

# Adaptive Bézier Degree Reduction and Splitting for Computationally Efficient Motion Planning

Ömür Arslan<sup>✉</sup>, Member, IEEE, and Aron Tiemessen

**Abstract**—As a parametric polynomial curve family, Bézier curves are widely used in safe and smooth motion design of intelligent robotic systems from flying drones to autonomous vehicles to robotic manipulators. In such motion planning settings, the critical features of high-order Bézier curves such as curve length, distance-to-collision, maximum curvature/velocity/acceleration are either numerically computed at a high computational cost or inexactly approximated by discrete samples. To address these issues, in this article we present a novel computationally efficient approach for adaptive approximation of high-order Bézier curves by multiple low-order Bézier segments at any desired level of accuracy that is specified in terms of a Bézier metric. Accordingly, we introduce a new Bézier degree reduction method, called *parameterwise matching reduction*, which approximates Bézier curves more accurately compared to the standard least squares and Taylor reduction methods. We also propose a new Bézier metric, called the *maximum control-point distance*, that can be computed analytically, has a strong equivalence relation with other existing Bézier metrics, and defines a geometric relative bound between Bézier curves. We provide extensive numerical evidence to demonstrate the effectiveness of our proposed Bézier approximation approach. As a rule of thumb, based on the degree-one matching reduction error, we conclude that an  $n^{\text{th}}$ -order Bézier curve can be accurately approximated by  $3(n - 1)$  quadratic and  $6(n - 1)$  linear Bézier segments, which is fundamental for Bézier discretization.

**Index Terms**—Bézier curves, path discretization, path smoothing, polynomial trajectory optimization, motion planning.

## I. INTRODUCTION

SAFE and smooth motion planning is essential for many autonomous robots. As a parametric smooth motion representation, polynomial curves find significant applications in safe robot motion design from flying drones [1]–[5] to autonomous vehicles [6]–[9] to robotic manipulators [10]–[13]. Polynomials expressed in different (e.g., monomial, Taylor, and Bernstein) bases offer different useful functional and geometric properties for computationally efficient motion planning. While the monomial (a.k.a. power) basis yields quadratic trajectory optimization objectives [1], polynomial Bézier curves in Bernstein basis

Manuscript received 13 April 2022; accepted 30 May 2022. Date of publication 27 July 2022; date of current version 6 December 2022. This article was recommended for publication by Associate Editor R. Murrieta-Cid and Editor F. Chaumette upon evaluation of the reviewers' comments. (*Corresponding author:* Ömür Arslan.)

The authors are with the Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, and also with Eindhoven AI Systems Institute, 5612 AZ Eindhoven, The Netherlands. (e-mail: o.arslan@tue.nl; a.j.c.tiemessen@student.tue.nl).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TRO.2022.3187296>.

Digital Object Identifier 10.1109/TRO.2022.3187296

1552-3098 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

TABLE I  
COMPUTATION OF BÉZIER CURVE FEATURES IN MOTION PLANNING

Bezier Curve Feature	$n \leq 2$	$n = 3$	$n > 3$
Arc Length	Analytic	Numeric	Numeric
Maximum Velocity	Analytic	Analytic	Numeric
Maximum Curvature	Analytic	Numeric	Numeric
Distance-to-Point	Analytic	Numeric	Numeric
Distance-to-Line-Segment	Analytic	Numeric	Numeric

have useful convexity and interpolation properties [4]: a Bézier curve is contained in the convex hull of its control points (i.e., parameters), and it smoothly interpolates between the first and last control point. A well known challenge of motion planning with polynomial and so Bézier curves is that the computational complexity increases with increasing curve degree [6]. Because critical curve features such as curve length, distance-to-collision, and maximum curvature/velocity/acceleration can be analytically determined only for low-order (e.g., linear and quadratic) polynomial curves and are numerically computed or inexactly approximated using discrete samples for higher order polynomials, as summarized in Table I.

In this article, we propose a new computationally efficient approach for adaptive approximation of high-order Bézier curves by multiple low-order Bézier segments at any desired level of accuracy specified in terms of a Bézier metric, as illustrated in Fig. 1. Our approach is based on an unexplored functional property of Bézier curves in motion planning: distance between Bézier curves can be measured analytically in terms of control points. Accordingly, we introduce a new analytic Bézier metric, called the *maximum control-point distance*, which can be used to geometrically bound Bézier curves with respect to each other, and defines tight upper bounds on other existing Bézier metrics. We also propose a new Bézier degree reduction method, called *parameterwise matching reduction*, which allows preserving certain curve points (e.g., end points) while performing degree reduction. Based on the degree-one parameterwise matching reduction error, we conclude that an  $n^{\text{th}}$ -order Bézier curve can be accurately approximated by  $3(n - 1)$  quadratic and  $6(n - 1)$  linear Bézier segments, which is a fundamental rule of thumb for Bézier discretization. In numerical simulations, we demonstrate the effectiveness of approximating high-order Bézier curves by linear and quadratic Bézier segments for fast and accurate computation of common curve features used in motion planning.

## A. Motivation and Related Literature

Autonomous robots and people interacting with them enjoy smooth motion in practice: jerky robot motion does not only

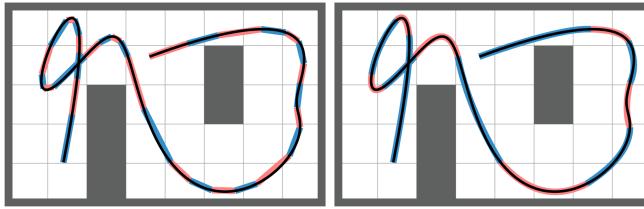


Fig. 1. Adaptive approximation of an 8<sup>th</sup>-order Bézier curve (black line), e.g., for computationally efficient, safe, and smooth drone navigation around obstacles, by multiple (left) linear and (right) quadratic Bézier segments (red and blue patches) whose length, maximum velocity/acceleration/curvature, and distance to obstacles (dark gray) can be computed analytically. The Bézier degree reduction and splitting is automatically performed by uniform matching reduction and binary search for a maximum control-point distance of 0.1 units.

cause mechanical and electrical failures and malfunctions, but also causes discomfort for the user. Most existing smooth motion planning methods follow a two-step approach: first find a piecewise linear path for a simplified version of the system to achieve a simplified version of a given task; and then perform path smoothing as postprocessing to satisfy the actual task and system requirements [14]. The first step, piecewise linear motion planning, is well established with many computationally effective (search- and sampling-based) planning algorithms for the fully actuated kinematic robot model [15]. The second step, path smoothing that aims to convert a piecewise linear reference plan into a smooth dynamically feasible trajectory satisfying both system and task constraints, is an active research topic, especially for real-time operation requirements. Due to their compact parametric form and functional properties, polynomial curves have recently received significant attention with promising potentials for computationally efficient path smoothing, especially for differential flat systems [16] such as cars [7], [8], quadrotors [1]–[3], and fixed-wing aircrafts [17], to name a few, whose control inputs can be expressed as a function of *flat system outputs* (represented by polynomials) and their derivatives. For example, while polynomials of degree 3–5 are often used for autonomous vehicles, polynomials of degree 5–10 are required for quadrotors. The major reason for the use of relatively low-order polynomials in practice is that the computational cost of planning with polynomials increases with increasing degree of polynomials [6], [14]. Our proposed approach enables handling high-order polynomials efficiently by approximating them with multiple low-order polynomial segments.

Convex optimization plays a key role in polynomial path smoothing. In polynomial trajectory optimization, the standard optimization objectives of total squared velocity, acceleration, jerk, and snap (i.e., the first, second, third, and fourth time derivatives of the position) of a robotic system can be written as a quadratic objective function of polynomial parameters [1], [17]. In order to take the full advantage of quadratic programming, the system and task constraints are often represented as linear or quadratic inequalities. For example, a piecewise linear reference plan can be used to construct a convex safe corridor around the reference plan to represent planning constraints as a collection of convex polytopes [18] or spheres [3]. Accordingly, polynomial

trajectory optimization is often formulated as a quadratic optimization problem, for example, by simply using a polynomial discretization [1], [2], [13]. This naturally raises a question about polynomial discretization: how many sample points along a polynomial are needed for an accurate representation of planning constraints. The existing methods use either manual or heuristic approaches to add extra samples if polynomial discretization fails [1], [2]. In this sense, our results offer a systematic solution for determining a proper discretization of polynomials to model planning constraints at any desired level of accuracy.

In polynomial trajectory optimization, the convexity property of Bézier curves makes them an attractive choice for handling convex system constraints within quadratic programming. Since Bézier curves are contained in the convex hull of their control points, trajectory optimization constraints are often enforced by constraining Bézier control points inside convex constraint sets [4], [19], [20]. This approach is effectively applied for smooth trajectory generation with Bezier curves over safe corridors [18] in various application settings; for example, for drone navigation in unknown environments [4], [5], autonomous driving [7], [20], [21], multirobot coordination [19], [22], and perception-aware navigation [23]. Although it performs reasonably well for low-order Bézier curves in practice, this simple but conservative approach is suboptimal for high-order Bézier curves since the convex hull of Bézier control points significantly overestimates the smallest convex region containing the actual curve, especially for higher order polynomials. On the other hand, exact and fast continuous constraint verification with polynomial curves is possible based on the separation of polynomial extremes [24], the sign change of polynomials [25], and their root existence test based on Sturm's theorem [26], but these methods result in highly complex nonlinear optimization constraints. Our approach for approximating high-order Bézier curves by low-order Bézier segments allows one to use the convexity of low-order Bézier curves in high-order polynomial trajectory optimization in a less conservative way.

Another appealing feature of Bézier curves for smooth robot motion design is that they smoothly interpolate between the first and last control points. This interpolation property is often leveraged for motion planning of nonholonomic systems with boundary conditions; for example, for waypoint smoothing [9], [27], [28] and smooth steering control [29], [30] in autonomous driving [31], [32], and path smoothing in sampling-based motion planning [11], [33]–[35]. Continuous curvature path smoothing with curvature constraints is applied for increasing passenger comfort while ensuring dynamical feasibility in autonomous vehicles for smooth lane change [36] and urban driving [8], [37]–[39]. Although path smoothing with curvature constraints can be performed analytically for low-order Bézier curves [40], the maximum curvature is numerically computed for high-order Bézier curves [21]. Thus, one can use our adaptive Bézier approximation approach to take the analytic advantages of low-order Béziers in path smoothing with high-order Béziers.

As a smooth motion primitive, polynomial curves are also used in search-based and sampling-based smooth motion planning of nonholonomic systems [41], [42] and robotic manipulators [13], [34], [42] as well as their reinforcement learning [12].

A challenge of planning with polynomial motion primitives is finding an informative and computationally efficient local metric for measuring the connectivity and travel cost. A natural travel cost measure is the arc length of polynomials, which can be analytically determined only for linear and quadratic polynomials. Using the proposed Bézier approximation method, one can accurately and efficiently measure the arc length of high-order polynomial curves by dividing them into multiple low-order polynomial segments.

Bézier curves are widely used in computer graphics and computer aided design (CAD) for efficiently representing complex shapes with few parameters in order to reduce the space complexity [43], [44]. Handling complex shapes with minimal space complexity often requires optimal reduction of Bézier curves based on different metrics [45]–[47]. This motivates many alternative approaches for degree reduction of Bézier curves [48] and their approximate conversions [49] (with end point constraints [50]). This article brings such CAD tools to the motion planning literature for reducing the time complexity, with important additions which, we believe, also contribute back to the CAD literature. Finally, smooth polynomial curves also find applications in smooth functional regression of noisy sample data [51], and our adaptive Bézier degree reduction and splitting approach can be applied for computationally efficient local smooth polynomial regression.

### B. Contributions and Organization of the Article

In this article, we present a novel systematic approach for adaptive discretization and approximation of high-order Bézier curves by multiple low-order Bézier curves for computationally efficient smooth motion planning with high-order polynomials. In summary, our main contributions are:

- 1) a new Bézier metric, called the *maximum control-point distance*, that defines an analytic tight upper bound on existing standard Bézier metrics such as the Hausdorff, parameterwise maximum, and Frobenius-norm distances of Bézier polynomials, and enables bounding Bézier curves geometrically with respect to each other;
- 2) a new Bézier degree reduction method, called *parameterwise matching reduction*, that approximates a Bézier curve by a single lower-order Bézier curve more accurately (e.g., by preserving end points) compared to the standard least squares and Taylor reductions;
- 3) a new adaptive Bézier degree reduction and splitting approach for approximating a high-order Bézier curve by multiple low-order Bézier segments at any desired level of accuracy that is specified in terms of a Bézier metric;
- 4) a new rule of thumb for accurately approximating high-order Bézier curves with a fixed finite collection of linear and quadratic Bézier curves.

With extensive numerical simulations, we demonstrate the effectiveness of the newly proposed methods. At a more conceptual level, this article for the first time introduces the use of Bézier metrics and degree reduction methods for local low-order approximation of high-order Bézier curves in order to enable computationally efficient smooth motion planning. To our knowledge, important curve properties (see Table I) used

in polynomial trajectory planning and optimization are often determined using manually selected piecewise linear approximations based on trial-and-error and expert knowledge. Our results mitigate this issue and offer new advanced tools that allow automated adaptive approximation of polynomial curves with linear, quadratic, or higher order Bézier segments.

The rest of the article is organized as follows. In Section II, we provide a background overview of Bézier curves, and the matrix representation, basis transformation and reparametrization of polynomial curves. In Section III, we describe how to measure the distance between Bézier curves and introduce a new Bézier metric. In Section IV, we present how to (approximately) represent a Bézier curve with more or fewer control points via degree elevation and reduction operations, and introduce a new degree reduction method. In Section V, we describe how to approximate high-order Bézier curves by low-order Bézier curves at any desired accuracy level, and present a rule of thumb for accurate Bézier approximations. In Section VI, we present numerical results to demonstrate the role of polynomial degree and the number of Bézier segments on approximation accuracy. In Section VII, we conclude with a summary of our research highlights and future directions.

## II. BÉZIER CURVES

In this section, we first briefly introduce Bézier curves and their important properties, and then continue with the matrix representation, basis transformation, and affine reparameterization of polynomial Bézier, monomial, and Taylor curves.

### A. Characteristic Properties of Bézier Curves

*Definition 1 (Bézier Curve):* In a  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ , a Bézier curve  $B_{p_0, \dots, p_n}(t)$  of degree  $n \in \mathbb{N}$ , associated with control points  $p_0, \dots, p_n \in \mathbb{R}^d$ , is a parametric polynomial curve defined for  $0 \leq t \leq 1$  as<sup>1</sup>

$$B_{p_0, \dots, p_n}(t) := \sum_{i=0}^n b_{i,n}(t) p_i \quad (1)$$

where  $b_{i,n}(t)$  denotes the  $i$ th Bernstein basis polynomial of degree  $n$  that is defined for  $i = 0, 1, \dots, n$  as

$$b_{i,n}(t) := \binom{n}{i} t^i (1-t)^{n-i}. \quad (2)$$

Key characteristics of Bézier and Bernstein polynomials are their recursion, derivative, and convexity properties [43], [44].

*Property 1 (Recursion):* A Bézier curve can be recursively determined as a convex combination of two Bézier curves of one degree lower as

$$B_{p_0, p_1, \dots, p_n}(t) = (1-t)B_{p_0, \dots, p_{n-1}}(t) + tB_{p_1, \dots, p_n}(t) \quad (3)$$

with the base case  $B_{p_0}(t) = p_0$ , which follows from the recursive definition of Bernstein polynomials:

$$b_{i,n}(t) = (1-t)b_{i,n-1}(t) + tb_{i-1,n-1}(t) \quad (4)$$

with base cases  $b_{0,0}(t) = 1$  and  $b_{i,n}(t) = 0$  for  $i < 0$  and  $i > n$ .

<sup>1</sup>The standard definition of Bézier curves is over the unit interval, and they are mathematically well-defined over all reals.

*Property 2 (Derivative):* The derivative of a Bézier curve is another Bézier curve of one degree lower and given by

$$\frac{d}{dt} B_{p_0, p_1, \dots, p_n}(t) = n B_{p_1 - p_0, \dots, p_n - p_{n-1}}(t) \quad (5)$$

since the Bernstein derivatives satisfy

$$\frac{d}{dt} b_{i,n}(t) = n(b_{i-1,n-1}(t) - b_{i,n-1}(t)). \quad (6)$$

*Property 3 (Convexity):* A Bézier curve is contained in the convex hull, denoted by  $\text{conv}$ , of its control points, i.e.,

$$B_{p_0, \dots, p_n}(t) \in \text{conv}(p_0, \dots, p_n) \quad \forall t \in [0, 1] \quad (7)$$

because Bernstein polynomials are nonnegative and sum to one, i.e., for any  $t \in [0, 1]$

$$b_{i,n}(t) \geq 0, \quad \text{and} \quad \sum_{i=0}^n b_{i,n}(t) = 1. \quad (8)$$

*Property 4 (Interpolation):* A Bézier curve smoothly interpolates between its first and last control point, i.e.,

$$B_{p_0, \dots, p_n}(0) = p_0 \quad \text{and} \quad B_{p_0, \dots, p_n}(1) = p_n \quad (9)$$

since Bernstein polynomials smoothly interpolates between

$$(b_{0,n}(0), \dots, b_{n,n}(0)) = (1, 0, \dots, 0) \quad (10a)$$

$$(b_{0,n}(1), \dots, b_{n,n}(1)) = (0, \dots, 0, 1). \quad (10b)$$

## B. Matrix Representation of Polynomial Curves

To effectively handle high-order Bézier curves with a large number of control points, it is convenient to use the matrix representation of Bézier curves in the form of

$$B_{p_0, \dots, p_n}(t) = \mathbf{P}_n b_n(t) \quad (11)$$

based on the control point matrix  $\mathbf{P}_n := [p_0, \dots, p_n] \in \mathbb{R}^{d \times (n+1)}$  and the Bernstein basis vector  $b_n(t) \in \mathbb{R}^{n+1}$  that is defined as

$$b_n(t) := \begin{bmatrix} b_{0,n}(t) \\ b_{1,n}(t) \\ \vdots \\ b_{n,n}(t) \end{bmatrix}. \quad (12)$$

Note that the Bernstein basis polynomials  $b_{0,n}(t), \dots, b_{n,n}(t)$  form a basis of  $n+1$  linearly independent polynomials for polynomials of degree  $n$  [43]. The two other widely used basis functions of the  $n^{\text{th}}$ -order polynomials are the monomial and Taylor basis vectors, respectively, defined as

$$m_n(t) := \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix}, \quad \text{and} \quad \tau_{n,t_o}(t) := \begin{bmatrix} 1 \\ t - t_o \\ \vdots \\ (t - t_o)^n \end{bmatrix} \quad (13)$$

where  $t_o \in \mathbb{R}$  is the Taylor offset term. Accordingly, like Bézier curves, one can define the *monomial* and *Taylor curves*, associated with control points  $\mathbf{Q} = [q_0, \dots, q_n] \in \mathbb{R}^{d \times (n+1)}$  and  $\mathbf{Y}_n = [y_0, \dots, y_n] \in \mathbb{R}^{d \times n+1}$ , respectively, as

$$M_{q_0, \dots, q_n}(t) := \sum_{i=0}^n q_i t^i = \mathbf{Q}_n m_n(t) \quad (14a)$$

$$\Upsilon_{y_0, \dots, y_n}(t, t_o) := \sum_{i=0}^n y_i (t - t_o)^i = \mathbf{Y}_n \tau_{n,t_o}(t). \quad (14b)$$

From their similar forms in (13), one can observe that the monomial and Taylor basis vectors (and so curves) are strongly related, i.e.,

$$\tau_{n,t_o}(t) = m_n(t - t_o). \quad (15)$$

Before continuing with the basis transformations of polynomial curves, we find it useful to define the Bernstein, monomial, and Taylor basis matrices associated with any set of reals  $t_0, \dots, t_m \in \mathbb{R}$ , respectively, as

$$\mathbf{b}_n(t_0, \dots, t_m) := [b_n(t_0), \dots, b_n(t_m)] \quad (16a)$$

$$\mathbf{m}_n(t_0, \dots, t_m) := [m_n(t_0), \dots, m_n(t_m)] \quad (16b)$$

$$\boldsymbol{\tau}_{n,t_o}(t_0, \dots, t_m) := [\tau_n(t_0), \dots, \tau_n(t_m)]. \quad (16c)$$

An important property of square polynomial basis matrices is nonsingularity.

*Lemma 1 (Invertible Polynomial Basis Matrices):* For any pairwise distinct<sup>2</sup>  $t_0, \dots, t_n \in \mathbb{R}$  and any Taylor offset  $t_o \in \mathbb{R}$ , the polynomial basis matrices  $\mathbf{b}_n(t_0, \dots, t_n)$ ,  $\mathbf{m}_n(t_0, \dots, t_n)$  and  $\boldsymbol{\tau}_{n,t_o}(t_0, \dots, t_n)$  are all invertible.

*Proof:* See Appendix D-A. ■

## C. Basis Transformations of Polynomial Curves

As expected, alternative representations of polynomial curves have their advantages (e.g., the convexity of Bezier curves, the totally ordered basis<sup>3</sup> of monomial curves, and the local approximation feature of Taylor curves). Fortunately, one can easily perform change of polynomial basis.

*Lemma 2 (Change of Basis via Parameterwise Correspondence):* The basis transformation matrices between Bernstein, monomial, and Taylor bases (with a Taylor offset  $t_o \in \mathbb{R}$ )

$$\mathbf{b}_n(t) = \mathbf{T}_m^b(n) \mathbf{m}_n(t) = \mathbf{T}_\tau^b(n, t_o) \boldsymbol{\tau}_{n,t_o}(t) \quad (17a)$$

$$\mathbf{m}_n(t) = \mathbf{T}_b^m(n) \mathbf{b}_n(t) = \mathbf{T}_\tau^m(n, t_o) \boldsymbol{\tau}_{n,t_o}(t) \quad (17b)$$

$$\boldsymbol{\tau}_{n,t_o}(t) = \mathbf{T}_b^\tau(n, t_o) \mathbf{b}_n(t) = \mathbf{T}_m^\tau(n, t_o) \mathbf{m}_n(t) \quad (17c)$$

can be computed using any pairwise distinct<sup>2</sup>  $t_0, \dots, t_n \in \mathbb{R}$  as

$$\mathbf{T}_m^b(n) = \mathbf{T}_b^m(n)^{-1} = \mathbf{b}_n(t_0, \dots, t_n) \mathbf{m}_n(t_0, \dots, t_n)^{-1}$$

$$\mathbf{T}_\tau^b(n, t_o) = \mathbf{T}_b^\tau(n, t_o)^{-1} = \mathbf{b}_n(t_0, \dots, t_n) \boldsymbol{\tau}_{n,t_o}(t_0, \dots, t_n)^{-1}$$

$$\mathbf{T}_m^\tau(n, t_o) = \mathbf{T}_\tau^m(n, t_o)^{-1} = \mathbf{m}_n(t_0, \dots, t_n) \boldsymbol{\tau}_{n,t_o}(t_0, \dots, t_n)^{-1}$$

*Proof:* See Appendix D-B. ■

It is useful to highlight that the elements of the basis transformation matrices between monomial and Bernstein (Taylor, respectively) bases can be explicitly determined and these matrices are upper (lower, respectively) triangular with positive diagonal elements, see Appendix A for details.

*Lemma 3 (Polynomial Curve Equivalence):* Bezier, monomial, and Taylor curves of degree  $n \in \mathbb{N}$  (associated with a Taylor offset  $t_o \in \mathbb{R}$ ) are equivalent, i.e., for any  $t \in \mathbb{R}$

$$B_{p_0, \dots, p_n}(t) = M_{q_0, \dots, q_n}(t) = \Upsilon_{y_0, \dots, y_n}(t, t_o) \quad (18a)$$

<sup>2</sup>For numerically stable matrix inversion, a proper choice of pairwise distinct reals  $t_0, \dots, t_n \in [0, 1]$  is the uniformly spaced parameters over the unit interval, i.e.,  $t_i = \frac{i}{n}$  for  $i = 0, \dots, n$ .

<sup>3</sup>The monomial basis satisfies  $1 < t < t^2 < \dots < t^n$  or  $1 > t > t^2 > \dots > t^n$  for  $t \neq 1$ .

$$\mathbf{P}_n \mathbf{b}_n(t) = \mathbf{Q}_n \mathbf{m}_n(t) = \mathbf{Y}_n \boldsymbol{\tau}_{n,t_o}(t) \quad (18b)$$

if and only if their respective control point matrices  $\mathbf{P}_n = [p_0, \dots, p_n]$ ,  $\mathbf{Q}_n = [q_0, \dots, q_n]$ , and  $\mathbf{Y}_n = [y_0, \dots, y_n]$  are related to each other by the associated basis transformations as

$$\mathbf{P}_n = \mathbf{Q}_n \mathbf{T}_b^m(n) = \mathbf{Y}_n \mathbf{T}_b^\tau(n, t_o) \quad (19a)$$

$$\mathbf{Q}_n = \mathbf{P}_n \mathbf{T}_m^b(n) = \mathbf{Y}_n \mathbf{T}_\tau^m(n, t_o) \quad (19b)$$

$$\mathbf{Y}_n = \mathbf{P}_n \mathbf{T}_b^\tau(n, t_o) = \mathbf{Q}_n \mathbf{T}_m^\tau(n, t_o). \quad (19c)$$

*Proof:* See Appendix D-C.  $\blacksquare$

#### D. Reparametrization of Polynomial Curves

A common polynomial curve operation is the affine reparametrization from one parameter interval to another; for example, for proper time allocation in order to satisfy control (e.g., velocity, acceleration, jerk) constraints [4], [52].

*Lemma 4 (Polynomial Curve Reparametrization):* Bézier, monomial, and Taylor curves of degree  $n \in \mathbb{N}$  with respective control point matrices  $\mathbf{P}_n = [p_0, \dots, p_n]$ ,  $\mathbf{Q}_n = [q_0, \dots, q_n]$ , and  $\mathbf{Y}_n = [y_0, \dots, y_n]$  (and a Taylor offset  $t_o \in \mathbb{R}$ ) can be affinely reparametrized from interval  $[a, b]$  to  $[c, d]$  (with  $a < b$  and  $c < d$ ) as

$$B_{\hat{p}_0, \dots, \hat{p}_n}(t) = B_{p_0, \dots, p_n} \left( \frac{b-a}{d-c}t + \frac{ad-bc}{d-c} \right) \quad (20a)$$

$$M_{\hat{q}_0, \dots, \hat{q}_n}(t) = M_{q_0, \dots, q_n} \left( \frac{b-a}{d-c}t + \frac{ad-bc}{d-c} \right) \quad (20b)$$

$$\Upsilon_{\hat{y}_0, \dots, \hat{y}_n}(t, \hat{t}_o) = \Upsilon_{y_0, \dots, y_n} \left( \frac{b-a}{d-c}t + \frac{ad-bc}{d-c}, t_o \right) \quad (20c)$$

with the corresponding reparametrized control point matrices  $\hat{\mathbf{P}}_n = [\hat{p}_0, \dots, \hat{p}_n]$ ,  $\hat{\mathbf{Q}}_n = [\hat{q}_0, \dots, \hat{q}_n]$ , and  $\hat{\mathbf{Y}}_n = [\hat{y}_0, \dots, \hat{y}_n]$  that are given by

$$\hat{\mathbf{P}}_n = \mathbf{P}_n \mathbf{b}_n(t_0, \dots, t_n) \mathbf{b}_n(\hat{t}_0, \dots, \hat{t}_n)^{-1} \quad (21a)$$

$$\hat{\mathbf{Q}}_n = \mathbf{Q}_n \mathbf{m}_n(t_0, \dots, t_n) \mathbf{m}_n(\hat{t}_0, \dots, \hat{t}_n)^{-1} \quad (21b)$$

$$\hat{\mathbf{Y}}_n = \mathbf{Y}_n \boldsymbol{\tau}_{n,t_o}(t_0, \dots, t_n) \boldsymbol{\tau}_{n,\hat{t}_o}(\hat{t}_0, \dots, \hat{t}_n)^{-1} \quad (21c)$$

where  $t_0, \dots, t_n \in \mathbb{R}$  are arbitrary pairwise distinct reals<sup>2</sup>, and  $\hat{t}_i = \frac{d-c}{b-a}t_i - \frac{ad-bc}{b-a}$  for  $i = 0, \dots, n$ , and  $\hat{t}_o = \frac{d-c}{b-a}t_o - \frac{ad-bc}{b-a}$ .

*Proof:* See Appendix D-D.  $\blacksquare$

### III. BÉZIER METRICS

Bézier curves can be compared using various distances [44]. In this section, we particularly consider Bézier distances that define a true metric over the space of Bézier curves and can be computed efficiently in terms of Bézier control points, which is critical for computationally efficient and accurate Bézier approximation later in Section V.

*Definition 2 (Bézier Metric):* A real-valued distance measure  $d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m})$  between two Bézier curves  $B_{p_0, \dots, p_n}(t)$  and  $B_{q_0, \dots, q_m}(t)$  over the unit interval  $[0, 1]$  is a true metric if

- 1) it is nonnegative, i.e.,

$$d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) \geq 0$$

- 2) it is zero only for Bezier curves that are identical, i.e.,

$$d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) = 0$$

$$\iff B_{p_0, \dots, p_n}(t) = B_{q_0, \dots, q_m}(t) \quad \forall t \in [0, 1]$$

- 3) it is symmetric, i.e.,

$$d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) = d(B_{q_0, \dots, q_m}, B_{p_0, \dots, p_n})$$

- 4) it satisfies the triangle inequality, i.e.,

$$d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m})$$

$$\leq d(B_{p_0, \dots, p_n}, B_{r_0, \dots, r_k}) + d(B_{r_0, \dots, r_k}, B_{q_0, \dots, q_m}).$$

#### A. L2-Norm & Frobenius-Norm Distances of Bézier Curves

A widely used Bézier metric is the L2-norm distance that can be analytically computed using control points [45], [46].

*Definition 3 (Bézier L2-Norm Distance):* The L2-norm distance of two Bézier curves  $B_{p_0, \dots, p_n}(t)$  and  $B_{q_0, \dots, q_m}(t)$  over the unit interval  $[0, 1]$  is defined as

$$d_{L2}(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) := \left( \int_0^1 \|B_{p_0, \dots, p_n}(t) - B_{q_0, \dots, q_m}(t)\|^2 dt \right)^{\frac{1}{2}}$$

where  $\|\cdot\|$  denotes the L2 (a.k.a. Euclidean) norm of vectors.

*Proposition 1 (Analytic Bezier L2-Norm Distance):* The L2-norm distance of the  $n^{\text{th}}$ -order Bézier curves  $B_{p_0, \dots, p_n}(t)$  and  $B_{q_0, \dots, q_n}(t)$ , with respective control point matrices  $\mathbf{P}_n = [p_0, \dots, p_n]$  and  $\mathbf{Q}_n = [q_0, \dots, q_n]$ , is explicitly given by

$$d_{L2}(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) = \text{tr} \left( (\mathbf{P}_n - \mathbf{Q}_n) \mathbf{W}_n (\mathbf{P}_n - \mathbf{Q}_n)^T \right)^{\frac{1}{2}}$$

where  $\text{tr}$  and  $(\cdot)^T$  denote the trace and transpose operators, respectively, and the Bézier L2-norm weight matrix  $\mathbf{W}_n \in \mathbb{R}^{(n+1) \times (n+1)}$  is defined as

$$[\mathbf{W}_n]_{i+1,j+1} := \frac{1}{2n+1} \frac{\binom{n}{i} \binom{n}{j}}{\binom{2n}{i+j}} \quad \forall i, j \in \{0, \dots, n\}.$$

*Proof:* See Appendix D-E.  $\blacksquare$

Hence, the L2-norm distance of Bézier curves is a weighted Frobenius norm of their control point difference, which motivates another standard Bézier metric.<sup>4</sup>

*Definition 4 (Bézier Frobenius-Norm Distance):* The Frobenius-norm distance of the  $n^{\text{th}}$ -order Bézier curves  $B_{p_0, \dots, p_n}(t)$  and  $B_{q_0, \dots, q_n}(t)$ , with control point matrices  $\mathbf{P}_n = [p_0, \dots, p_n]$  and  $\mathbf{Q}_n = [q_0, \dots, q_n]$ , is defined as

$$\begin{aligned} d_F(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) &:= \|\mathbf{P}_n - \mathbf{Q}_n\|_F \\ &= \text{tr} \left( (\mathbf{P}_n - \mathbf{Q}_n)(\mathbf{P}_n - \mathbf{Q}_n)^T \right)^{\frac{1}{2}} \\ &= \sqrt{\sum_{i=0}^n \|p_i - q_i\|^2} \end{aligned}$$

where  $\|\cdot\|_F$  denotes the Frobenius norm of matrices.

<sup>4</sup>Similarly, one can define alternative matrix-norm-induced distance metrics for Bézier curves; however, we are particularly interested in L2-norm and Frobenius-norm distances because they are strongly related with the optimal least squares reduction of Bézier curves discussed in Section IV-B.

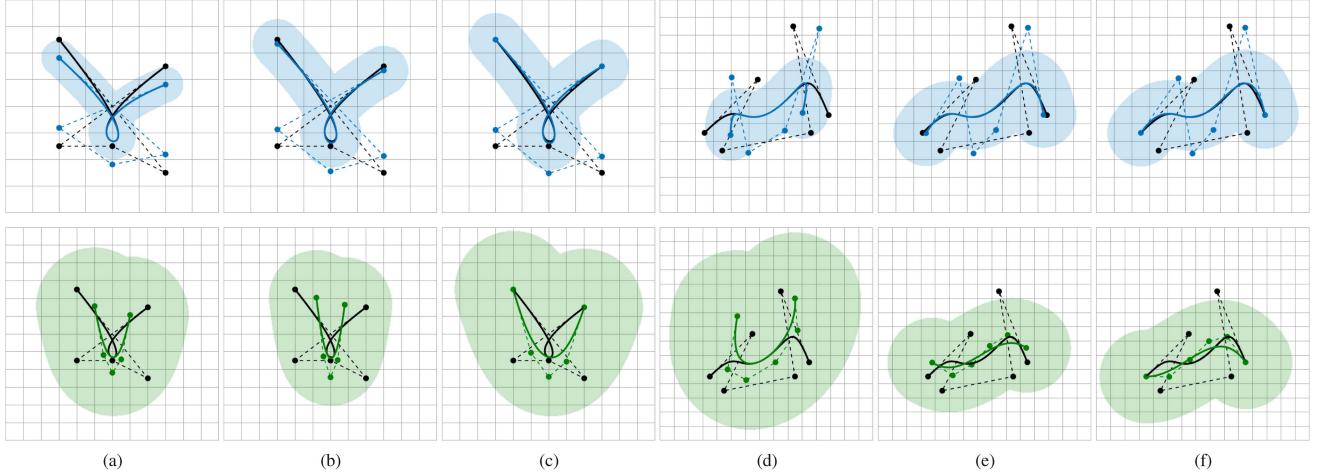


Fig. 2. Maximum control-point distance defines a geometric bound on a Bézier curve (black) relative to another Bézier curve (blue, green), for example, its degree reduction. (top) Degree-one and (bottom) degree-two reduction based on (a), (d) Taylor reduction, (b), (e) least squares reduction, (c), (f) uniform matching reduction. Here, the reduced Bézier curves are elevated again to have the same number of control points.

It is useful to remark that any distance measure of the  $n^{\text{th}}$ -order Bézier curves can be adapted to handle Bézier curves of different orders via degree elevation (see Section IV-A).<sup>5</sup>

### B. Hausdorff and Maximum Distances of Bézier Curves

As an alternative to norm-induced algebraic Bézier metrics, one can also compare Bézier curves based on set-theoretic distance measures as follows.

**Definition 5 (Bézier Hausdorff & Maximum Distances):** The Haussdorff and (parameterwise) maximum distances between two Bézier curves  $B_{p_0, \dots, p_n}(t)$  and  $B_{q_0, \dots, q_m}(t)$  over the unit interval  $[0,1]$  are, respectively, defined as

$$d_H(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m})$$

$$:= \max \left( \begin{array}{l} \max_{t_p \in [0,1]} \min_{t_q \in [0,1]} \|B_{p_0, \dots, p_n}(t_p) - B_{q_0, \dots, q_m}(t_q)\|, \\ \max_{t_q \in [0,1]} \min_{t_p \in [0,1]} \|B_{p_0, \dots, p_n}(t_p) - B_{q_0, \dots, q_m}(t_q)\| \end{array} \right)$$

$$d_M(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) := \max_{t \in [0,1]} \|B_{p_0, \dots, p_n}(t) - B_{q_0, \dots, q_m}(t)\|.$$

Unfortunately, both the Haussdorff and maximum distances of Bézier curves do not accept an analytic solution in terms of control points in general, but can be analytically bounded above by the maximum distance of Bézier control points.

### C. Control-Point Distance of Bézier Curves

We introduce a new analytic Bézier metric that defines a relative geometric bound between Bézier curves, see Fig. 2.

<sup>5</sup>Let  $d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m})$  be a distance measure for the  $n^{\text{th}}$ -order Bézier curves. It can be extended to any arbitrary Bézier curves  $B_{p_0, \dots, p_n}$  and  $B_{q_0, \dots, q_m}$  as

$$\begin{aligned} d(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) \\ := d(B_{[p_0, \dots, p_n]E(n, \max(n, m))}, B_{[q_0, \dots, q_m]E(m, \max(n, m))}) \end{aligned}$$

where  $E(n, m)$  denotes the elevation matrix defined in Definition 7.

**Definition 6 (Bezier Control-Point Distance):** The maximum control-point distance of  $n^{\text{th}}$ -order Bézier curves is defined as

$$d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) := \max_{i=0, \dots, n} \|p_i - q_i\|.$$

**Proposition 2 (Bézier Distance Order):** The Frobenius, Haussdorff, and parameterwise & control-pointwise maximum distances of the  $n^{\text{th}}$ -order Bézier curves satisfy

$$\begin{aligned} d_H(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) &\leq d_M(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) \\ &\leq d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) \\ &\leq d_F(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) \\ &\leq \sqrt{n} d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}). \end{aligned}$$

*Proof:* See Appendix D-F. ■

**Proposition 3 (Relative Bézier Bound):** In the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ , an  $n^{\text{th}}$ -order Bézier curve is contained in the dilation of an another  $n^{\text{th}}$ -order Bézier curve by their maximum control-point distance, i.e.,

$$B_{p_0, \dots, p_n}([0, 1])$$

$$\subseteq B_{q_0, \dots, q_n}([0, 1]) \oplus \mathcal{B}_d(d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n})) \quad (22)$$

$$\subseteq \text{conv}(q_0, \dots, q_n) \oplus \mathcal{B}_d(d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n})) \quad (23)$$

where  $\mathcal{B}_d(r) := \{x \in \mathbb{R}^d \mid \|x\| \leq r\}$  denotes the  $d$ -dimensional closed Euclidean ball of radius  $r \geq 0$  centered at the origin.

*Proof:* See Appendix D-G. ■

The ordering (a.k.a. equivalence) relation of Bézier distances in Proposition 2 makes the maximum control-point distance a computationally efficient tool for discriminative comparison of Bézier curves independent of degree  $n$ , whereas the Frobenius-norm distance tends to increase with increasing  $n$ . Moreover, similar to the convexity in Property 3, the relative bound of Bézier curves via the maximum control-point distance in Proposition 3 offers an alternative way of constraining Bézier control points for safety and constraint verification in motion planning. We continue below with how different Bézier metrics behave under Bézier degree elevation and reduction operations.

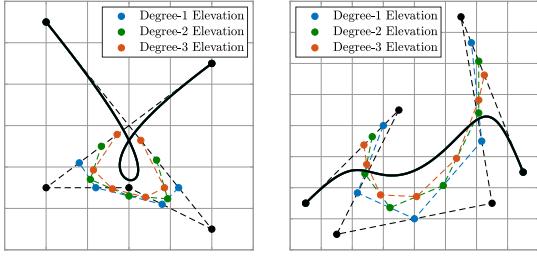


Fig. 3. Degree elevation represents a Bézier curve exactly by more control points that asymptotically converge to the curve as degree goes to infinity.

#### IV. BEZIER DEGREE ELEVATION AND REDUCTION

In this section, we briefly summarize the degree elevation and reduction operations of Bézier curves for (approximately) representing them with more or fewer control points. In particular, degree reduction is another building block of high-order Bézier approximations with multiple low-order Bézier segments. Accordingly, we introduce a new degree reduction method for approximating Bézier geometry more accurately.

##### A. Degree Elevation

Degree elevation generates an exact representation of Bézier curves with more control points, as illustrated in Fig. 3.

**Definition 7 (Degree Elevation):** A Bézier curve  $B_{q_0, \dots, q_m}$  of higher degree  $m$  with control points  $q_0, \dots, q_m$  is said to be the *degree elevation* of another Bézier curve  $B_{p_0, \dots, p_n}$  of lower degree  $n \leq m$  with control points  $p_0, \dots, p_n$  if and only if the curves are parameterwise identical, i.e.,

$$B_{q_0, \dots, q_m}(t) = B_{p_0, \dots, p_n}(t) \quad \forall t \in \mathbb{R}. \quad (24)$$

Degree elevation can be analytically computed as follows.

**Proposition 4 (Elevated Control Points):** A Bézier curve  $B_{q_0, \dots, q_m}$  of degree  $m$  is the degree elevation of a Bézier curve  $B_{p_0, \dots, p_n}$  of degree  $n \leq m$  if and only if the control point matrices  $\mathbf{Q}_m = [q_0, \dots, q_m]$  and  $\mathbf{P}_n = [p_0, \dots, p_n]$  satisfy

$$\mathbf{Q}_m = \mathbf{P}_n \mathbf{E}(n, m) \quad (25)$$

where the degree elevation matrix  $\mathbf{E}(n, m)$  is defined as

$$\mathbf{E}(n, m) := \mathbf{T}_m^b(n) \mathbf{I}_{(n+1) \times (m+1)} \mathbf{T}_b^m(m) \quad (26)$$

and  $\mathbf{I}_{n+1 \times (m+1)}$  is the  $(n+1) \times (m+1)$  rectangular identify matrix with ones in the main diagonal and zeros elsewhere.

*Proof:* See Appendix D-H. ■

Observe that (26) leverages the change of basis between Bernstein and monomial bases because degree elevation of monomial curves is trivial.

Higher order Bernstein basis vectors can also be obtained from lower ones via degree elevation, which offers another way of determining the elevation matrix.

**Proposition 5 (Elevated Bernstein Basis):** For any  $n \leq m$ , the elevation matrix  $\mathbf{E}(n, m)$  relates Bernstein basis vectors as

$$b_n(t) = \mathbf{E}(n, m) b_m(t) \quad \forall t \in \mathbb{R}. \quad (27)$$

Hence,  $\mathbf{E}(n, m)$  can be determined as

$$\mathbf{E}(n, m) = \mathbf{b}_n(t_0, \dots, t_m) \mathbf{b}_m(t_0, \dots, t_m)^{-1} \quad (28)$$

where  $t_0, \dots, t_m \in \mathbb{R}$  are an arbitrary selection of pairwise distinct curve parameters, i.e.,  $t_i \neq t_j$  for all  $i \neq j$ . ■

*Proof:* See Appendix D-I. ■

In fact, the elements of the degree elevation matrix can be determined explicitly [44], [45].

**Proposition 6 (Elevation Matrix Elements):** For any  $n \leq m$ , the elements of the elevation matrix  $\mathbf{E}(n, m)$  are given by

$$[\mathbf{E}(n, m)]_{i+1, j+1} = \begin{cases} \frac{\binom{n}{i} \binom{m-n}{j-i}}{\binom{m}{j}}, & \text{if } m - n \geq j - i \geq 0 \\ 0, & \text{otherwise,} \end{cases}$$

where  $i = 0, \dots, n$  and  $j = 0, \dots, m$ . ■

*Proof:* See Appendix D-J. ■

**1) Important Elevation Matrix Properties:** The degree elevation matrices are full rank and have unit column sum [45].

**Proposition 7 (Full Rank Elevation Matrix):** For any  $n \leq m$ , the elevation matrix  $\mathbf{E}(n, m) \in \mathbb{R}^{(n+1) \times (m+1)}$  is full rank of  $n+1$ , i.e.,  $\text{rank}(\mathbf{E}(n, m)) = n+1$ .

*Proof:* See Appendix D-K. ■

**Proposition 8 (Elevation Matrix Row & Column Sum):** For any  $n \leq m$ , the sum of each column of the elevation matrix  $\mathbf{E}(n, m)$  is one, whereas each of its rows sums to  $\frac{m+1}{n+1}$ , i.e.,

$$\mathbf{1}_{1 \times (n+1)} \mathbf{E}(n, m) = \mathbf{1}_{1 \times (m+1)} \quad (29a)$$

$$\mathbf{E}(n, m) \mathbf{1}_{(m+1) \times 1} = \frac{m+1}{n+1} \mathbf{1}_{(n+1) \times 1} \quad (29b)$$

where  $\mathbf{1}_{n \times m}$  denotes the  $n \times m$  matrix of all ones. ■

*Proof:* See Appendix D-L. ■

**2) Bézier Metrics Under Degree Elevation:** Different Bézier metrics behave differently under degree elevation: the L2-norm distance stays constant, the Frobenius norm distance might increase, and the control-point distance is nonincreasing.

**Proposition 9 (Invariance of Bézier L2-norm distance under degree elevation):** The L2-norm distance of Bézier curves,  $B_{p_0, \dots, p_n}(t)$ , and  $B_{q_0, \dots, q_m}(t)$  over the unit interval  $[0, 1]$ , are preserved under degree elevation, i.e.,

$$\begin{aligned} d_{L2}(B_{[p_0, \dots, p_n]} \mathbf{E}(n, k), B_{[q_0, \dots, q_m]} \mathbf{E}(m, h)) \\ = d_{L2}(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_m}) \end{aligned}$$

for any  $k \geq n$  and  $h \geq m$ . ■

*Proof:* See Appendix D-M. ■

**Proposition 10 (Elevated Frobenius Distance):** Under degree elevation, the Frobenius distance of the  $n^{\text{th}}$ -order Bézier curves satisfies for any  $m \geq n \in \mathbb{N}$  that

$$\begin{aligned} d_F(B_{[p_0, \dots, p_n]} \mathbf{E}(n, m), B_{[q_0, \dots, q_n]} \mathbf{E}(n, m))^2 \\ \leq \frac{m+1}{n+1} d_F(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n})^2. \end{aligned}$$

*Proof:* See Appendix D-N. ■

**Proposition 11 (Nonincreasing Elevated Control-Point Distance):** The maximum control-point distance of the  $n^{\text{th}}$ -order Bézier curves is nonincreasing under degree elevation, i.e.,

$$\begin{aligned} d_C(B_{[p_0, \dots, p_n]} \mathbf{E}(n, m), B_{[q_0, \dots, q_n]} \mathbf{E}(n, m)) \\ \leq d_C(B_{p_0, \dots, p_n}, B_{q_0, \dots, q_n}) \end{aligned}$$

for any  $m \geq n \in \mathbb{N}$ . ■

*Proof:* See Appendix D-O. ■

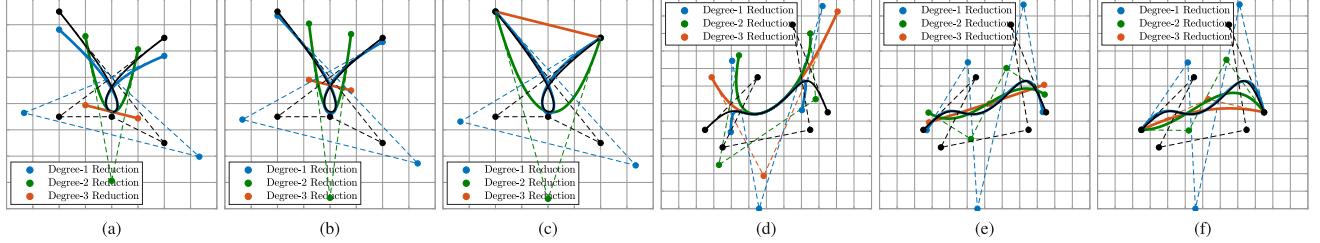


Fig. 4. Degree reduction approximately represents Bézier curves with fewer control points. (a), (d) Taylor degree reduction with a Taylor offset  $t_o = 0.5$ , (b), (e) Least squares reduction, (c), (f) Uniform matching reduction with uniformly spaced parameters  $t_0, \dots, t_m$  over the unit interval, i.e.,  $t_i = \frac{i}{m}$  for  $i = 0, \dots, m$ .

Thus, the ordering relation of Bézier distances in Proposition 2, the relative geometric bound of Bézier curves in Proposition 3, and the nonincreasing property under degree elevation in Proposition 11 make the maximum control-point distance an intuitive analytic metric for comparing Bézier curves.

Finally, it is useful to note that Bézier control points get closer to the curve with increasing elevation degree.

*Proposition 12 ([44]) (Asymptotic Behavior of Degree Elevation):* As elevation degree goes to infinity, all elevated Bézier control points are contained in the Bézier curve, i.e.,

$$\lim_{m \rightarrow \infty} [p_0, \dots, p_n] \mathbf{E}(n, m) \subseteq \{B_{p_0, \dots, p_n}(t) \mid 0 \leq t \leq 1\}.$$

## B. Degree Reduction

As opposed to degree elevation, Bézier degree reduction aims to approximately represent a Bézier curve with fewer control points, as illustrated in Fig. 4. Hence, degree reduction is naturally defined as the inverse of degree elevation.<sup>6</sup>

*Definition 8 (Degree Reduction):* A Bézier curve  $B_{q_0, \dots, q_m}$  of lower degree  $m$  with control points  $\mathbf{Q}_m = [q_0, \dots, q_m]$  is said to be a *degree reduction* of another Bézier curve  $B_{p_0, \dots, p_n}$  of higher degree  $n \geq m$  with control points  $\mathbf{P}_n = [p_0, \dots, p_n]$  if and only if the control points are related to each other by

$$\mathbf{Q}_m = \mathbf{P}_n \mathbf{R}(n, m) \quad (30)$$

where  $\mathbf{R}(n, m) \in \mathbb{R}^{(n+1) \times (m+1)}$  denotes a degree reduction matrix that is a right inverse of the elevation matrix  $\mathbf{E}(m, n)$ , i.e.,

$$\mathbf{E}(m, n) \mathbf{R}(n, m) = \mathbf{I}_{(m+1) \times (m+1)}. \quad (31)$$

That is to say, the degree elevation from  $m$  to  $n$  followed by a degree reduction from  $n$  to  $m$  preserves Bézier curves; but, the reverse is not correct in general. Also note that the right inverse of the elevation matrix is not unique, which allows many alternative ways of constructing a reduction matrix.

1) *Least Squares Reduction:* A standard choice for degree reduction is the pseudoinverse of the elevation matrix [44].

*Definition 9 (Least Squares Reduction):* The *least squares* reduction matrix  $\mathbf{R}_{L2}(n, m)$  is defined as the pseudo-inverse of the elevation matrix  $\mathbf{E}(m, n)$  that is explicitly given by

$$\mathbf{R}_{L2}(n, m) = \mathbf{E}(m, n)^T (\mathbf{E}(m, n) \mathbf{E}(m, n)^T)^{-1}. \quad (32)$$

Note that  $\mathbf{E}(m, n) \mathbf{E}(m, n)^T$  is invertible for any  $m \leq n \in \mathbb{N}$  because  $\mathbf{E}(m, n)$  is full rank (Proposition 7), which implies  $\mathbf{E}(m, n) \mathbf{R}_{L2}(n, m) = \mathbf{I}_{(m+1) \times (m+1)}$ . An example of least

<sup>6</sup>Many existing notions of Bézier degree reduction methods that are defined in terms of different Bézier distances (possibly with end-point constraints) can be unified using the inverse of degree elevation [48].

squares reduction is presented in Fig. 4(b) and (e), where the Bézier end-points are not preserved after degree reduction.

The least squares degree reduction is known to be optimal in the sense of the L2- and Frobenius-norm distances [45], [53].

*Proposition 13 (Optimality of Least Squares Reduction):*

With respect to the L2-norm and Frobenius-norm distances, the optimal  $m^{\text{th}}$ -order Bezier curve  $B_{q_0, \dots, q_m}(t)$  with control points  $\mathbf{Q}_m = [q_0, \dots, q_m]$  that is closest to a higher  $n^{\text{th}}$ -order Bézier curve  $B_{p_0, \dots, p_n}(t)$  with control points  $\mathbf{P}_n = [p_0, \dots, p_n]$  is given by the least squares reduction, i.e.,

$$\begin{aligned} \mathbf{Q}_m &= \operatorname{argmin}_{q_0, \dots, q_m \in \mathbb{R}^d} d_{L2}(B_{p_0, \dots, p_n}, B_{[q_0, \dots, q_m]} \mathbf{E}(m, n)) \\ &= \operatorname{argmin}_{q_0, \dots, q_m \in \mathbb{R}^d} d_F(B_{p_0, \dots, p_n}, B_{[q_0, \dots, q_m]} \mathbf{E}(m, n)) \\ &= \mathbf{P}_n \mathbf{R}_{L2}(n, m). \end{aligned}$$

*Proof:* See Appendix D-P. ■

2) *Taylor Reduction:* Another classical degree reduction is Taylor approximation that preserves local derivatives.

*Definition 10 (Taylor Reduction):* The Taylor reduction matrix  $\mathbf{R}_{\tau, t_o}(n, m)$  for approximating an  $n^{\text{th}}$ -order Bezier curve around  $t_o \in \mathbb{R}$  by a lower  $m^{\text{th}}$ -order Bezier curve is defined as

$$\mathbf{R}_{\tau, t_o}(n, m) := \mathbf{T}_\tau^b(n, t_o) \mathbf{I}_{(n+1) \times (m+1)} \mathbf{T}_b^\tau(m, t_o) \quad (33)$$

where  $\mathbf{T}_\tau^b$  and  $\mathbf{T}_b^\tau$  are the Taylor-to-Bernstein and the Bernstein-to-Taylor basis transformation matrices in Lemma 2.

In other words, Taylor reduction perform a basis transformation from Bernstein-to-Taylor basis, and ignores some higher order Taylor basis elements, and then comes back to Bernstein basis. Hence, it has a strong bias and local expressiveness around the Taylor offset  $t_o$ , as illustrated in Fig. 4(a) and (d).

*Proposition 14 (Taylor Reduction as Elevation Inverse):* The Taylor reduction matrix  $\mathbf{R}_{\tau, t_o}(n, m)$  is a right inverse of the elevation matrix  $\mathbf{E}(m, n)$ , i.e., for any  $m \leq n \in \mathbb{N}$

$$\mathbf{E}(m, n) \mathbf{R}_{\tau, t_o}(n, m) = \mathbf{I}_{(m+1) \times (m+1)}. \quad (34)$$

*Proof:* See Appendix D-Q. ■

Observe in Fig. 4 that both the least-squares and Taylor reduction methods offer less freedom in controlling the shape of Bézier approximations; for example, the end points of the original curve are unpreserved after degree reduction, which is essential for path smoothing with boundary conditions [14].

3) *Parameterwise Matching Reduction:* In order to accurately approximate curve shape and geometry, we propose a new parameterwise matching reduction method that preserves a finite set of curve points after degree reduction.

*Definition 11 (Parameterwise Matching Reduction):* For Bézier degree reduction from a higher degree  $n$  to a lower degree

$m \leq n$ , the parameterwise matching degree reduction matrix  $\mathbf{R}_{t_0, \dots, t_m}(n, m)$  associated with pairwise distinct parameters  $t_0, \dots, t_m \in \mathbb{R}$  (i.e.,  $t_i \neq t_j$  for all  $i \neq j$ ) is defined as

$$\mathbf{R}_{t_0, \dots, t_m}(n, m) := \mathbf{b}_n(t_0, \dots, t_m) \mathbf{b}_m(t_0, \dots, t_m)^{-1} \quad (35)$$

where  $\mathbf{b}_n(t_0, \dots, t_m)$  is the Bernstein basis matrix in (16).

For example, a numerically stable choice of  $t_0, \dots, t_m$  over the unit interval is the uniformly spaced parameters in  $[0, 1]$ , i.e.,  $t_i = \frac{i}{m}$  for  $i = 0, \dots, m$ . We call the corresponding reduction operation as the *uniform matching reduction*.

As expected, the parameterwise matching reduction of Bézier curves keeps curve points unchanged at  $t_0, \dots, t_m$ .

**Proposition 15 (Preserved Points of Matching Reduction):** For any pairwise distinct  $t_0, \dots, t_m \in \mathbb{R}$ , a Bézier curve  $B_{p_0, \dots, p_n}(t)$  of degree  $n$  and its parameterwise matching reduction  $B_{q_0, \dots, q_m}(t)$  of degree  $m \leq n$  with control points

$$[q_0, \dots, q_m] = [p_0, \dots, p_n] \mathbf{R}_{t_0, \dots, t_m}(n, m) \quad (36)$$

match at the curve parameters  $t_0, \dots, t_m$ , i.e.,

$$B_{q_0, \dots, q_m}(t) = B_{p_0, \dots, p_n}(t) \quad \forall t = t_0, \dots, t_m. \quad (37)$$

*Proof:* See Appendix D-R. ■

**Proposition 16 (Matching Reduction as Elevation Inverse):** For any  $m \leq n$  and pairwise distinct reals  $t_0, \dots, t_m \in \mathbb{R}$ , the parameterwise matching degree matrix  $\mathbf{R}_{t_0, \dots, t_m}(n, m)$  is a right inverse of the elevation matrix  $\mathbf{E}(m, n)$ ,

$$\mathbf{E}(m, n) \mathbf{R}_{t_0, \dots, t_m}(n, m) = \mathbf{I}_{(m+1) \times (m+1)}. \quad (38)$$

*Proof:* See Appendix D-S. ■

Another interesting connection between degree elevation and matching reduction is their shared matrix form.

**Proposition 17 (Shared Form of Matching Reduction and Elevation Matrix):** For any  $n, m \in \mathbb{N}$  and any pairwise distinct reals  $t_0, \dots, t_m \in \mathbb{R}$ , the Bernstein basis matrices satisfy<sup>7</sup>

$$\mathbf{b}_n(t_0, \dots, t_m) \mathbf{b}_m(t_0, \dots, t_m)^{-1} = \begin{cases} \mathbf{E}(n, m) & , \text{if } n \leq m \\ \mathbf{R}_{t_0, \dots, t_m}(n, m), & \text{if } n \geq m \end{cases}$$

where  $\mathbf{E}(n, n) = \mathbf{R}_{t_0, \dots, t_m}(n, n) = \mathbf{I}_{(n+1) \times (n+1)}$ .

*Proof:* It follows from Proposition 5 and Definition 11. ■

A critical property of matching reduction is that the degree-one reduction error can be determined analytically.

**Proposition 18 (Degree-One Matching Reduction Error):** For any pairwise distinct  $t_0, \dots, t_n \in \mathbb{R}$ , the difference between a Bézier curve  $B_{p_0, \dots, p_{n+1}}(t)$  and its parameterwise matching degree reduction  $B_{q_0, \dots, q_n}(t)$ , with control points

$$[q_0, \dots, q_n] = [p_0, \dots, p_{n+1}] \mathbf{R}_{t_0, \dots, t_n}(n+1, n) \quad (39)$$

is given by

$$B_{p_0, \dots, p_{n+1}}(t) - B_{q_0, \dots, q_n}(t) = \Delta p \prod_{i=0}^n (t - t_i) \quad (40)$$

<sup>7</sup>It is important to highlight that for any distinct  $t_0, \dots, t_n \in \mathbb{R}$ , the inverse of the Bernstein matrix can be computed analytically using the Bernstein-to-monomial basis transformation  $\mathbf{b}_n(t_0, \dots, t_n) = \mathbf{T}_m^b \mathbf{m}(t_0, \dots, t_n)$  as

$$\mathbf{b}_n(t_0, \dots, t_n)^{-1} = \mathbf{m}_n(t_0, \dots, t_n)^{-1} \mathbf{T}_m^b(n)$$

since the inverse of the monomial (a.k.a. Vandermonde) matrix is analytically available [54], and the elements of Bernstein-to-monomial basis transformation  $\mathbf{T}_m^b(n)$  can be determined explicitly, see Appendix D2.

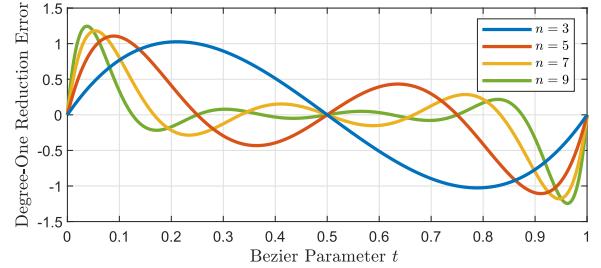


Fig. 5. Scaled degree-one reduction error function  $\varepsilon_n(t)$  for uniform matching reduction of Bézier curves that is defined as  $\varepsilon_n(t) = \frac{\prod_{i=0}^n t - \frac{i}{n}}{\prod_{i=0}^{n-1} \frac{1}{2^n} - \frac{i}{n}}$ .

where

$$\Delta p = \sum_{i=0}^{n+1} (-1)^{n+1-i} \binom{n+1}{i} p_i. \quad (41)$$

*Proof:* See Appendix D-U. ■

It is important to observe that the degree-one matching reduction difference vector  $\Delta p$  is independent of the selection of matching parameters  $t_0, \dots, t_n$  where the reduction error is zero.<sup>8</sup> Moreover, the polynomial product form of the degree-one matching reduction error, illustrated in Fig. 5, plays a key role in determining how many local low-order Bézier segments are needed for approximating high-order Bézier curves accurately, as discussed below in Section V.

## V. ADAPTIVE DEGREE REDUCTION AND SPLITTING OF BÉZIER CURVES

In this section, we consider the problem of approximating high-order Bézier curves by multiple lower order Bézier segments. We first describe how Bézier approximations can be performed over a given finite partition of the unit interval, and then propose linear and binary search methods for adaptive approximation of high-order Bézier curves by lower order Bézier segments at any desired accuracy level. We also present a rule of thumb for accurate Bézier discretization.

### A. Bézier Approximation Over a Partition of the Unit Interval

Consider an  $n^{\text{th}}$ -order Bézier curve  $B_{p_0, \dots, p_n}(t)$  with control points  $\mathbf{P}_n = [p_0, \dots, p_n]$  defined over the unit interval  $[0, 1]$ . Suppose  $T = [t_0, \dots, t_k]$  is an ordered list of distinct parameters that defines a partition of the unit interval into  $k$  splits, i.e.,  $0 = t_0 < t_1 < \dots < t_{k-1} < t_k = 1$ . Accordingly, the Bézier curve  $B_{p_0, \dots, p_n}(t)$  can be locally approximated over each parameter subinterval  $[t_{i-1}, t_i]$  by a lower  $m^{\text{th}}$ -order Bezier curve  $B_{q_{0,i}, \dots, q_{m,i}}(t)$  whose control points  $\mathbf{Q}_{m,i} = [q_{0,i}, \dots, q_{m,i}]$  is obtained based on a choice of a degree reduction matrix  $\mathbf{R}(n, m)$  (see Definition 8) as

$$\mathbf{Q}_{m,i} = \text{DegreeReduction}(\mathbf{P}_{n,i}, m) := \mathbf{P}_{n,i} \mathbf{R}(n, m) \quad (42)$$

<sup>8</sup>Finding optimal matching parameters that minimize the peak reduction error is an open research problem and outside the scope of this article. We observe from Fig. 5 that optimal matching parameters should be nonuniformly spaced with a bias toward the ends points. In this article, we consider the uniformly spaced matching parameters and their adaptive selection based on binary search in Section V.

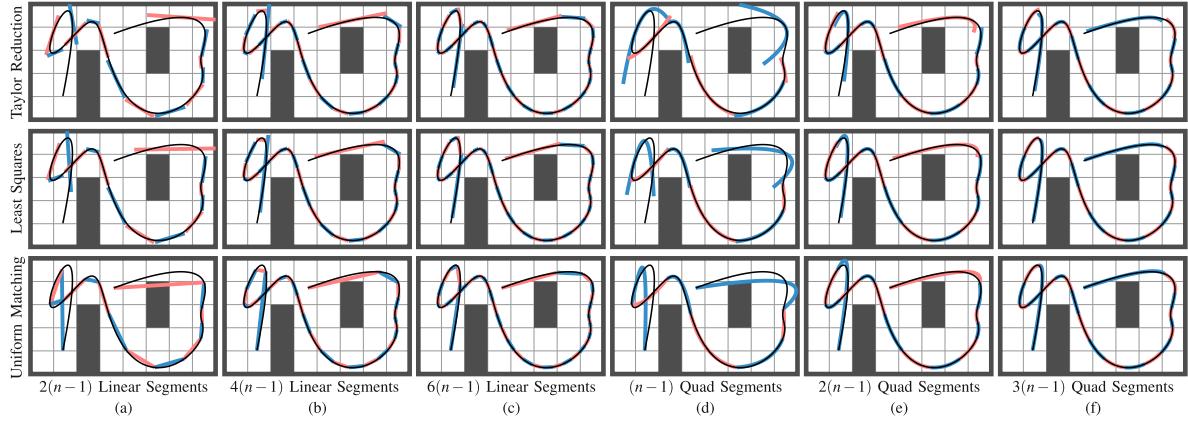


Fig. 6. Approximation of a Bézier curve of degree  $n = 8$  by (a, b, c) linear and (d, e, f) quadratic Bézier segments using (top) Taylor, (middle) least squares, and (bottom) uniform matching reduction. The Taylor offset is set to be 0.5, and a uniform partition of the unit interval is used with (a)  $2(n - 1)$ , (b)  $4(n - 1)$ , (c)  $6(n - 1)$  partition elements for linear approximations, and (d)  $(n - 1)$ , (e)  $2(n - 1)$ , (f)  $3(n - 1)$  partition elements for quadratic approximations. Bézier approximation rule: An  $n^{\text{th}}$ -order Bézier curve can be approximated accurately by  $3(n - 1)$  quadratic and  $6(n - 1)$  linear uniform matching Bézier curves.

---

**Algorithm 1:** Bézier Approximation Over a Partition of the Unit Interval.

---

**Input:**  $\mathbf{P}_n \in \mathbb{R}^{d \times n}$ : Bezier Control Points  
 $m \in \mathbb{N}$ : Reduction Degree  
 $T = [t_0, \dots, t_k]$ : Partition of the Unit Interval  
**Output:**  $\mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ : List of Reduced Control Points where  $B_{\mathbf{Q}_{m,i}}([0, 1])$  approximates  $B_{\mathbf{P}_n}([t_{i-1}, t_i])$

---

```

1:  $k \leftarrow \text{length}(T) - 1$            // Number of Segments
2: for  $i \leftarrow 1$  to  $k$  do
3:    $\mathbf{P}_{n,i} \leftarrow \text{Reparametrize}(\mathbf{P}_n, [t_{i-1}, t_i])$ 
4:    $\mathbf{Q}_{m,i} \leftarrow \text{DegreeReduction}(\mathbf{P}_{n,i}, m)$ 
5: return  $\mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ 

```

---

using the reparametrization  $B_{p_{0,i}, \dots, p_{n,i}}(t)$  of  $B_{p_0, \dots, p_n}(t)$  from  $[t_{i-1}, t_i]$  to the unit interval  $[0, 1]$  with new control points  $\mathbf{P}_{n,i} = [p_{0,i}, \dots, p_{n,i}]$  (see Lemma 4) that are obtained as

$$\begin{aligned} \mathbf{P}_{n,i} &= \text{Reparameterize}(\mathbf{P}_i, [t_{i-1}, t_i]) \\ &:= \mathbf{b}_n \left( 0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1 \right) \mathbf{b}_n \left( s_i(0), s_i \left( \frac{1}{n} \right), \dots, s_i \left( \frac{n-1}{n} \right), s_i(1) \right)^{-1} \end{aligned}$$

where  $s_i(t) = B_{t_{i-1}, t_i}(t) = t_{i-1}(1-t) + t_i t$  is the Bézier parameter scaling function. Hence, as described in Algorithm 1, the high-order Bézier curve  $B_{p_0, \dots, p_n}([0, 1])$  can be approximated by a collection of low-order Bézier segments  $B_{q_{0,i}, \dots, q_{m,i}}([0, 1])$  constructed over each partition element  $[t_{i-1}, t_i]$  such that  $B_{\mathbf{Q}_{m,i}}([0, 1])$  approximates  $B_{\mathbf{P}_n}([t_{i-1}, t_i])$ .

In Fig. 6, we illustrate approximating an 8<sup>th</sup>-order Bézier curve with linear and quadratic Bézier segments over the uniform partitions of the unit interval using Taylor, least squares, and uniform matching reductions. As seen in Fig. 6, the uniform matching reduction performs better in approximately representing the original curve shape than the least squares reduction which performs better than Taylor reduction. Also notice that the end points of the original curve are kept unchanged only under the uniform matching reduction.

### B. Bézier Approximation Rule

A practical question of approximating high-order Bézier curves by a finite number of low-order Bézier segments is what the required number of local Bézier segments is for an accurate Bézier discretization. As expected, the answer depends on the desired level of approximation accuracy and the degree of Bézier curves. In this part, we provide an answer based on the structural form of the Bézier approximation error of degree-one matching reduction (Proposition 18), and the numerical analysis of Bézier approximations in Section VI.

*A Rule of Thumb for Bézier Approximations:* An  $n^{\text{th}}$ -order Bézier curve can be accurately approximated by  $3(n - 1)$  quadratic and  $6(n - 1)$  linear Bézier curves obtained via uniform reduction.

According to Proposition 18, an  $n^{\text{th}}$ -order Bézier curve can be written as the sum of an  $(n - 1)^{\text{th}}$ -order reduced Bézier curve and a degree-one matching reduction error, which is a polynomial of order  $n$  in the product form. Note that the  $(n - 1)^{\text{th}}$ -order reduced Bézier curve can be better approximated with the same number of low-order Bézier segments than the original  $n^{\text{th}}$ -order Bézier curve. Hence, one can determine the required number of low-order Bézier segments for accurately approximating high-order Bézier curves by exploiting the functional form of the reduction error. As seen in Fig. 5, the degree-one matching reduction error of the  $n^{\text{th}}$ -order Bézier curves has  $(n - 1)$  extreme (local maximum and minimum) points. This implies that a proper approximation of the  $n^{\text{th}}$ -order Bézier curves structurally requires at least  $(n - 1)$  quadratic Bézier curves which has a single extremum. Because of the asymmetry of the approximation error around each extreme point, one needs at least  $2(n - 1)$  quadratic segments. Since the uniform matching reduction uses uniformly spaced parameters for approximation, we observe in our numerical studies in Section VI that the asymmetry around each extreme point can be better handled with  $3(n - 1)$  quadratic patches in practice. Our numerical analysis also shows that approximating  $n^{\text{th}}$ -order Bézier curves by  $3(n - 1)$  quadratic segments ensures a normalized (i.e., scale

---

**Algorithm 2:** Adaptive Degree Reduction and Splitting via Linear Search.

---

**Input:**  $\mathbf{P}_n \in \mathbb{R}^{d \times n}$ : Bezier Control Points  
 $m \in \mathbb{N}$ : Reduction Degree  
 $\varepsilon > 0$ : Approximation Tolerance

**Output:**  $T = [t_0, \dots, t_k]$ : Split Intervals  
 $\mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ : List of Reduced Control Points where  $B_{\mathbf{Q}_{m,i}}([0, 1])$  approximates  $B_{\mathbf{P}_n}([t_{i-1}, t_i])$

---

```

1:  $k \leftarrow 1$                                 // Number of Segments
2:  $T \leftarrow [0, 1]$                           // Initial Partition
3: while  $k < \text{length}(T)$  do
4:   for  $i \leftarrow 1$  to  $k$  do
5:      $\mathbf{P}_{n,i} \leftarrow \text{Reparametrize}(\mathbf{P}_n, [t_{i-1}, t_i])$ 
6:      $\mathbf{Q}_{m,i} \leftarrow \text{DegreeReduction}(\mathbf{P}_{n,i}, m)$ 
7:     if  $\text{BezierDistance}(\mathbf{P}_{n,i}, \mathbf{Q}_{m,i} \mathbf{E}(m, n)) > \varepsilon$  then
8:        $T \leftarrow [0, \frac{1}{k+1}, \dots, \frac{k}{k+1}, 1]$ 
9:       break
10:     $k \leftarrow k + 1$ 
11:  $k \leftarrow \text{length}(T) - 1$                   // Number of Segments
12: return  $T, \mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ 

```

---

invariant) approximation error below the order of  $10^{-3}$  for computing important curve features such as curve length, distance-to-point/line, and maximum velocity/acceleration. Similarly, since a quadratic polynomial structurally requires at least two linear curve segments for a proper representation of its unique extremum, we also observe from our numerical studies that approximating the  $n^{\text{th}}$ -order Bézier curves by  $6(n-1)$  linear Bézier segments yields a normalized approximation error in the order of  $10^{-3}$ . Finally, these results suggest that an accurate approximation of high-order Bézier curves with low-order Bézier segments requires more control points than the original curve, due to the relation between descriptive power and functional complexity of Bézier curves.

### C. Adaptive Bézier Approximation via Bézier Metrics

The Bézier approximation rule above holds for any Bézier curve in general, and ensures a proper structural representation of high-order Bézier curves at a certain level of accuracy. However, high-order Bézier curves might have redundant control points, for example, consider the degree elevation of Bézier curves in Fig. 3, and also different application settings might require different levels of approximation accuracy. Hence, it is desirable to perform Bézier approximation that is tailored to individual Bézier curves and can adaptively select the required number of Bézier segments and the partition of the unit interval based on the desired level of accuracy. In this part, we extend Bézier approximations over a given partition of the unit interval by incorporating a search strategy to automatically determine a partition of the unit interval in order to achieve a desired level of measurable approximation accuracy.

As discussed in Section V-A, an  $n^{\text{th}}$ -order Bézier curve  $B_{p_0, \dots, p_n}(t)$  can be locally approximated by the  $m^{\text{th}}$ -order Bezier segments over each element of a  $k$ -partition  $T = [t_0, \dots, t_k]$  of the unit interval by applying curve

---

**Algorithm 3:** Adaptive Degree Reduction and Splitting via Binary Search.

---

**Input:**  $\mathbf{P}_n \in \mathbb{R}^{d \times n}$ : Bezier Control Points  
 $m \in \mathbb{N}$ : Reduction Degree  
 $\varepsilon > 0$ : Approximation Tolerance

**Output:**  $T = [t_0, \dots, t_k]$ : Split Intervals  
 $\mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ : List of Reduced Control Points where  $B_{\mathbf{Q}_{m,i}}([0, 1])$  approximates  $B_{\mathbf{P}_n}([t_{i-1}, t_i])$

---

```

1:  $k \leftarrow 1$                                 // Segment Counter
2:  $T \leftarrow [0, 1]$                           // Initial Partition
3: while  $k < \text{length}(T)$  do
4:    $\mathbf{P}_{n,k} \leftarrow \text{Reparametrize}(\mathbf{P}_n, [t_{k-1}, t_k])$ 
5:    $\mathbf{Q}_{m,k} \leftarrow \text{DegreeReduction}(\mathbf{P}_{n,k}, m)$ 
6:   if  $\text{BezierDistance}(\mathbf{P}_{n,k}, \mathbf{Q}_{m,k} \mathbf{E}(m, n)) > \varepsilon$  then
7:      $T \leftarrow [t_0, \dots, t_{k-1}, \frac{t_{k-1}+t_k}{2}, t_k]$ 
8:   else
9:      $k \leftarrow k + 1$ 
10:  $k \leftarrow \text{length}(T) - 1$                   // Number of Segments
11: return  $T, \mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$ 

```

---

reparametrization and degree reduction as

$$\mathbf{P}_{n,i} = \text{Reparameterize}(\mathbf{P}_i, [t_{i-1}, t_i]) \quad (43)$$

$$\mathbf{Q}_{m,i} = \text{DegreeReduction}(\mathbf{P}_{n,i}, m). \quad (44)$$

Hence, one can measure the quality of approximating  $B_{p_0, i, \dots, p_n, i}(t)$  by  $B_{q_0, i, \dots, q_m, i}(t)$  using a Bézier metric  $d_B$  (Definition 2) and degree elevation (Proposition 4) as

$$\begin{aligned} \text{BezierDistance}(\mathbf{P}_{n,i}, \mathbf{Q}_{m,i} \mathbf{E}(m, n)) \\ := d_B(B_{p_0, i, \dots, p_n, i}, B_{q_0, i, \dots, q_m, i} \mathbf{E}(m, n)). \end{aligned}$$

Accordingly, in Algorithm 2 and Algorithm 3, respectively, we present a linear- and a binary-search approach for automatically finding a proper partition  $T = [t_0, \dots, t_k]$  of the unit interval (and the associated control points  $\mathbf{Q}_{m,1}, \dots, \mathbf{Q}_{m,k}$  of local Bézier segments) where the distance between the actual curve and its degree reduction is below a certain desired approximation tolerance  $\varepsilon > 0$ . Note that linear search assumes uniform partitions of the unit interval whereas binary search might result in a nonuniform partition of the unit interval depending on the shape of the input Bézier curve. As a result, as illustrated in Fig. 7, binary search often achieves the same level of approximation quality as linear search by using a significantly less number of Bézier segments. Another important observation in Fig. 7 is that although uniform matching still outperforms least squares and Taylor approximations, the quality of adaptive Bézier approximation is less dependent on the choice of a reduction method. Finally, as expected, the required number of Bézier segments in-

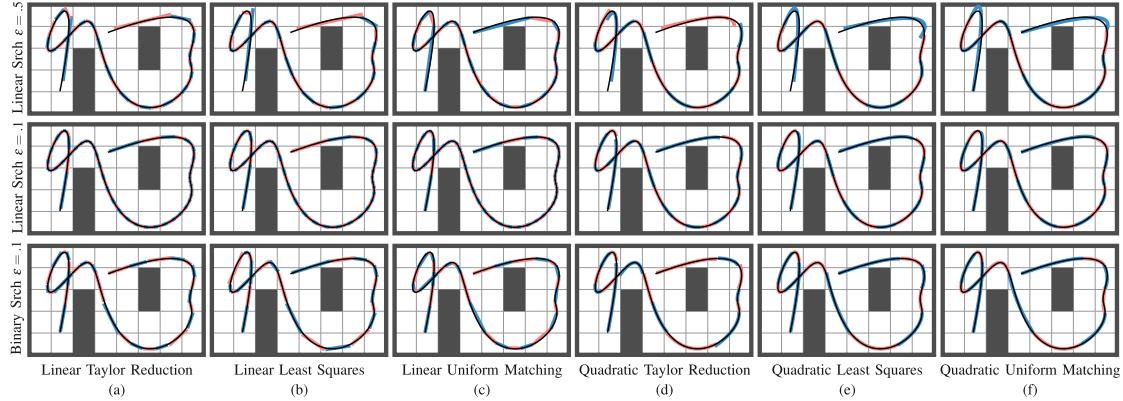


Fig. 7. Adaptive approximation of a 8<sup>th</sup>-order Bézier curve (black dashed line) by (a, b, c) linear and (d, e, f) quadratic Bézier curve segments (red and blue patches) that are automatically obtained via (top, middle) linear and (bottom) binary search based on (a, d) Taylor approximation, (b, e) least squares reduction, and (c, f) uniform matching reduction. The Bézier splitting is automatically done based on a desired approximation tolerance  $\varepsilon$  specified in term of the Bézier maximum control-point distance: (top)  $\varepsilon = 0.5$  units, (middle, bottom)  $\varepsilon = 0.1$  units.

creases exponentially with the increasing approximation quality, which is further discussed in the following Section VI.

## VI. NUMERICAL ANALYSIS OF BÉZIER APPROXIMATIONS

In this section, we provide numerical evidence to show the effectiveness of uniform matching reduction over least squares and Taylor reductions by investigating how Bézier approximation accuracy depends on the number of curve segments. We also demonstrate how the automatically adjusted number of curve segments in adaptive Bézier approximation depends on Bézier degree and approximation tolerance.

### A. Approximation Accuracy Versus Number of Segments

To investigate the role of number of segments in approximation accuracy, we consider the following Bézier features:

- 1) *Curve Length*: The arc length of Bézier curves is an essential criterion in optimal motion planning to find motion trajectories that reduce travel distance.
- 2) *Distance-to-Point*: The distance of a Bézier curve to a point is often used in constrained motion planning for determining parameterwise Bézier intersections and the maximum velocity/acceleration along Bézier curves.
- 3) *Distance-to-Line*: The distance of a Bézier curve to a line segment (or a polyline/polygon) is a common distance-to-collision measure in safe motion planning.
- 4) *Maximum Curvature*: Curvature-constrained motion planning of nonholonomic systems requires an efficient computation of maximum curvature of Bézier curves.

The aforementioned Bézier features can be determined analytically only for linear and quadratic Bézier curves. For high-order Bézier curves, we suggest computing these curve features efficiently using Bézier approximations by linear and quadratic Bézier segments. Since these curve features are nonnegative, to determine the approximation quality, we define the normalized approximation error of a Bézier feature using its actual and approximate calculations as

$$\text{Approximation Error} = \frac{|\text{Feature}_{\text{Approx}} - \text{Feature}_{\text{Actual}}|}{\text{Feature}_{\text{Approx}} + \text{Feature}_{\text{Actual}}} \quad (45)$$

where the actual curve features are computed using a dense discrete samples of Bézier curves.

To determine approximation error statistics, we randomly generate Bézier control points that are uniformly distributed over the unit box  $[0, 1] \times [0, 1]$ . For the distance-to-point criterion, we select the origin as the point of interest; and for distance-to-line, we select the horizontal side of the unit box (i.e., the line segment joining the origin  $(0, 0)$  to point  $(1, 0)$ ). In Figs. 8–11, we provide sample statistics (mean and standard deviation) of normalized approximation errors of curve length, distance-to-point, distance-to-line, and maximum curvature versus the number of segments. It is visibly clear that the Bézier approximation with uniform matching reduction achieves significantly better performance in capturing curve length, distance-to-point and distance-to-line compared to the least squares and Taylor approximations. Especially, the end-point preservation property of uniform matching reduction plays a key role for its superior performance for the distance-to-point/line criteria presented in Figs. 9–10. We observe in Fig. 8 that Bézier approximations with linear segments have comparable accuracy for all three reduction methods, which can be explained by the limited representation power of linear curve segments. On the other hand, uniform matching reduction shows a superior performance for Bézier approximations with quadratic segments. Finally, as illustrated in Fig. 11, we see that all Bézier degree reduction methods perform equally well for approximating the maximum curvature of Bézier curves. This can be explained by the limited expressiveness of quadratic segments for approximating the first and second derivatives of Bézier curves since curvature is a function of the first and second curve derivatives.

### B. Number of Segments Versus Bézier Degree and Tolerance

In this part, we numerically study how the number of segments automatically determined in adaptive Bézier approximation depends on the order of the Bézier curve and the approximation tolerance (specified in terms of the maximum control-point distance). We consider randomly generated Bézier control points

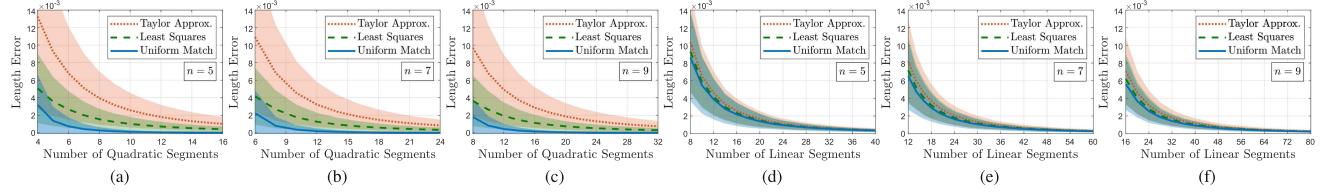


Fig. 8. Normalized length error statistics of approximating the  $n^{\text{th}}$ -order Bézier curves by (a, b, c) quadratic and (d, e, f) linear Bézier segments: (a, d)  $n = 5$ , (b, e)  $n = 7$ , (c, f)  $n = 9$ , where the mean and the standard deviation of the error are presented by the line and a shaded region, respectively.

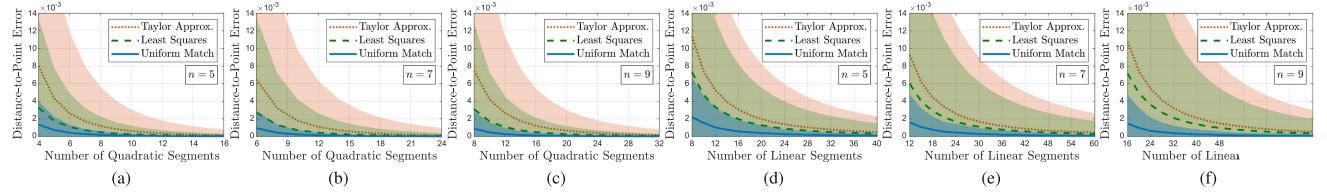


Fig. 9. Normalized distance-to-point error statistics of approximating the  $n^{\text{th}}$ -order Bézier curves by (a, b, c) quadratic and (d, e, f) linear Bézier segments: (a, d)  $n = 5$ , (b, e)  $n = 7$ , (c, f)  $n = 9$ , where the mean and the standard deviation of the error are presented by a line and a shaded region, respectively.

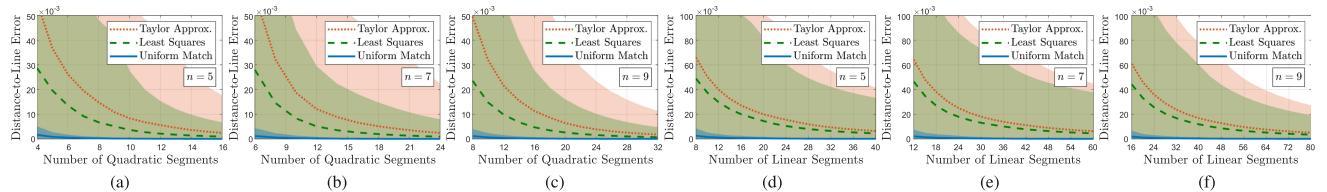


Fig. 10. Normalized distance-to-line error statistics of approximating  $n^{\text{th}}$ -order Bézier curves by (a, b, c) quadratic and (d, e, f) linear Bézier segments: (a, d)  $n = 5$ , (b, e)  $n = 7$ , (c, f)  $n = 9$ , where the mean and the standard deviation of the error are presented by a line and a shaded region, respectively.

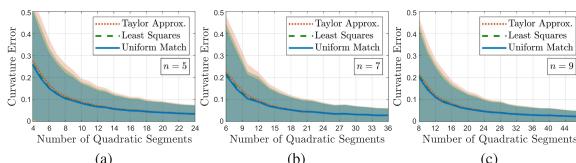


Fig. 11. Normalized maximum curvature error statistics of quadratic approximations of Bézier curves for different number of segments: (a)  $n = 5$ , (b)  $n = 7$ , (c)  $n = 9$ , where the mean and the standard deviation of the error are presented by a line and a shaded region, respectively. Note that for numerical stability we set an upper bound of 1000 units on the maximum curvature, and any sample case with a larger maximum curvature is rejected.

over the unit box  $[0, 1] \times [0, 1]$ . To ensure scale invariance, we rescale Bézier control points to have a sample variance of unity. In Fig. 12, we present the average number of segments used in adaptive approximation of Bézier curves of different orders. For a fixed choice of an approximation tolerance, we observe that the number of segments grows linearly with the Bézier degree for linear search whereas the grow rate is sublinear for binary search. This is strongly aligned with the Bézier approximation rule proposed in Section V-B. Finally, as illustrated in Fig. 13, the average number of segments used in adaptive Bézier approximation grows exponentially with the negated order of magnitude

of approximation tolerance  $\varepsilon$ , because the higher the accuracy the higher the spatial resolution.

## VII. CONCLUSION

In this article, we introduce a novel adaptive Bézier approximation method that automatically splits and performs degree reduction on high-order Bézier curves to approximately represent them by multiple low-order Bézier segments at any given approximation tolerance measured by a Bézier metric. Accordingly, we propose a new maximum control-point distance for efficient and informative comparison of Bézier curves. We show that the maximum control-point distance defines a tight upper bound on standard Bézier metrics such as Hausdorff, parameterwise maximum, and Frobenious-norm distance of Bézier curves and can be used to geometrically bound Bézier curves with respect to each others. To better maintain the original curve shape, we also propose a new parameterwise matching reduction method that allows one to preserve a certain set of curve points (e.g., end points) after degree reduction. The matching reduction shows a superior approximation performance compared to standard least squares and Taylor approximations. Based on the explicit form of degree-one matching reduction error, we also suggest a rule of thumb for approximating the  $n^{\text{th}}$ -order Bézier curves by  $3(n - 1)$  quadratic and  $6(n - 1)$  linear Bézier

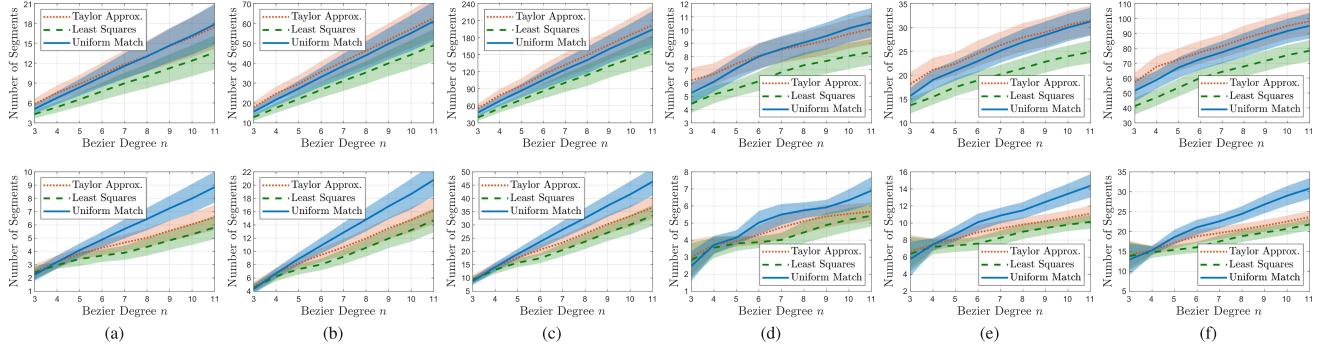


Fig. 12. Relation between the number of segments and Bézier degree in adaptive Bézier approximation by (top) linear and (bottom) quadratic segments using (a, b, c) linear and (d, e, f) binary search. Here, the approximation tolerance is measured by the maximum control-point distance and set to be (a, d) 0.1, (b, e) 0.01, and (c, f) 0.001 units.

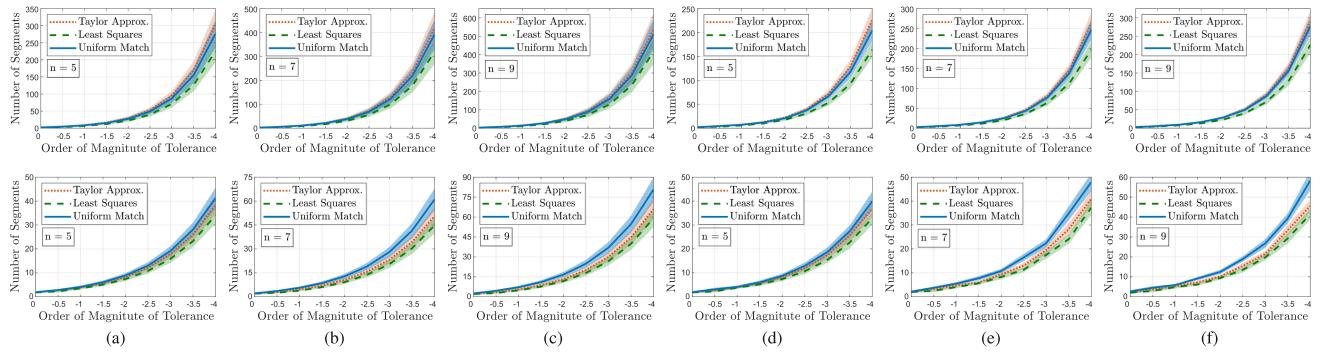


Fig. 13. Relation between the number of segments and approximation tolerance in adaptive Bézier approximation by (top) linear and (bottom) quadratic segments using (a, b, c) linear and (d, e, f) binary search for Bézier order (a, d)  $n = 5$ , (b, e)  $n = 7$ , and (c, f)  $n = 9$ . Here, the approximation tolerance is measured by the maximum control-point distance.

segments. Our extensive numerical studies demonstrate the effectiveness of the proposed methods and the validity of our Bezier approximation rule. Work now in progress targets applying these Bézier approximation tools in sensor-based reactive motion planning and trajectory optimization of nonholonomically constrained mobile robots and autonomous vehicles [55].

## APPENDIX A POLYNOMIAL BASIS TRANSFORMATION MATRICES

In this part, we provide the explicit formulas for the elements of polynomial basis transformation matrices.

**Lemma 5 ([43]) (Monomial and Bernstein Basis Transformation):** The transformation matrices between monomial and Bernstein basis vectors, i.e.,

$$\mathbf{m}_n(t) = \mathbf{T}_b^m(n)b_n(t) \quad (46a)$$

$$b_n(t) = \mathbf{T}_m^b(n)\mathbf{m}_n(t) \quad (46b)$$

are explicitly given by<sup>9</sup>

$$[\mathbf{T}_m^b(n)]_{i+1,j+1} = \begin{cases} (-1)^{(j-i)} \binom{n}{j} \binom{j}{i}, & \text{if } i \leq j \\ 0, & \text{otherwise} \end{cases} \quad (47a)$$

$$[\mathbf{T}_b^m(n)]_{i+1,j+1} = \begin{cases} \binom{j}{i}, & \text{if } i \leq j \\ 0, & \text{otherwise} \end{cases} \quad (47b)$$

<sup>9</sup>The transformation matrices between monomial and Bernstein bases are upper triangular with positive diagonal elements and so are invertible.

where  $i, j \in [0, 1, \dots, n]$ , and they are the inverse of each other

$$\mathbf{T}_b^m(n)^{-1} = \mathbf{T}_m^b(n). \quad (48)$$

**Lemma 6 ([56]) (Monomial-Taylor Basis Transformation):** The transformation between monomial and Taylor bases, i.e.,

$$\mathbf{m}_n(t) = \mathbf{T}_\tau^m(n, t_o)\tau_{n,t_o}(t) \quad (49a)$$

$$\tau_{n,t_o}(t) = \mathbf{T}_m^\tau(n, t_o)\mathbf{m}_n(t) \quad (49b)$$

are explicitly given by<sup>10</sup>

$$[\mathbf{T}_m^\tau(n, t_o)]_{i+1,j+1} = \begin{cases} \binom{i}{j}(-t_o)^{i-j}, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases} \quad (50a)$$

$$[\mathbf{T}_\tau^m(n, t_o)]_{i+1,j+1} = \begin{cases} \binom{i}{j}(t_o)^{i-j}, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases} \quad (50b)$$

where  $i, j \in \{0, \dots, n\}$ , and they are the inverse of each other

$$\mathbf{T}_\tau^m(n, t_o)^{-1} = \mathbf{T}_m^\tau(n, t_o). \quad (51)$$

Accordingly, the transformation matrices between the Bernstein and Taylor bases, i.e.,

$$\tau_{n,t_o}(t) = \mathbf{T}_b^\tau(n, t_o)b_n(t) \quad (52a)$$

$$b_n(t) = \mathbf{T}_\tau^b(n, t_o)\tau_{n,t_o}(t) \quad (52b)$$

can be determined using the monomial basis as

$$\mathbf{T}_b^\tau(n, t_o) = \mathbf{T}_b^m(n)\mathbf{T}_m^\tau(n, t_o) \quad (53a)$$

<sup>10</sup>The transformation matrices between monomial and Taylor bases are lower triangular with all ones in the main diagonal and so are invertible.

$$\mathbf{T}_\tau^b(n, t_o) = \mathbf{T}_\tau^m(n, t_o) \mathbf{T}_m^b(n) \quad (53b)$$

where  $\mathbf{T}_b^\tau(n, t_o)^{-1} = \mathbf{T}_\tau^b(n, t_o)$ .

## APPENDIX B ON REPARAMETRIZATION OF POLYNOMIAL CURVES

In this part, we show how affine reparametrization of polynomial curves can be performed explicitly via Taylor basis.

*Lemma 7 ([56]):* The Bernstein, monomial and Taylor basis vectors of degree  $n$  (associated with a Taylor offset  $t_o \in \mathbb{R}$ ) can be affinely reparametrized from interval  $[a, b]$  to  $[c, d]$  (with  $a < b$  and  $c < d$ ) as

$$\begin{aligned} b_n(t) &= \mathbf{T}_\tau^b(n, \hat{t}_o) \text{diag}(\mathbf{m}_n(\frac{b-a}{d-c})) \mathbf{T}_b^\tau(n, t_o) b_n(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}) \\ m_n(t) &= \mathbf{T}_\tau^m(n, \hat{t}_o) \text{diag}(\mathbf{m}_n(\frac{b-a}{d-c})) \mathbf{T}_m^\tau(n, t_o) m_n(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}) \\ \tau_{n, \hat{t}_o}(t) &= \text{diag}\left(\mathbf{m}_n(\frac{b-a}{d-c})\right) \tau_{n, t_o}\left(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}\right) \end{aligned}$$

where  $\text{diag}$  denotes the diagonal matrix with diagonal entries specified with its argument, and the reparametrized Taylor offset is given by the associated affine transformation as

$$\hat{t}_o = \frac{d-c}{b-a} t_a - \frac{ad-bc}{b-a}. \quad (54)$$

*Lemma 8 ([56]): (Polynomial Curve Reparametrization)* Bézier, monomial, and Taylor curves of degree  $n \in \mathbb{N}$  with respective control point matrices  $\mathbf{P}_n = [p_0, \dots, p_n]$ ,  $\mathbf{Q}_n = [q_0, \dots, q_n]$ , and  $\mathbf{Y}_n = [y_0, \dots, y_n]$  (and a Taylor offset  $t_o \in \mathbb{R}$ ) can be affinely reparametrized from interval  $[a, b]$  to  $[c, d]$  (with  $a < b$  and  $c < d$ ) as

$$B_{\hat{p}_0, \dots, \hat{p}_n}(t) = B_{p_0, \dots, p_n}(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}) \quad (55a)$$

$$M_{\hat{q}_0, \dots, \hat{q}_n}(t) = M_{q_0, \dots, q_n}(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}) \quad (55b)$$

$$\Upsilon_{\hat{y}_0, \dots, \hat{y}_n}(t, \hat{t}_o) = \Upsilon_{y_0, \dots, y_n}(\frac{b-a}{d-c}t + \frac{ad-bc}{d-c}, t_o) \quad (55c)$$

with the corresponding reparametrized control point matrices  $\hat{\mathbf{P}}_n = [\hat{p}_0, \dots, \hat{p}_n]$ ,  $\hat{\mathbf{Q}}_n = [\hat{q}_0, \dots, \hat{q}_n]$ , and  $\hat{\mathbf{Y}}_n = [\hat{y}_0, \dots, \hat{y}_n]$  that are given by

$$\hat{\mathbf{P}}_n = \mathbf{P}_n \mathbf{T}_\tau^b(n, t_o) \text{diag}\left(\mathbf{m}_n(\frac{d-c}{b-a})\right) \mathbf{T}_b^\tau(n, \hat{t}_o) \quad (56a)$$

$$\hat{\mathbf{Q}}_n = \mathbf{Q}_n \mathbf{T}_\tau^m(n, t_o) \text{diag}\left(\mathbf{m}_n(\frac{d-c}{b-a})\right) \mathbf{T}_m^\tau(n, \hat{t}_o) \quad (56b)$$

$$\hat{\mathbf{Y}}_n = \mathbf{Y}_n \text{diag}\left(\mathbf{m}_n(\frac{d-c}{b-a})\right) \quad (56c)$$

where  $\hat{t}_o = \frac{d-c}{b-a} t_o - \frac{ad-bc}{b-a}$ .

## APPENDIX C MATCHING REDUCTION IN MONOMIAL BASIS

The matching reduction matrix can be explicitly computed using the monomial basis.

*Lemma 9 ([56]) (Matching Reduction in Monomial Basis):* For any  $n \geq m \in \mathbb{N}$  and distinct  $t_0, \dots, t_m \in \mathbb{R}$ , the parameterwise matching reduction matrix  $\mathbf{R}_{t_0, \dots, t_m}(n, m)$  can be

computed using monomial basis as

$$\begin{aligned} \mathbf{R}_{t_0, \dots, t_m}(n, m) &= \mathbf{T}_m^b(n) \mathbf{m}_n(t_0, \dots, t_m) \times \mathbf{m}_m(t_0, \dots, t_m)^{-1} \mathbf{T}_b^m(m) \\ &= \mathbf{T}_m^b(n) \begin{bmatrix} \mathbf{I}_{(m+1) \times (m+1)} \\ \alpha_1(t_0, \dots, t_m) \\ \vdots \\ \alpha_{n-m}(t_0, \dots, t_m) \end{bmatrix} \mathbf{T}_b^m(m) \end{aligned}$$

with row vectors  $\alpha_i(t_0, \dots, t_m) = [\alpha_{i,0}, \dots, \alpha_{i,m}]$  that are recursively defined as

$$\alpha_{i+1,k} = \alpha_{i,k-1} + \alpha_{i,m} \alpha_{1,k}$$

where base conditions of  $\alpha_{i,-1} = 0$  and  $\alpha_{1,k}$  satisfying

$$t^{m+1} - \sum_{k=0}^m \alpha_{1,k} t^k = \prod_{k=0}^m (t - t_k).$$

## APPENDIX D PROOFS

### A. Proof of Lemma 1

*Proof:* The result follows from that  $n^{\text{th}}$ -order Bernstein polynomials, as well as monomials and Taylor polynomials of degree less than or equal to  $n$ , define a basis of  $n+1$  linearly independent polynomials for the  $n^{\text{th}}$ -order polynomials [43].

Alternatively, one can verify the result using polynomial basis transformations as follows. The monomial basis matrix  $\mathbf{m}_n(t_0, \dots, t_n)$ , by definition, equals to the Vandermonde matrix, which is nonsingular for distinct  $t_0, \dots, t_n$  [54]. The Bezier and Taylor basis matrices are also nonsingular due to the change of basis relation, i.e.,

$$\mathbf{b}_n(t_0, \dots, t_n) = \mathbf{T}_m^b(n) \mathbf{m}_n(t_0, \dots, t_n) \quad (57a)$$

$$\tau_{n, t_o}(t_0, \dots, t_n) = \mathbf{T}_\tau^b(n, t_o) \mathbf{m}_n(t_0, \dots, t_n) \quad (57b)$$

where  $\mathbf{T}_m^b(n)$  and  $\mathbf{T}_\tau^b(n, t_o)$  are invertible triangular basis transformation matrices (see Lemmas 5 & 6). ■

### B. Proof of Lemma 2

*Proof:* Consider the basis transformation matrix  $\mathbf{T}_m^b(n)$  from monomial to Bernstein basis. It follows by definition that

$$\mathbf{b}_n(t_0, \dots, t_n) = \mathbf{T}_m^b(n) \mathbf{m}_n(t_0, \dots, t_n). \quad (58)$$

Since the monomial basis matrix  $\mathbf{m}_n(t_0, \dots, t_n)$  is invertible for any distinct  $t_0, \dots, t_n \in \mathbb{R}$  (Lemma 1), we obtain

$$\mathbf{T}_m^b(n) = \mathbf{b}_n(t_0, \dots, t_n) \mathbf{m}_n(t_0, \dots, t_n)^{-1}. \quad (59)$$

Similarly, the result can be verified for any change of basis between Bernstein, Taylor, and monomial bases. ■

### C. Proof of Lemma 3

*Proof:* Let us focus on the equivalence of Bézier curves to monomial curves. The equivalence of Bézier and monomial curves means that for any distinct  $t_0, \dots, t_n \in \mathbb{R}$  one has

$$\mathbf{B}_{p_0, \dots, p_n}(t_0, \dots, t_n) = \mathbf{M}_{p_0, \dots, p_n}(t_0, \dots, t_n) \quad (60)$$

$$\mathbf{P}_n \mathbf{b}_n(t_0, \dots, t_n) = \mathbf{Q}_n \mathbf{m}_n(t_0, \dots, t_n) \quad (61)$$

$$= \mathbf{Q}_n \mathbf{T}_b^m(n) \mathbf{b}_n(t_0, \dots, t_n). \quad (62)$$

Since the Bernstein basis matrix  $\mathbf{b}_n(t_0, \dots, t_n)$  is invertible for any distinct  $t_0, \dots, t_n$  (Lemma 1), one can conclude that

$$\mathbf{P}_n = \mathbf{Q}_n \mathbf{T}_b^m(n) \quad (63)$$

which can be similarly extended for other polynomial curve equivalence relations. ■

#### D. Proof of Lemma 4

*Proof:* For any  $t_0, \dots, t_n \in \mathbb{R}$ , by definition, affine Bezier reparametrization satisfies for  $\hat{t}_i = \frac{d-c}{b-a}t_i - \frac{ad-bc}{b-a}$  that

$$\hat{\mathbf{P}}_n \mathbf{b}_n(\hat{t}_0, \dots, \hat{t}_n) = \mathbf{P}_n \mathbf{b}_n(t_0, \dots, t_n). \quad (64)$$

Hence, we have the result since Bernstein basis matrices are invertible for any distinct parameters (Lemma 1), which also extends in a similar way to Taylor and monomial curves. ■

#### E. Proof of Proposition 1

*Proof:* Using the following properties of Bernstein polynomials [44]:

$$\int_0^1 b_{i,n}(t) dt = \frac{1}{n+1}, \text{ and } b_{i,n}(t) b_{j,m}(t) = \frac{\binom{n}{i} \binom{n}{j}}{\binom{n+m}{i+j}} b_{i+j, n+m}(t)$$

one can verify the result as

$$\begin{aligned} d_{L2}(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n})^2 &= \int_0^1 \|\mathbf{B}_{p_0-q_0, \dots, p_n-q_n}(t)\|^2 dt \\ &= \sum_{i=0}^n \sum_{j=0}^n (\mathbf{p}_i - \mathbf{q}_i)^T (\mathbf{p}_j - \mathbf{q}_j) \int_0^1 b_{i,n}(t) b_{j,n}(t) dt \\ &= \sum_{i=0}^n \sum_{j=0}^n (\mathbf{p}_i - \mathbf{q}_i)^T (\mathbf{p}_j - \mathbf{q}_j) \frac{\binom{n}{i} \binom{n}{j}}{\binom{2n}{i+j}} \int_0^1 b_{i+j, 2n}(t) dt \\ &= \sum_{i=0}^n \sum_{j=0}^n \frac{1}{2n+1} \frac{\binom{n}{i} \binom{n}{j}}{\binom{2n}{i+j}} (\mathbf{p}_i - \mathbf{q}_i)^T (\mathbf{p}_j - \mathbf{q}_j) \\ &= \text{tr} \left( (\mathbf{P}_n - \mathbf{Q}_n) \mathbf{W}_n (\mathbf{P}_n - \mathbf{Q}_n)^T \right). \end{aligned}$$

■

#### F. Proof of Proposition 2

*Proof:* By Definition 5, the Bézier parameterwise-maximum distance defines an upper bound on the Bézier Haussdoff distance, i.e.,

$$d_H(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}) \leq d_M(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}). \quad (65)$$

Similarly, the equivalence relation of the Frobenius distance and the control-point distances is evident from Definition 4 and Definition 6 as

$$\begin{aligned} d_C(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}) &= \max_{i=0, \dots, n} \|\mathbf{p}_i - \mathbf{q}_i\| \\ &\leq d_F(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}) \\ &= \sqrt{\sum_{i=0}^n \|\mathbf{p}_i - \mathbf{q}_i\|^2} \\ &\leq \sqrt{n} d_C(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}). \end{aligned}$$

Hence, the result follows from Jensen's equality for the squared Euclidean distance because

$$\begin{aligned} \|\mathbf{B}_{p_0, \dots, p_n}(t) - \mathbf{B}_{q_0, \dots, q_n}(t)\|^2 &= \|\mathbf{B}_{p_0-q_0, \dots, p_n-q_n}(t)\|^2 \\ &= \left\| \sum_{i=0}^n b_{i,n}(t)(\mathbf{p}_i - \mathbf{q}_i) \right\|^2 \leq \sum_{i=0}^n b_{i,n}(t) \|\mathbf{p}_i - \mathbf{q}_i\|^2 \\ &\leq \max_i \|\mathbf{p}_i - \mathbf{q}_i\|^2. \end{aligned}$$

Note that the Bernstein polynomials sum to one over  $t \in [0, 1]$ , i.e.,  $\sum_{i=0}^n b_{i,n}(t) = 1$  for all  $t \in [0, 1]$  (Property 3). ■

#### G. Proof of Proposition 3

*Proof:* The ordering relation of Bézier metrics in Proposition 2 and the convexity of Bézier curves in Property 3 imply that

$$\begin{aligned} \min_{t' \in [0,1]} \|\mathbf{B}_{p_0, \dots, p_n}(t) - \mathbf{B}_{q_0, \dots, q_n}(t')\| \\ \leq \|\mathbf{B}_{p_0, \dots, p_n}(t) - \mathbf{B}_{q_0, \dots, q_n}(t)\| \\ \leq d_M(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}) \leq d_C(\mathbf{B}_{p_0, \dots, p_n}, \mathbf{B}_{q_0, \dots, q_n}) \end{aligned}$$

and so the result follows. ■

#### H. Proof of Proposition 4

*Proof:* The sufficiency of elevated control points in (25) can be verified as

$$\mathbf{B}_{q_0, \dots, q_m}(t) = \mathbf{Q}_m \mathbf{b}_m(t) \quad (66a)$$

$$= \mathbf{P}_n \mathbf{E}(n, m) \mathbf{b}_m(t) \quad (66b)$$

$$= \mathbf{P}_n \mathbf{T}_m^b(n) \mathbf{I}_{(n+1) \times (m+1)} \mathbf{T}_b^m(m) \mathbf{b}_m(t) \quad (66c)$$

$$= \mathbf{P}_n \mathbf{T}_m^b(n) \mathbf{I}_{(n+1) \times (m+1)} \mathbf{m}_m(t) \quad (66d)$$

$$= \mathbf{P}_n \mathbf{T}_m^b(n) \mathbf{m}_n(t) \quad (66e)$$

$$= \mathbf{P}_n \mathbf{b}_n(t) = \mathbf{B}_{p_0, \dots, p_n}(t). \quad (66f)$$

To show the necessity of parameterwise coincidence, consider distinct parameters  $t_0, \dots, t_m \in \mathbb{R}$  with  $t_i \neq t_j$  for all  $i \neq j$ . Since square Bernstein basis matrices of distinct parameters are invertible (Lemma 1), using the coinciding curve points at  $t_0, \dots, t_m$ , i.e.,

$$\mathbf{P}_n \mathbf{b}_n(t_0, \dots, t_m) = \mathbf{Q}_m \mathbf{b}_m(t_0, \dots, t_m) \quad (67)$$

one can obtain an explicit expression for  $\mathbf{E}(n, m)$  as

$$\mathbf{Q}_m = \mathbf{P}_n \mathbf{b}_n(t_0, \dots, t_m) \mathbf{b}_m(t_0, \dots, t_m)^{-1} \quad (68)$$

$$= \mathbf{P}_n \mathbf{E}(n, m) \quad (69)$$

which can be further simplified using the Bernstein-to-monomial basis transformation as

$$\begin{aligned} \mathbf{E}(n, m) &= \mathbf{b}_n(t_0, \dots, t_m) \mathbf{b}_m(t_0, \dots, t_m)^{-1} \\ &= \mathbf{T}_m^b(n) \mathbf{m}_n(t_0, \dots, t_m) \mathbf{m}_m(t_0, \dots, t_m)^{-1} \mathbf{T}_b^m(m) \\ &= \mathbf{T}_m^b(n) \mathbf{I}_{(n+1) \times (m+1)} \mathbf{T}_b^m(m). \end{aligned}$$

■

### I. Proof of Proposition 5

*Proof:* By definition in (26), we have

$$\mathbf{E}(n, m)\mathbf{b}_m(t) = \mathbf{T}_m^b(n)\mathbf{I}_{(n+1) \times (m+1)}\mathbf{T}_b^m(m)\mathbf{b}_m(t) \quad (70)$$

$$= \mathbf{T}_m^b(n)\mathbf{I}_{(n+1) \times (m+1)}\mathbf{m}_m(t) \quad (71)$$

$$= \mathbf{T}_m^b(n)\mathbf{m}_n(t) = \mathbf{b}_n(t). \quad (72)$$

Therefore, the result follows since the Bernstein basis matrix  $\mathbf{b}_m(t_0, \dots, t_m)$  is invertible (Lemma 1).  $\blacksquare$

### J. Proof of Proposition 6

*Proof:* We below provide a proof by induction.

- Base Case: ( $n \leq m \leq n+1$ ): If  $m = n$ , then one trivially has  $\mathbf{E}(n, n) = \mathbf{I}_{(n+1) \times (n+1)}$ . If  $m = n+1$ , then

$$[\mathbf{E}(n, n+1)]_{i+1, j+1} = \begin{cases} 1 - \frac{j}{n+1}, & \text{if } j = i \\ \frac{j}{n+1}, & \text{if } j = i+1 \\ 0, & \text{otherwise} \end{cases} \quad (73)$$

which follows from (27) and the following degree-one elevation property of Bernstein polynomials [43]:

$$b_{i,n}(t) = \left(1 - \frac{i}{n+1}\right) b_{i,n+1}(t) + \frac{i+1}{n+1} b_{i+1,n+1}(t).$$

Also note that  $\binom{n}{j}/\binom{n+1}{j} = 1 - \frac{j}{n+1}$  and  $\binom{n}{j-1}/\binom{n+1}{j} = \frac{j}{n+1}$ . Hence, the result holds for the base case.

- Induction Step ( $m > n+1$ ): Suppose the results holds for  $\mathbf{E}(n, m-1)$ , then one can determine  $\mathbf{E}(n, m)$  as

$$\mathbf{E}(n, m) = \mathbf{E}(n, m-1)\mathbf{E}(m-1, m) \quad (74)$$

because the degree elevation operation preserves the original Bézier curve exactly. Hence, it follows from (72) that

$$\begin{aligned} [\mathbf{E}(n, m)]_{i+1, j+1} &= \sum_{k=0}^m [\mathbf{E}(n, m-1)]_{i+1, k+1} \\ &\quad \times [\mathbf{E}(m-1, m)]_{k+1, j+1} \\ &= [\mathbf{E}(n, m-1)]_{i+1, j} [\mathbf{E}(m-1, m)]_{j, j+1} \\ &\quad + [\mathbf{E}(n, m-1)]_{i+1, j+1} [\mathbf{E}(m-1, m)]_{j+1, j+1} \\ &= \frac{j}{m} [\mathbf{E}(n, m-1)]_{i+1, j} + \left(1 - \frac{j}{m}\right) [\mathbf{E}(n, m-1)]_{i+1, j+1}. \end{aligned}$$

Note that  $[\mathbf{E}(n, m-1)]_{i+1, j+1} \neq 0$  iff  $0 \leq j-i \leq m-n-1$ ; and  $[\mathbf{E}(n, m-1)]_{i+1, j} \neq 0$  iff  $1 \leq j-i \leq m-n$ . Hence, we complete the induction step by checking the following cases:

- 1) If  $j-i > m-n$  or  $j-i < 0$ , then  $[\mathbf{E}(n, m-1)]_{i+1, j+1} = 0$  and  $[\mathbf{E}(n, m-1)]_{i+1, j} = 0$ , and so

$$[\mathbf{E}(n, m)]_{i+1, j+1} = 0. \quad (75)$$

- 2) If  $j-i = m-n$ , then  $[\mathbf{E}(n, m-1)]_{i+1, j+1} = 0$  and so

$$\begin{aligned} [\mathbf{E}(n, m)]_{i+1, j+1} &= \frac{j}{m} [\mathbf{E}(n, m-1)]_{i+1, j} \\ &= \frac{j}{m} \frac{\binom{n}{i}}{\binom{m-1}{j-1}} = \frac{\binom{n}{i}}{\binom{m}{j}}. \quad (76) \end{aligned}$$

- 3) If  $j-i = 0$ , then  $[\mathbf{E}(n, m-1)]_{i+1, j} = 0$  and so

$$[\mathbf{E}(n, m)]_{i+1, j+1} = \left(1 - \frac{j}{m}\right) [\mathbf{E}(n, m-1)]_{i+1, j+1} \quad (77)$$

$$= \left(1 - \frac{j}{m}\right) \frac{\binom{n}{i}}{\binom{m-1}{j}} = \frac{\binom{n}{i}}{\binom{m}{j}}. \quad (78)$$

- 4) Otherwise (i.e.,  $0 < j-i < m-n$ ), we have

$$\begin{aligned} [\mathbf{E}(n, m)]_{i+1, j+1} &= \frac{j}{m} [\mathbf{E}(n, m-1)]_{i+1, j} \\ &\quad + \left(1 - \frac{j}{m}\right) [\mathbf{E}(n, m-1)]_{i+1, j+1} \\ &= \frac{j}{m} \frac{\binom{n}{i} \binom{m-n-1}{j-i-1}}{\binom{m-1}{j-1}} + \left(1 - \frac{j}{m}\right) \frac{\binom{n}{i} \binom{m-n-1}{j-i}}{\binom{m-1}{j}} \quad (79) \\ &= \frac{\binom{n}{i} \binom{m-n-1}{j-i-1}}{\frac{m}{j} \binom{m-1}{j-1}} + \frac{\binom{n}{i} \binom{m-n-1}{j-i}}{\frac{m-j}{m-j} \binom{m-1}{j}} \quad (80) \\ &= \frac{\left(\frac{n}{i}\right)}{\left(\frac{m}{j}\right)} \left( \binom{m-n-1}{j-i-1} + \binom{m-n-1}{j-i} \right) \quad (81) \\ &= \frac{\binom{n}{i} \binom{m-n}{j-i}}{\left(\frac{m}{j}\right)}. \quad (82) \end{aligned}$$

### K. Proof of Proposition 7

*Proof:* The result can be verified using either (26) or (28) with the fact that if a square matrix  $\mathbf{B}$  is full rank (i.e., invertible), then  $\text{rank}(\mathbf{AB}) = \text{rank}(\mathbf{A})$  for any matrix  $\mathbf{A}$  that is conformable for the multiplication  $\mathbf{AB}$  [57].  $\blacksquare$

### L. Proof of Proposition 8

*Proof:* The column-sum property of the elevation matrix follows from Proposition 5

$$\mathbf{E}(n, m) = \mathbf{b}_n(t_0, \dots, t_m)\mathbf{b}_m(t_0, \dots, t_m)^{-1} \quad (83)$$

and the convexity of Bernstein polynomials (Property 3)

$$\begin{aligned} \mathbf{1}_{1 \times (n+1)}\mathbf{b}_n(t_0, \dots, t_m) &= \mathbf{1}_{1 \times (m+1)} \\ \mathbf{1}_{1 \times (m+1)}\mathbf{b}_m(t_0, \dots, t_m) &= \mathbf{1}_{1 \times (m+1)}\mathbf{b}_m(t_0, \dots, t_m)^{-1} \\ &= \mathbf{1}_{1 \times (m+1)} \end{aligned}$$

where  $t_0, \dots, t_m \in \mathbb{R}$  are any distinct reals.

The row-sum property of the elevation matrix can be proven by induction as follows.

- 1) Base Case ( $n \leq m \leq n+1$ ): If  $m = n$ , then one has  $\mathbf{E}(n, n) = \mathbf{I}_{(n+1) \times (n+1)}$  and so the result holds. For  $m = n+1$ ,

$$[\mathbf{E}(n, n+1)]_{i+1, j+1} = \begin{cases} 1 - \frac{i}{n+1}, & \text{if } j = i \\ \frac{i+1}{n+1}, & \text{if } j = i+1 \\ 0, & \text{otherwise.} \end{cases}$$

Hence, the row sum of  $\mathbf{E}(n, n+1)$  is  $\frac{n+2}{n+1}$ , i.e.,

$$\sum_{j=0}^{n+1} [\mathbf{E}(n, n+1)]_{i+1, j+1} = 1 - \frac{i}{n+1} + \frac{i+1}{n+1} = \frac{n+2}{n+1}.$$

- 2) Induction ( $m > n + 1$ ). Suppose that the result holds for  $\mathbf{E}(n, m - 1)$ . Hence, using  $\mathbf{E}(n, m) = \mathbf{E}(n, m - 1)\mathbf{E}(m - 1, m)$ , one can conclude that the row sum of multiplication of two matrices is the multiplication of the their row sums, i.e.,

$$\begin{aligned} & \sum_{j=0}^m [\mathbf{E}(n, m)]_{i+1, j+1} \\ &= \sum_{j=0}^m \sum_{k=0}^{m-1} [\mathbf{E}(n, m - 1)]_{i+1, k+1} [\mathbf{E}(m - 1, m)]_{k+1, j+1} \\ &= \left( \sum_{k=0}^{m-1} [\mathbf{E}(n, m - 1)]_{i+1, k+1} \right) \left( \sum_{j=0}^m [\mathbf{E}(m - 1, m)]_{k+1, j+1} \right) \\ &= \frac{m}{n+1} \frac{m+1}{m} = \frac{m+1}{n+1}. \end{aligned}$$

■

### M. Proof of Proposition 9

*Proof:* The result directly follows from Definition 3 of the L2-norm distance because degree elevation exactly represents Bézier curves with more control points (Definition 7). ■

### N. Proof of Proposition 10

*Proof:* Using the column- and row-sum property of the elevation matrix in Proposition 8, one can obtain the result by applying Jensen's inequality as

$$\begin{aligned} d_F(\mathbf{B}_{[p_0, \dots, p_n]}\mathbf{E}(n, m), \mathbf{B}_{[q_0, \dots, q_n]}\mathbf{E}(n, m)) &= \|(\mathbf{P}_n - \mathbf{Q}_n)\mathbf{E}(n, m)\|_F^2 \\ &= \sum_{j=0}^m \|(\mathbf{P}_n - \mathbf{Q}_n)\mathbf{E}(n, m)_{j+1}\|^2 \\ &\leq \sum_{j=0}^m \sum_{i=0}^n [\mathbf{E}(n, m)]_{i+1, j+1} \|p_i - q_i\|^2 \\ &= \sum_{i=0}^n \|p_i - q_i\|^2 \sum_{j=0}^m [\mathbf{E}(n, m)]_{i+1, j+1} \\ &= \frac{m+1}{n+1} d_F(\mathbf{B}_{[p_0, \dots, p_n]}, \mathbf{B}_{[q_0, \dots, q_n]})^2 \end{aligned}$$

where  $\mathbf{E}(n, m)_{j+1}$  denotes the  $(j+1)^{\text{th}}$  column of  $\mathbf{E}(n, m)$ . ■

### O. Proof of Proposition 11

*Proof:* Let  $\mathbf{E}(n, m) = [e_0, \dots, e_m]$ . Then, the result can be verified using Jensen's inequality and the unit column sum property of the elevation matrix (Proposition 8) as follows:

$$\begin{aligned} d_C(\mathbf{B}_{[p_0, \dots, p_n]}\mathbf{E}(n, m), \mathbf{B}_{[q_0, \dots, q_n]}\mathbf{E}(n, m)) &= \max_{i=0 \dots m} \|(\mathbf{P} - \mathbf{Q})e_i\| \\ &\leq \max_{j=0, \dots, n} \|p_j - q_j\| = d_C(\mathbf{B}_{[p_0, \dots, p_n]}, \mathbf{B}_{[q_0, \dots, q_n]}) \end{aligned}$$

where the Jensen's inequality and  $\mathbf{1}^T e_i = 1$  imply that

$$\|(\mathbf{P} - \mathbf{Q})e_i\| \leq \max_{j=0, \dots, n} \|p_j - q_j\|. \quad (84)$$

and the explicit form of the L2-norm distance in Proposition 1, one can verify the optimality of the least squares reduction with respect to the L2-norm distance as follows:

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{Q}_m} d_{L2}(\mathbf{B}_{[p_0, \dots, p_n]}, \mathbf{B}_{[q_0, \dots, q_m]}\mathbf{E}(m, n))^2 \\ &= \frac{\partial}{\partial \mathbf{Q}_m} \text{tr}((\mathbf{P}_n - \mathbf{Q}_m\mathbf{E}(m, n))\mathbf{W}_n(\mathbf{P}_n - \mathbf{Q}_m\mathbf{E}(m, n))^T) \\ &= 2(\mathbf{Q}_m\mathbf{E}(m, n) - \mathbf{P}_n)\mathbf{W}_n \end{aligned}$$

which equals to zero for  $\mathbf{Q}_m = \mathbf{P}_n\mathbf{R}(n, m)$ . Thus, the global optimality follows from the convexity of the squared L2-norm distance.

Similarly, due to its strong relation with linear least squares, the Frobenius-norm distance of Bézier curves

$$d_F(\mathbf{B}_{[p_0, \dots, p_n]}, \mathbf{B}_{[q_0, \dots, q_m]}\mathbf{E}(m, n)) = \|\mathbf{P}_n - \mathbf{Q}_m\mathbf{E}(m, n)\|_F \quad (85)$$

is minimized via the pseudoinverse  $\mathbf{E}(m, n)^+$  of  $\mathbf{E}(m, n)$  at

$$\mathbf{Q}_m = \mathbf{P}_n\mathbf{E}(m, n)^+ = \mathbf{P}_n\mathbf{R}_{L2}(n, m).$$

■

### Q. Proof of Proposition 14

*Proof:* The result can be verified using (26) and (33) as

$$\begin{aligned} & \mathbf{E}(m, n)\mathbf{R}_{\tau, t_o}(n, m) \\ &= \mathbf{T}_m^b(m)\mathbf{I}_{(m+1) \times (n+1)}\mathbf{T}_b^m(n, t_o)\mathbf{I}_{(n+1) \times (m+1)}\mathbf{T}_b^\tau(m, t_o) \\ &= \mathbf{T}_m^b(m) \underbrace{\mathbf{I}_{m \times n}\mathbf{T}_\tau^m(n, t_o)\mathbf{I}_{n \times m}}_{=\mathbf{T}_\tau^m(m, t_o)} \mathbf{T}_b^\tau(m, t_o) \\ &= \mathbf{T}_m^b(m)\mathbf{T}_\tau^m(m, t_o)\mathbf{T}_b^\tau(m, t_o) \\ &= \mathbf{T}_m^b(m)\mathbf{T}_b^m(m) = \mathbf{I}_{(m+1) \times (m+1)}. \end{aligned}$$

■

### R. Proof of Proposition 15

*Proof:* Since  $\mathbf{b}(t_0, \dots, t_m)^{-1}\mathbf{b}_m(t_0, \dots, t_m) = \mathbf{I}_{(m+1) \times (m+1)}$ , we have for any  $t_i \in \{t_0, \dots, t_m\}$  that

$$\begin{aligned} \mathbf{B}_{[q_0, \dots, q_m]}(t_i) &= [q_0, \dots, q_m]\mathbf{b}_m(t_i) \\ &= [p_0, \dots, p_n]\mathbf{R}_{t_0, \dots, t_m}(n, m)\mathbf{b}_m(t_i) \\ &= [p_0, \dots, p_n]\mathbf{b}_n(t_0, \dots, t_m)\mathbf{b}_m(t_0, \dots, t_m)^{-1}\mathbf{b}_m(t_i) \\ &= [p_0, \dots, p_n]\mathbf{b}_n(t_i) = \mathbf{B}_{[p_0, \dots, p_n]}(t_i). \end{aligned}$$

Thus, the matching reduction preserves the curve at  $t_i$ .

### S. Proof of Proposition 16

*Proof:* Consider some additional distinct parameters  $t_{m+1}, \dots, t_n \in \mathbb{R}$  that are different from  $t_0, \dots, t_m$ . Then, the result can be verified using Proposition 5 as

$$\begin{aligned} & \mathbf{E}(m, n)\mathbf{R}_{t_0, \dots, t_m}(n, m) \\ &= \mathbf{E}(m, n)\mathbf{b}_n(t_0, \dots, t_m)\mathbf{b}_m(t_0, \dots, t_m)^{-1} \\ &= \mathbf{b}_m(t_0, \dots, t_m)\mathbf{b}_m(t_0, \dots, t_m)^{-1} \\ &= \mathbf{I}_{(m+1) \times (m+1)}. \end{aligned}$$

■

### P. Proof of Proposition 13

*Proof:* Using the following matrix identities [58]

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{XA}) = \mathbf{A}^T, \quad \text{and} \quad \frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{XAX}^T) = \mathbf{X}(\mathbf{A} + \mathbf{A}^T)$$

### T. Proof of Proposition 18

*Proof:* The matching reduction matrix  $\mathbf{R}_{t_0, \dots, t_n}(n+1, n)$  can be expressed in the monomial basis using the basis transformation between Bernstein and monomial bases as

$$\begin{aligned} \mathbf{R}_{t_0, \dots, t_n}(n+1, n) &= \mathbf{b}_{n+1}(t_0, \dots, t_n) \mathbf{b}_n(t_0, \dots, t_n)^{-1} \\ &= \mathbf{T}_m^b(n+1) \mathbf{m}_{n+1}(t_0, \dots, t_n) \mathbf{m}_n(t_0, \dots, t_n)^{-1} \mathbf{T}_b^m(n) \\ &= \mathbf{T}_m^b(n+1) \left[ \begin{array}{c} \mathbf{I}_{(n+1) \times (n+1)} \\ [t_0^{n+1}, \dots, t_n^{n+1}] \mathbf{m}_n(t_0, \dots, t_n)^{-1} \end{array} \right] \mathbf{T}_b^m(n). \end{aligned}$$

Hence, the degree-one matching reduction difference can be written for  $\mathbf{Q}_n = [q_0, \dots, q_n]$  and  $\mathbf{P}_{n+1} = [p_0, \dots, p_{n+1}]$  as

$$\begin{aligned} B_{p_0, \dots, p_{n+1}}(t) - B_{q_0, \dots, q_n}(t) &= \mathbf{P}_{n+1} \mathbf{b}_{n+1}(t) - \mathbf{Q}_n \mathbf{b}_n(t) \\ &= \mathbf{P}_{n+1} \mathbf{T}_m^b(n+1) \mathbf{m}_{n+1}(t) - \mathbf{P}_{n+1} \mathbf{T}_b^m(n+1) \\ &\quad \times \left[ \begin{array}{c} \mathbf{I}_{(n+1) \times (n+1)} \\ [t_0^{n+1}, \dots, t_n^{n+1}] \mathbf{m}_n(t_0, \dots, t_n)^{-1} \end{array} \right] \mathbf{T}_b^m(n) \mathbf{b}_n(t) \\ &= \mathbf{P}_{n+1} \mathbf{T}_m^b(n+1) \left( \mathbf{m}_{n+1}(t) \right. \\ &\quad \left. - \left[ \begin{array}{c} \mathbf{I}_{(n+1) \times (n+1)} \\ [t_0^{n+1}, \dots, t_n^{n+1}] \mathbf{m}_n(t_0, \dots, t_n)^{-1} \end{array} \right] \mathbf{m}_n(t) \right) \\ &= \mathbf{P}_{n+1} \mathbf{T}_m^b(n+1) \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ &\quad \times \left( t^{n+1} - [t_0^{n+1}, \dots, t_n^{n+1}] \mathbf{m}_n(t_0, \dots, t_n)^{-1} \mathbf{m}_n(t) \right). \end{aligned}$$

Now observe that for any distinct  $t_0, \dots, t_n \in \mathbb{R}$  one has

$$t^{n+1} - [t_0^{n+1}, \dots, t_n^{n+1}] \mathbf{m}_n(t_0, \dots, t_n)^{-1} \mathbf{m}_n(t) = \prod_{i=0}^n (t - t_i)$$

which is zero at  $t = t_0, \dots, t_n$ . We also have from Lemma 5

$$\begin{aligned} [\mathbf{T}_m^b(n+1)]_{i+1, n+2} &= (-1)^{n+1-i} \binom{n+1}{n+1} \binom{n+1}{i} \\ &= (-1)^{n+1-i} \binom{n+1}{i}. \end{aligned}$$

Hence, the matching reduction difference is given by

$$\begin{aligned} B_{p_0, \dots, p_{n+1}}(t) - B_{q_0, \dots, q_n}(t) &= \left( \sum_{i=0}^{n+1} (-1)^{n+1-i} \binom{n+1}{i} p_i \right) \prod_{i=0}^n (t - t_i). \end{aligned}$$

### REFERENCES

- [1] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2520–2525.
- [2] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Proc. Int. Symp. Robot. Res.*, pp. 649–666, 2016.
- [3] W. Ding, W. Gao, K. Wang, and S. Shen, "An efficient b-spline-based kinodynamic replanning framework for quadrotors," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1287–1306, Dec. 2019.
- [4] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 344–351.
- [5] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 922–938, Apr. 2022.
- [6] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [7] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2997–3004, Jun. 2019.
- [8] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using Bézier curves and MPC," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2016, pp. 826–833.
- [9] J. Pérez, J. Godoy, J. Villagrá, and E. Onieva, "Trajectory generator for autonomous vehicles in urban environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 409–414.
- [10] H. Ozaki and C.-J. Lin, "Optimal b-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1996, pp. 3592–3597.
- [11] K. Hauser and V. Ng-Thow-Hing, "Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 2493–2498.
- [12] C. Scheiderer, T. Thun, and T. Meisen, "Bézier curve based continuous and smooth motion planning for self-learning industrial robots," *Procedia Manuf.*, vol. 38, pp. 423–430, 2019.
- [13] X. Zhao, Z. Cao, W. Geng, Y. Yu, M. Tan, and X. Chen, "Path planning of manipulator based on RRT-connect and Bézier curve," in *Proc. IEEE Int. Conf. CYBER Technol. Automat., Control, Intell. Syst.*, 2019, pp. 649–653.
- [14] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, 2018, Art no. 3170.
- [15] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [16] M. J. Van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *Int. J. Robust Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, 1998.
- [17] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 969–1002, 2015.
- [18] S. Liu et al., "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017.
- [19] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 856–869, Aug. 2018.
- [20] J.-W. Choi, R. E. Curry, and G. H. Elkaim, "Continuous curvature path generation based on Bézier curves for autonomous vehicles," *IAENG Int. J. Appl. Math.*, vol. 40, no. 2, pp. 91–101, 2010.
- [21] D. González, J. Pérez, R. Lattarulo, V. Milanés, and F. Nashashibi, "Continuous curvature planning with obstacle avoidance capabilities in urban scenarios," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2014, pp. 1430–1435.
- [22] S. Tang and V. Kumar, "Safe and complete trajectory generation for robot teams with higher-order dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1894–1901.
- [23] J. A. Preiss, K. Hausman, G. S. Sukhatme, and S. Weiss, "Trajectory optimization for self-calibration and navigation," in *Proc. Robot., Sci. Syst.*, 2017.
- [24] N. Bucki and M. W. Mueller, "Rapid collision detection for multicopter trajectories," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7234–7239.
- [25] M. Tang, R. Tong, Z. Wang, and D. Manocha, "Fast and exact continuous collision detection with Bernstein sign classification," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 1–8, 2014.
- [26] Z. Wang, X. Zhou, C. Xu, J. Chu, and F. Gao, "Alternating minimization based trajectory generation for quadrotor aggressive flight," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4836–4843, Jun. 2020.
- [27] A. Artuñedo, J. Godoy, and J. Villagrá, "Smooth path planning for urban autonomous driving using openstreetmaps," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 837–842.

- [28] H. Li, Y. Luo, and J. Wu, "Collision-free path planning for intelligent vehicles based on Bézier curve," *IEEE Access*, vol. 7, pp. 123334–123340, 2019.
- [29] I. Bae, J. Moon, H. Park, J. H. Kim, and S. Kim, "Path generation and tracking based on a Bézier curve for a steering rate controller of autonomous vehicles," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2013, pp. 436–441.
- [30] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic Bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 6108–6113.
- [31] L. Han, H. Yashiro, H. Tehrani Nik Nejad, Q. H. Do, and S. Mita, "Bézier curve based path planning for autonomous vehicle in urban environment," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 1036–1042.
- [32] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intell. Transp. Syst.*, vol. 14, no. 13, pp. 1882–1891, 2020.
- [33] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 2427–2433.
- [34] B. Han and S. Liu, "RRT based obstacle avoidance path planning for 6-Dof manipulator," in *Proc. IEEE Data Driven Control Learn. Syst. Conf.*, 2020, pp. 822–827.
- [35] J. Pan, L. Zhang, and D. Manocha, "Collision-free and smooth trajectory computation in cluttered environments," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1155–1175, 2012.
- [36] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise Bézier curve for autonomous vehicle," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, 2013, pp. 17–22.
- [37] R. Cimurs, J. Hwang, and I. H. Suh, "Bézier curve-based smoothing for path planner with curvature constraint," in *Proc. IEEE Int. Conf. Robot. Comput.*, 2017, pp. 241–248.
- [38] X. Bu, H. Su, W. Zou, and P. Wang, "Curvature continuous path smoothing based on cubic Bézier curves for car-like vehicles," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2015, pp. 1453–1458.
- [39] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using b-spline curves," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 23–56, 2015.
- [40] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 561–568, Jun. 2010.
- [41] M. Elbanhawi, M. Simic, and R. Jazar, "Randomized bidirectional b-spline parameterization motion planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 406–419, Feb. 2016.
- [42] K. Yang et al., "Spline-based RRT path planner for non-holonomic robots," *J. Intel. Robot. Syst.*, vol. 73, no. 1, pp. 763–782, 2014.
- [43] R. T. Farouki, "The Bernstein polynomial basis: A centennial retrospective," *Comput. Aided Geom. Des.*, vol. 29, no. 6, pp. 379–419, 2012.
- [44] G. E. Farin, *Curves and Surfaces for CAGD: A Practical Guide*. Burlington, MA, USA: Morgan Kaufmann, 2002.
- [45] B.-G. Lee and Y. Park, "Distance for Bézier curves and degree reduction," *Bull. Aust. Math. Soc.*, vol. 56, no. 3, pp. 507–515, 1997.
- [46] M. Eck, "Least squares degree reduction of Bézier curves," *Comput.-Aided Des.*, vol. 27, no. 11, pp. 845–851, 1995.
- [47] B.-G. Lee, Y. Park, and J. Yoo, "Application of Legendre–Bernstein basis transformations to degree elevation and degree reduction," *Comput. Aided Geometric Des.*, vol. 19, no. 9, pp. 709–718, 2002.
- [48] H. Sunwoo and N. Lee, "A unified matrix representation for degree reduction of Bézier curves," *Comput. Aided Geometric Des.*, vol. 21, no. 2, pp. 151–164, 2004.
- [49] Y. Park, U. J. Choi, and H.-J. Kim, "Approximate conversion of Bézier curves," *Bull. Aust. Math. Soc.*, vol. 51, no. 1, pp. 153–162, 1995.
- [50] G.-D. Chen and G.-J. Wang, "Optimal multi-degree reduction of Bézier curves with constraints of endpoints continuity," *Comput. Aided Geometric Des.*, vol. 19, no. 6, pp. 365–377, 2002.
- [51] S. Zhou and X. Shen, "Spatially adaptive regression splines and accurate knot selection schemes," *J. Amer. Statist. Assoc.*, vol. 96, no. 453, pp. 247–259, 2001.
- [52] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal time allocation for quadrotor trajectory generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4715–4722.
- [53] D. Lutterkort, J. Peters, and U. Reif, "Polynomial degree reduction in the L<sub>2</sub>-norm equals best euclidean approximation of Bézier coefficients," *Comput. Aided Geometric Des.*, vol. 16, no. 7, pp. 607–612, 1999.
- [54] V.-E. Neagoe, "Inversion of the van der Monde matrix," *IEEE Signal Process. Lett.*, vol. 3, no. 4, pp. 119–120, Apr. 1996.
- [55] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds," *Int. J. Robot. Res.*, vol. 38, no. 2–3, pp. 196–223, 2019.
- [56] O. Arslan and A. Tiemessen, "Adaptive Bézier degree reduction and splitting for computationally efficient motion planning (Technical Report)," 2022, *arXiv:2201.07834*.
- [57] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [58] K. B. Petersen and M. S. Pedersen, *The matrix cookbook*. Technical Univ. Denmark, Nov. 2012, version 20121115.



**Ömür Arslan** (Member, IEEE) received the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2016 and the B.Sc. and M.Sc. degrees in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2007 and from Bilkent University, Ankara, in 2009, respectively.

He is an Assistant Professor with the Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands. His research interests include mobile robotics, motion planning, multirobot systems, mobile sensor networks, dynamical systems, machine learning, and machine perception.



**Aron Tiemessen** received the B.Sc. and M.Sc. degrees in mechanical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2019 and 2022, respectively.

His research interests include mobile robotics, motion planning, and robot perception.