definition of the polygonal channel shown in Fig. 5(b). Initial knots and control points are determined using the method detailed in Section III-D, adopting $d = 5$ and $n_c = 2$. The following interpolation constraints are added to the optimisation problem:

$$\boldsymbol{p}(0) = \boldsymbol{b}_{\boldsymbol{\omega}}{}^{0} \qquad \boldsymbol{p}(u^{m+d+1}) = \boldsymbol{b}_{\boldsymbol{\omega}}{}^{m}$$
$$\delta\boldsymbol{p}(0)/\delta\tau = \psi_s \qquad \delta\boldsymbol{p}(u^{m+d+1})/\delta\tau = \psi_t$$
$$\delta^2\boldsymbol{p}(0)/\delta\tau^2 = \psi_s' \qquad \delta^2\boldsymbol{p}(u^{m+d+1})/\delta\tau^2 = \psi_t'$$

As reported in Table I, comparison with the solution using nonrational splines denotes a reduction in the objective function optimal value.

If the channel is bounded by polylines, these functions can be directly employed to define the constraints of the problem. In this regard, Fig. 6(a) reports the solution obtained with our method from the example provided in [5], compared with the approach based on nonrational splines proposed by Jung and Tsiotras in the same reference. In both scenarios, the optimal solutions are characterized by quartic splines, and the objective function (8) is adopted during the minimisation (weights set to unity in the solution proposed in [5]). As already detailed in Section III-F, under specific conditions, this objective function can be associated with the distributed curvature over the optimal curve. Hence, Fig. 6(b) illustrates the curvature distribution along the curve for both cases, revealing lower values for the proposed method.

B. Quarter Circle

Let the radius of two circles be $r_i = 3$ m and $r_e = 5$ m with their centres at the origin of the xy plane. These curves can be easily represented exactly by quadric NURBS with a nine-point square control polygon [28]. Focusing on the 90° arcs in the first quadrant, the channel is delimited by the two arcs.

With reference to Table I, Fig. 7 shows some solutions of our method for a progressive channel refinement. In all the illustrated scenarios, the desired solution is a cubic NURBS, and knots are determined using Algorithm 1 with $n_c = 1$. A progressive

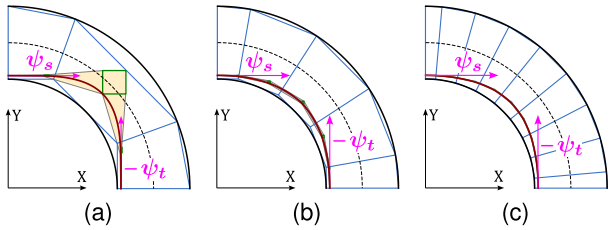

Fig. 7. Quarter circle benchmark. (a) 3 channel pieces. (b) 5 channel pieces. (c) 9 channel pieces.

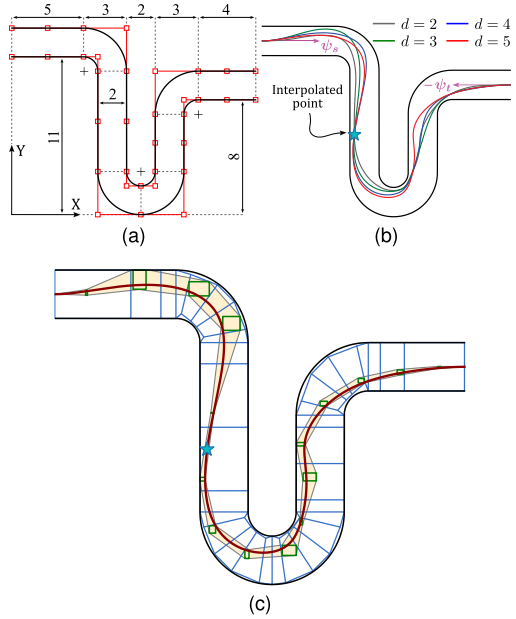

Fig. 8. Syphon benchamrk. (a) Rational splines delimiting the channel. (b) Comparison of solutions using NURBS of different degrees. (c) Polygonal channel obtained after localised knot insertions, and detail on solution with quintic NURBS. The star marker denotes the interpolated waypoint.

improvement in terms of objective function can be seen as the channel is refined. In all analysed solutions, we have a reduction in the objective function optimal value compared to the solution using a nonrational spline.

C. Syphon

This benchmark proposes a solution to the channel problem shown in Fig. 8(a), consisting of constant-width linear sections with circular blends. The problem, in addition to setting the end first derivatives, also requires the interpolation of the waypoint $[6.35.6]$. Desiring a low number of control points, we deliberately select the initial state. Knots are computed from these first using the chord length method and then applying the averaging algorithm to the resulting breakpoints. The correspondence of the Greville abscissae with the channel pieces can be easily deduced from Fig. 8(c), which details the solution for a quintic NURBS. Fig. 8(b) summarises the solutions for the resulting curve degree from 2 to 5. Based on the results presented in Table I, it is evident that in all the cases examined, there is a significant improvement in the optimal value of the objective function in the NURBS-based solution compared to the solution using nonrational spline.

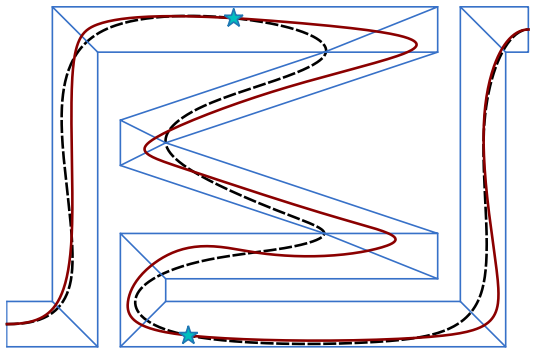

Fig. 9. Kano channel benchmark [26]. Solid line report our solution using NURBS. Dashed line report our solution using nonrational spline. The star marker denotes the interpolated waypoints.

D. "Kano" Polygonal Channel

This example proposes the solution to the polygonal channel benchmark reported in [26]. From Fig. 9, we note that some channel pieces are triangular. This condition may also occur when the boundaries are non-linear, e.g. due to a narrowing of the channel (shown in Fig. 2(c)). In the proposed solution, the two waypoints $[1014.5]$ and $[80.5]$ are also interpolated. Fig. 9 shows the optimal curves of the proposed algorithm using both NURBS and nonrational splines (i.e. for $\boldsymbol{\omega} = \mathbf{1}$). From the results shown in Table I, it follows that the use of rational splines leads to a significant reduction in the objective function value after minimisation.

V. Conclusion

In this letter, we introduce a novel approach based on rational splines for planning an optimal path within a planar channel. The main results can be summarized as follows:

- The proposed method provides a generalisation of the planar channel problem to rational splines. Through different numerical tests, we show that formulating the problem using NURBS expands the domain of feasible solutions, resulting in a significant reduction in the optimal value of the objective function.
- Our approach allows users to directly import channel geometry from standard CAD software platforms and refine the resulting polygonal channel. As shown in the example provided in Section IV-B, this procedure defines a meshing operation that constitutes a secondary method to achieve convergence of the algorithm.
- Although the method detailed in Section II does not guarantee the generation of a SLEFE that always contains the curve, the minimisation of the objective function (8) needs algorithm iterations only in exceptional cases. Consequently, although a check is always necessary, no iterations were needed in all the conducted tests.

Appendix
Submatrix P Definition

We define the following parameters:

$$\Delta g^k = g^k - g^{k-1} > 0, \qquad k = 1, \dots, m$$

$$K^k = \Delta g^{k+1} \Delta g^k, \qquad k = 1, \dots, m-1$$

$$\boldsymbol{P} = \begin{bmatrix} A^0 & -B^0 & C^0 & & & & & 0 \\ & & \ddots & & & & & \\ & & C^{k-2} & -B^{k-1} & A^k & -B^k & C^k & \\ & & & & & & \ddots & \\ 0 & & & & & & C^{m-2} & -B^{m-1} & A^m \end{bmatrix}$$

$$A^k = \begin{cases} \frac{\text{pow}(\Delta g^2,2)}{\text{pow}(K^1,2)}, & k=0 \\ \frac{\text{pow}(\Delta g^1+\Delta g^2,2)}{\text{pow}(K^1,2)} + \frac{\text{pow}(\Delta g^3,2)}{\text{pow}(K^2,2)}, & k=1 \\ \frac{\text{pow}(\Delta g^{m-2},2)}{\text{pow}(K^{m-2},2)} + \frac{\text{pow}(\Delta g^{m-1}+\Delta g^m,2)}{\text{pow}(K^{m-1},2)}, & k=m-1 \\ \frac{\text{pow}(\Delta g^{m-1},2)}{\text{pow}(K^{m-1},2)}, & k=m \\ \frac{\text{pow}(\Delta g^{k-1},2)}{\text{pow}(K^{k-1},2)} + \frac{\text{pow}(\Delta g^k+\Delta g^{k+1},2)}{\text{pow}(K^k,2)} & \\ \quad + \frac{\text{pow}(\Delta g^{k+2},2)}{\text{pow}(K^{k+1},2)}, & \text{otherwise} \end{cases}$$

$$B^k = \begin{cases} \frac{(\Delta g^1+\Delta g^2)\Delta g^2}{\text{pow}(K^1,2)}, & k=0 \\ \frac{\Delta g^{m-1}(\Delta g^{m-1}+\Delta g^m)}{\text{pow}(K^{m-1},2)}, & k=m-1 \\ \frac{\Delta g^k(\Delta g^k+\Delta g^{k+1})}{\text{pow}(K^k,2)} & \\ \quad + \frac{(\Delta g^{k+1}+\Delta g^{k+2})\Delta g^{k+2}}{\text{pow}(K^{k+1},2)}, & \text{otherwise} \end{cases}$$

$$C^k = \frac{\Delta g^{k+1}\Delta g^{k+2}}{\text{pow}(K^{k+1},2)} > 0, \qquad k = 0,\ldots,m-2$$

The submatrix $\boldsymbol{P}$ in matrix $\bar{\boldsymbol{P}}$ in (8) is defined by the symmetrical and pentadiagonal submatrices unnumbered equation shown at the top of the page:

REFERENCES

[1] T. Berglund, H. Jonsson, and I. Soderkvist, "An obstacle-avoiding minimum variation B-spline problem," in *Proc. IEEE Int. Conf. Geometric Model. Graph.*, 2003, pp. 156–161, doi: 10.1109/GMAG.2003.1219681.

[2] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles," *IEEE Trans. Automat. Sci. Eng.*, vol. 7, no. 1, pp. 167–172, Jan. 2010, doi: 10.1109/TASE.2009.2015886.

[3] D. Jung, "Hierarchical path planning and control of a small fixed-wing UAV: Theory and experimental validation," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, USA, 2007.

[4] D. Jung, J. Ratti, and P. Tsiotras, "Real-time implementation and validation of a new hierarchical path planning scheme of UAVs via hardware-in-the-loop simulation," *J. Intell. Robot. Syst.*, vol. 54, pp. 163–181, 2009, doi: 10.1007/s10846-008-9255-0.

[5] D. Jung and P. Tsiotras, "On-line path generation for unmanned aerial vehicles using B-spline path templates," *J. Guid., Control, Dyn.*, vol. 36, no. 6, pp. 1642–1653, 2013, doi: 10.2514/1.60780.

[6] E. Selinger and L. Linsen, "Efficient curvature-optimised g2-continuous path generation with guaranteed error bound for 5-axis machining," in *Proc. IEEE 13th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2018, pp. 59–70, doi: 10.5220/0006537400590070.

[7] E. Guilbert, E. Saux, and M. Daniel, "Combining geometrical and mechanical displacements for suppressing intersections in cartographic generalization," WSCG Short Communications, Union Agency-Science Press, Pilsen, Czech Republic, 2004, pp. 87–94.

[8] D. Nguyen, M. Pauley, and B. Jüttler, "Isogeometric segmentation: Construction of auxiliary curves," *Comput.-Aided Des.*, vol. 70, pp. 89–99, 2016, doi: 10.1016/j.cad.2015.06.014.

[9] K. Meng, Y. Jia, H. Yang, F. Niu, Y. Wang, and D. Sun, "Motion planning and robust control for the endovascular navigation of a microrobot," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4557–4566, Jul. 2020, doi: 10.1109/TII.2019.2950052.

[10] T. Weingartshofer, B. Bischof, M. Meiringer, C. Hartl-Nesic, and A. Kugi, "Optimisation-based path planning framework for industrial manufacturing processes with complex continuous paths," *Robot. Comput.- Integr. Manuf.*, vol. 82, 2023, Art. no. 102516, doi: 10.1016/j.rcim.2022.102516.

[11] J. Kim and E. A. Croft, "Online near time-optimal trajectory planning for industrial robots," *Robot. Comput.-Integr. Manuf.*, vol. 58, pp. 158–171, 2019, doi 10. 1016/j.rcim.2019.02.009.

[12] M. Gadaleta, M. Pellicciari, and G. Berselli, "Optimization of the energy consumption of industrial robots for automatic code generation," *Robot. Comput.-Integr. Manuf.*, vol. 57, pp. 452–464, 2019, doi: 10.1016/j.rcim.2018.12.020.

[13] J. Huang, P. Hu, K. Wu, and M. Zeng, "Optimal time-jerk trajectory planning for industrial robots," *Mech. Mach. Theory*, vol. 121, pp. 530–544, 2018, doi: 10.1016/j.mechmachtheory.2017.11.006.

[14] F. Rubio, C. Llopis-Albert, F. Valero, and J. L. Suñer, "Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory," *Robot. Auton. Syst.*, vol. 86, pp. 106–112, 2016, doi: 10.1016/j.robot.2016.09.008.

[15] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019, doi: 10.1109/LRA.2019.2927938.

[16] D. Lutterkort and J. Peters, "Tight linear envelopes for splines," *Numerische Mathematik*, vol. 89, pp. 735–748, 2001, doi: 10.1007/s002110100181.

[17] D. Lutterkort, "Envelopes for nonlinear geometry," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, 2000.

[18] D. Lutterkort and J. Peters, "Smooth paths in a polygonal channel," in *Proc. 15th Annu. Symp. Comput. Geometry*, 1999, pp. 316–321.

[19] M. Riboli, M. Jaccard, M. Silvestri, A. Aimi, and C. Malara, "Collision-free and smooth motion planning of dual-arm cartesian robot based on B-spline representation," *Robot. Auton. Syst.*, vol. 170, 2023, Art. no. 104534, doi: 10.1016/j.robot.2023.104534.

[20] D. Lutterkort and J. Peters, "Linear envelopes for uniform B-spline curves," in *Proc. Int. Conf. Curves Surfaces*, 1999, pp. 239–246.

[21] J. Peters and X. Wu, "Sleves for planar spline curves," *Comput. Aided Geometric Des.*, vol. 21, no. 6, pp. 615–635, 2004, doi: 10.1016/j.cagd.2004.04.004.

[22] A. Myles and J. Peters, "Threading splines through 3D channels," *Comput.-Aided Des.*, vol. 37, no. 2, pp. 139–148, 2005, doi: 10.1016/j.cad.2004.04.004.

[23] J. Selinger and L. Linsen, "Efficient curvature-optimised g2-continuous path generation with guaranteed error bound for 3-axis machining," in *Proc. IEEE 15th Int. Conf. Inf. Visualisation*, 2011, pp. 519–527, doi: 10.1109/IV.2011.31.

[24] H. Kano, H. Fujioka, and C. F. Martin, "Optimal smoothing spline with constraints on its derivatives," *SICE J. Control, Meas., Syst. Integration*, vol. 7, no. 2, pp. 104–111, 2014, doi: 10.9746/jcmsi.7.104.

[25] H. Kano and H. Fujioka, "Spline trajectory planning for path with piecewise linear boundaries," in *Proc. 9th EUROSIM Congr. Model. Simul., 57th SIMS Conf. Simul. Model.*, 2016, pp. 439–445, doi: 10.3384/ecp17142439.

[26] H. Kano and H. Fujioka, "Spline trajectory planning for road-like path with piecewise linear boundaries allowing double corner points," *SICE J. Control, Meas., Syst. Integration*, vol. 11, no. 6, pp. 429–437, 2018, doi: 10.9746/jcmsi.11.429.

[27] C. de Boor, "A practical guide to splines," in *Applied Mathematical Sciences*, New York, NY, USA: Springer, 2001.

[28] L. A. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY, USA: Springer-Verlag, 1996.