Chapter 3
Piecewise Bezier Curves Path Planning with Continuous Curvature Constraint for Autonomous Driving

Ji-Wung Choi, Renwick Curry, and Gabriel Elkaim

Abstract We present two practical path planning algorithms based on Bezier curves for autonomous vehicles operating under waypoints and corridor constraints. Bezier curves have useful properties for the trajectory generation problem. This paper describes how the algorithms apply these properties to generate the reference trajectory for vehicles to satisfy the path constraints. Both algorithms generate the piecewise-Bezier-curves path such that the curves segments are joined smoothly with C^2 constraint which leads to continuous curvature along the path. The degree of the curves are minimized to prevent them from being numerically unstable. Additionally, we discuss the constrained optimization problem that optimizes the resulting path for a user-defined cost function.

1 Introduction

Bezier Curves were invented in 1962 by the French engineer Pierre Bezier for designing automobile bodies. Today Bezier Curves are widely used in computer graphics and animation. The Bezier curves have useful properties for the path generation problem as described in Section 2 of this paper. Hence many path planning techniques for autonomous vehicles have been discussed based on Bezier Curves in the literature. Cornell University Team for 2005 DARPA Grand Challenge used a path planner based on Bezier curves of degree 3 in a sensing/action feedback loop to generate smooth paths that are consistent with vehicle dynamics [5]. Skrjanc proposed a new cooperative collision avoidance method for multiple robots with constraints and known start and goal velocities based on Bezier curves of degree 4 [6]. In this method, four control points out of five are placed such that

J.-W. Choi (✉)
Autonomous Systems Lab, University of California, Santa Cruz, CA 95064, USA
e-mail: jwchoi@soe.ucsc.edu

S.-I. Ao et al. (eds.), *Machine Learning and Systems Engineering*,
Lecture Notes in Electrical Engineering 68,
DOI 10.1007/978-90-481-9419-3_3,

desired positions and velocities of the start and the goal point are satisfied. The fifth point is obtained by minimizing penalty functions. Jolly described Bezier curve based approach for the path planning of a mobile robot in a multi-agent robot soccer system [3]. The resulting path is planned such that the initial state of the robot and the ball, and an obstacle avoidance constraints are satisfied. The velocity of the robot along the path is varied continuously to its maximum allowable levels by keeping its acceleration within the safe limits. When the robot is approaching a moving obstacle, it is decelerated and deviated to another Bezier path leading to the estimated target position.

Our previous works introduced two path planning algorithms based on Bezier curves for autonomous vehicles with waypoints and corridor constraints [1, 2]. Both algorithms join cubic Bezier curve segments smoothly to generate the reference trajectory for vehicles to satisfy the path constraints. Also, both algorithms are constrained in that the path must cross over a bisector line of corner area such that the tangent at the crossing point is normal to the bisector. Additionally, the constrained optimization problem that optimizes the resulting path for user-defined cost function was discussed. Although the simulations provided in that paper showed the generation of smooth routes, discontinuities of the yaw angular rate appeared at junction nodes between curve segments. This is because the curve segments are constrained to connect each other by only C^1 continuity, so the curvature of the path is discontinuous at the nodes. (Section 2 describes this more detail.)

To resolve this problem, we propose new path planning algorithms. The algorithms impose constraints such that curve segments are C^2 continuous in order to have curvature continuous for every point on the path. In addition, they give the reference path more freedom by eliminating redundant constraints used in previous works, such as the tangent being normal to the bisector and symmetry of curve segments on corner area. The degree of each Bezier curve segments are determined by the minimum number of control points to satisfy imposed constraints while cubic Bezier curves are used for every segments in previous works. The optimized resulting path is obtained by computing the constrained optimization problem that controls the tradeoff between shortest and smoothest path generation. Furthermore, the proposed algorithms satisfy the initial and goal orientation as well as position constraint while previous works only made the position constraint satisfied. The numerical simulation results demonstrate the improvement of trajectory generation in terms of smoother steering control and smaller cross track error.

2 Bezier Curve

A Bezier Curve of degree n can be represented as

$$\mathbf{P}(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_i, \quad t \in [0, 1]$$

where $\mathbf{P}_i$ are control points and $B_i^n(t)$ is a Bernstein polynomial given by

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i, \quad i \in \{0, 1, \ldots, n\}$$

Bezier Curves have useful properties for path planning:

- They always start at $\mathbf{P}_0$ and stop at $\mathbf{P}_n$.
- They are always tangent to $\overline{\mathbf{P}_0\mathbf{P}_1}$ and $\overline{\mathbf{P}_n\mathbf{P}_{n-1}}$ at $\mathbf{P}_0$ and $\mathbf{P}_n$ respectively.
- They always lie within the convex hull consisting of their control points.

2.1 The de Casteljau Algorithm

The de Casteljau algorithm describes a recursive process to subdivide a Bezier curve $\mathbf{P}(t)$ into two segments. The subdivided segments are also Bezier curves. Let $\{\mathbf{P}_0^0, \mathbf{P}_1^0, \ldots, \mathbf{P}_n^0\}$ denote the control points of $\mathbf{P}(t)$. The control points of the segments can be computed by

$$\begin{aligned} \mathbf{P}_i^j &= (1-\tau)\mathbf{P}_i^{j-1} + \tau\,\mathbf{P}_{i+1}^{j-1}, \\ &\tau \in (0,1),\ j \in \{1, \ldots, n\},\ i \in \{0, \ldots, n-j\} \end{aligned} \tag{1}$$

Then, $\{\mathbf{P}_0^0, \mathbf{P}_0^1, \ldots, \mathbf{P}_0^n\}$ are the control points of one segment and $\{\mathbf{P}_0^n, \mathbf{P}_1^{n-1}, \ldots, \mathbf{P}_n{}^0\}$ are the another. (See an example in Fig. 1.) Note that, by applying the properties of Bezier Curves described in the previous subsection, both of the subdivided curves end at $\mathbf{P}_0^n$ and the one is tangent to $\overline{\mathbf{P}_0^{n-1}\mathbf{P}_0^n}$ and the other is to $\overline{\mathbf{P}_0^n\mathbf{P}_1^{n-1}}$ at the point. Since $\mathbf{P}_0^n$ is chosen on $\overline{\mathbf{P}_0^{n-1}\mathbf{P}_1^{n-1}}$ by using (1), the three points $\mathbf{P}_0^{n-1}$, $\mathbf{P}_0^n$, and $\mathbf{P}_1^{n-1}$ are collinear.

Remark 1. A Bezier curve $\mathbf{P}(t)$ constructed by control points $\{\mathbf{P}_0^0, \mathbf{P}_1^0, \ldots, \mathbf{P}_n^0\}$ always passes through the point $\mathbf{P}_0^n$ and is tangent to $\overline{\mathbf{P}_0^{n-1}\mathbf{P}_1^{n-1}}$ at $\mathbf{P}_0^n$.

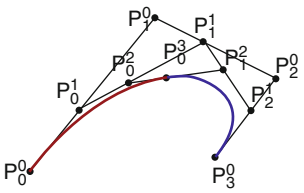

Fig. 1 Subdividing a cubic Bezier curve with $\tau = 0.4$ by the de Casteljau Algorithm

2.2 Derivatives, Continuity and Curvature

The derivatives of a Bezier curve can be determined by its control points. For a Bezier curve $\mathbf{P}(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_i$, the first derivative is given by

$$\dot{\mathbf{P}}(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t)\mathbf{D}_i \tag{2}$$

where $\mathbf{D}_i$, control points of $\dot{\mathbf{P}}(t)$ are

$$\mathbf{D}_i = n(\mathbf{P}_{i+1} - \mathbf{P}_i)$$

Geometrically, (2) provides us with a tangent vector. The higher order derivative can be obtained by using the relationship of (2), iteratively.

Two Bezier curves $\mathbf{P}(t)$ and $\mathbf{Q}(t)$ are said to be C^k at t_0 continuous if

$$\mathbf{P}(t_0) = \mathbf{Q}(t_0),\ \dot{\mathbf{P}}(t_0) = \dot{\mathbf{Q}}(t_0),\ \ldots,\ \mathbf{P}^{(k)}(t_0) = \mathbf{Q}^{(k)}(t_0) \tag{3}$$

The curvature of a Bezier curve $\mathbf{P}(t) = (x(t), y(t))$ at t is given by

$$\kappa(t) = \frac{|\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)|}{(\dot{x}^2(t) + \dot{y}^2(t))^{\frac{3}{2}}} \tag{4}$$

Lemma 1. *For the path constructed by two Bezier curve segments* $\mathbf{P}(t)|_{t\in[t_0,t_1]}$ *and* $\mathbf{Q}(t)|_{t\in[t_1,t_2]}$, *if* $\mathbf{P}(t)$ *and* $\mathbf{Q}(t)$ *are at least* C^2 *continuous at* t_1 *then the path has continuous curvature for every point on it.*

Proof. The curvature is expressed in terms of the first and the second derivative of a curve in (4). Since the Bezier curves are defined as polynomial functions of t, their k-th derivative for all $k = 1, 2, \ldots$ are continuous. Hence, they always have continuous curvature for all t. For two different Bezier curves $\mathbf{P}(t)$ and $\mathbf{Q}(t)$, it is sufficient that $\kappa(t_1)$, the curvature at the junction node is continuous if $\dot{\mathbf{P}}(t) = \dot{\mathbf{Q}}(t)$ and $\ddot{\mathbf{P}}(t) = \ddot{\mathbf{Q}}(t)$ are continuous at t_1.

3 Problem Statement

Consider the control problem of a ground vehicle with a mission defined by waypoints and corridor constraints in a two-dimensional free-space. Our goal is to develop and implement an algorithm for navigation that satisfies these constraints. That is divided into two parts: path planning and path following.

To describe each part, we need to introduce some notation. Each waypoint is denoted as $\mathbf{W}_i \in \mathbb{R}^2$ for $i \in \{1, 2, \ldots, N\}$, where $N \in \mathbb{R}$ is the total number of waypoints. Corridor width is denoted as w_j for $j \in \{1, \ldots, N-1\}$, j-th widths of each segment between two waypoints, $\mathbf{W}_j$ and $\mathbf{W}_{j+1}$. For the dynamics of the vehicle, the state and the control vector are denoted $\mathbf{q}(t) = (x_c(t), y_c(t), \psi(t))^T$ and $\mathbf{u}(t) = (v(t), \omega(t))^T$ respectively, where (x_c, y_c) represents the position of the center of gravity of the vehicle. The yaw angle ψ is defined to the angle from the X axis. v is the longitudinal velocity of the vehicle. $\omega = \dot{\psi}$ is the yaw angular rate. State space is denoted as $\mathcal{C} = \mathbb{R}^3$. We assume that the vehicle follows that

$$\dot{\mathbf{q}}(t) = \begin{pmatrix} \cos\psi(t) & 0 \\ \sin\psi(t) & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}(t)$$

With the notation defined above, the path planning problem is formulated as: Given an initial position and orientation and a goal position and orientation of the vehicle, generate a path λ specifying a continuous sequence of positions and orientations of the vehicle satisfying the path constraints [4]. In other words, we are to find a continuous map

$$\lambda : [0, 1] \rightarrow \mathcal{C}$$

with

$$\lambda(0) = \mathbf{q}_{init} \quad \text{and} \quad \lambda(1) = \mathbf{q}_{goal}$$

where $\mathbf{q}_{init} = (\mathbf{W}_1, \psi_0)$ and $\mathbf{q}_{goal} = (\mathbf{W}_N, \psi_f)$ are the initial and goal states of the path, respectively.

Given a planned path, we use the path following technique with feedback corrections as illustrated in Fig. 2. A position and orientation error is computed every 50 ms. A point $\mathbf{z}$ is computed with the current longitudinal velocity and

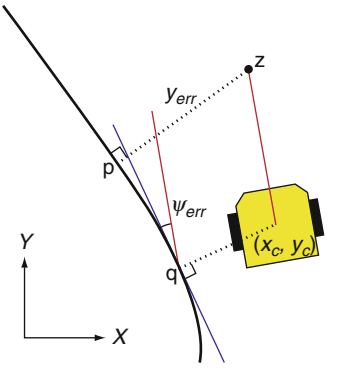

Fig. 2 The position error is measured from a point **z**, projected in front of the vehicle, and onto the desired curve to point **p**

heading of the vehicle from the current position. **z** is projected onto the reference trajectory at point **p** such that **zp** is normal to the tangent at **p**. The cross track error y_{err} is defined by the distance between **z** and **p**. The steering control ω uses a PID controller with respect to cross track error y_{err}.

$$\delta\omega = k_p y_{err} + k_d \frac{dy_{err}}{dt} + k_i \int y_{err} dt$$

4 Path Planning Algorithm

In this section, two path planning methods based on Bezier curves are proposed. To describe the algorithms, let us denote $\hat{\mathbf{b}}_j$ as the unit vector codirectional with the outward bisector of $\angle \mathbf{W}_{j-1}\mathbf{W}_j\mathbf{W}_{j+1}$ for $j \in \{2, \ldots, N-1\}$ as illustrated in Fig. 3. The planned path must cross over the bisectors under the waypoint and the corridor constraints. The location of the crossing point is represented as $\mathbf{W}_j + d_j \cdot \hat{\mathbf{b}}_j$, where $d_j \in \mathbb{R}$ is a scalar value. The course is divided into segments G_i by bisectors. G_i indicates the permitted area for vehicles under corridor constraint w_i, from $\mathbf{W}_i$ to $\mathbf{W}_{i+1}$.

Bezier curves constructed by large numbers of control points are numerically unstable. For this reason, it is desirable to join low-degree Bezier curves together in a smooth way for path planning [6]. Thus both of the algorithms use a set of low-degree Bezier curves such that the neighboring curves are C^2 continuous at their end nodes. This will lead to continuous curvature on the resulting path by Lemma 1.

The Bezier curves used for the path plannings are denoted as ${}^i\mathbf{P}(t) = \sum_{k=0}^{n_i} B_k^{n_i}(t) \cdot {}^i\mathbf{P}_k$ for $i \in \{1, \ldots, M\}$, $t \in [0, 1]$, where M is the total number of the Bezier curves and n_i is the degree of ${}^iP$. The planned path λ is a concatenation of all ${}^iP$ such that

$$\lambda((i-1+t)/M) = {}^i P(t), \quad i \in \{1, \ldots, M\},\ t \in [0, 1]$$

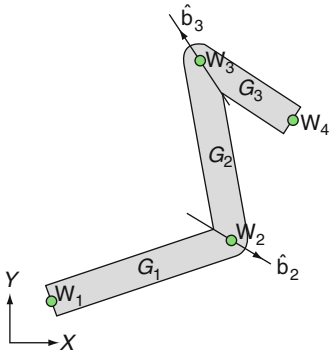

Fig. 3 An example of the course with four waypoints. *Gray area* is the permitted area for vehicles under a corridor constraint

4.1 Path Planning Placing Bezier Curves within Segments (BS)

In this path planning method (BS), the Bezier curve ${}^{i}\mathbf{P}$ for $i \in \{1, \ldots, N-1\}$ are used within each segment G_i. The adjacent curves, ${}^{j-1}\mathbf{P}$ and ${}^{j}\mathbf{P}$ are C^2 continuous at the crossing point, $\mathbf{W}_j + d_j \cdot \hat{\mathbf{b}}_j$. The control points of ${}^{i}\mathbf{P}$, ${}^{i}\mathbf{P}_k$ for $k = \{0, \ldots, n_i\}$ are determined to maintain these conditions.

- The beginning and the end point are $\mathbf{W}_1$ and $\mathbf{W}_N$.

$$ {}^{1}\mathbf{P}_0 = \mathbf{W}_1, \quad {}^{N-1}\mathbf{P}_{n_{N-1}} = \mathbf{W}_N \tag{5} $$

- The beginning and end orientation of λ are ψ_0 and ψ_f.

$$ {}^{1}\mathbf{P}_1 = \mathbf{W}_1 + l_0(\cos\psi_0, \sin\psi_0), \quad l_0 \in \mathbb{R}^+ \tag{6a} $$

$$ {}^{N-1}\mathbf{P}_{n_{N-1}-1} = \mathbf{W}_N - l_f(\cos\psi_f, \sin\psi_f), \quad l_f \in \mathbb{R}^+ \tag{6b} $$

- ${}^{j-1}\mathbf{P}$ and ${}^{j}\mathbf{P}$, $\forall j \in \{2, \ldots, N-1\}$ are C^2 continuous at the crossing point.

$$ {}^{j-1}\mathbf{P}_{n_{j-1}} = {}^{j}\mathbf{P}_0 = \mathbf{W}_j + d_j \cdot \hat{\mathbf{b}}_j \tag{7a} $$

$$ n_{j-1}({}^{j-1}\mathbf{P}_{n_{j-1}} - {}^{j-1}\mathbf{P}_{n_{j-1}-1}) = n_j({}^{j}\mathbf{P}_1 - {}^{j}\mathbf{P}_0) \tag{7b} $$

$$ \begin{aligned} &n_{j-1}(n_{j-1}-1)({}^{j-1}\mathbf{P}_{n_{j-1}} - 2\cdot{}^{j-1}\mathbf{P}_{n_{j-1}-1} + {}^{j-1}\mathbf{P}_{n_{j-1}-2}) \\ &\quad = n_j(n_j-1)({}^{j}\mathbf{P}_2 - 2\,{}^{j}\mathbf{P}_1 + {}^{j}\mathbf{P}_0) \end{aligned} \tag{7c} $$

- The crossing points are bounded within the corridor.

$$ |d_j| < \forall \frac{1}{2}\min(w_{j-1}, w_j) \tag{8} $$

- ${}^{i}\mathbf{P}_k$, $k = \{1, \ldots, n_i - 1\}$ always lie within the area of G_i.

$$ {}^{i}\mathbf{P}_1 \in G_i, \ \ldots, \ {}^{i}\mathbf{P}_{n_i-1} \in G_i \tag{9} $$

Equations (6a, b) are derived by using the tangent property of Bezier curves at their end points. Equations (7a–c) are obtained by applying (2) and (3). Equation (9) makes the resulting Bezier curve satisfy the corridor constraint by the convex hull property. At each crossing point, three control points of each adjacent Bezier curve are dedicated to the C^2 continuity constraint by (2), (4), and Lemma 1. So the minimum number of control points to satisfy the constraints independent of the others are six for ${}^{i}\mathbf{P}$, $i \in \{2, \ldots, N-2\}$. On other hand, ${}^{1}\mathbf{P}$ needs five: three to

connect ${}^{2}\mathbf{P}$ plus two to connect $\mathbf{W}_1$ with slope of ψ_0. Likewise, ${}^{2N-3}\mathbf{P}$ needs five. Thus, n_i is determined as

$$n_i = \begin{cases} 4, & i \in \{1, N-1\} \\ 5, & i \in \{2, \ldots, N-2\} \end{cases}$$

Note that ${}^{1}\mathbf{P}_0$ and ${}^{N-1}\mathbf{P}_{n_{N-1}}$ are fixed in (5). ${}^{1}\mathbf{P}_1$ and ${}^{N-1}\mathbf{P}_{n_{N-1}-1}$ are fixed in (6a, b). ${}^{j-1}\mathbf{P}_{n_{j-1}}$ and ${}^{j}\mathbf{P}_0$ rely on d_j in (7a–c). ${}^{j-1}\mathbf{P}_{n_{j-1}-1}$ and ${}^{j-1}\mathbf{P}_{n_{j-1}-2}$ rely on ${}^{j}\mathbf{P}_1$ and ${}^{j-1}\mathbf{P}_2$. So the free variables are, $\forall j \in \{2, \ldots, N-1\}$, $\mathbf{P}_1 = \{{}^{j}\mathbf{P}_1\}$, $\mathbf{P}_2 = \{{}^{j}\mathbf{P}_2\}$, $\mathbf{d} = \{d_j\}$, and $\mathbf{L} = \{l_0, l_f\}$. The number of the variables or the degrees of freedom is $5N - 8$. The variables are computed by minimizing the constrained optimization problem:

$$\min_{\mathbf{P}_1, \mathbf{P}_2, \mathbf{d}, \mathbf{L}} J = \sum_{i=1}^{N-1} J_i \tag{10}$$

subject to (8) and (9), where J_i is the cost function of ${}^{i}\mathbf{P}(t)$, given by

$$J_i = \int_0^1 [(a_i(\dot{x}^2(t) + \dot{y}^2(t)) + b_i |{}^{i}\kappa(t)|^2] dt \tag{11}$$

where $a_i \in \mathbb{R}$ and $b_i \in \mathbb{R}$ are constants that control the tradeoff between arc lengths versus curvatures of the resulting path.

4.2 Path Planning Placing Bezier Curves on Corners (BC)

Another path planning method (BC) adds quadratic Bezier curves on the corner area around $\mathbf{W}_j$, $j \in \{2, \ldots, N-1\}$. The quadratic Bezier curves are denoted as ${}^{j}\mathbf{Q}(t) = \sum_{k=0}^{2} B_k^2(t) \cdot {}^{j}\mathbf{Q}_k^0$ intersects the j-th bisector. The first and the last control point, ${}^{j}\mathbf{Q}_0^0$ and ${}^{j}\mathbf{Q}_2^0$ are constrained to lie within G_{j-1} and G_j, respectively. Within each segment G_i, another Bezier curve is used to connect the end points of ${}^{j}\mathbf{Q}$ with C^2 continuity and/or $\mathbf{W}_1$ with slope of ψ_0 and/or $\mathbf{W}_N$ with ψ_f. Hence, ${}^{j}\mathbf{Q}$ is the curve segments of even number index:

$${}^{2(j-1)}\mathbf{P}(t) = {}^{j}\mathbf{Q}(t), \quad j \in \{2, \ldots, N-1\}$$

The degree of ${}^{i}\mathbf{P}$ is determined by the minimum number of control points to satisfy C^2 continuity constraint, independently:

$$n_i = \begin{cases} 4, & i \in \{1, 2N-3\} \\ 2, & i \in \{2, 4, \ldots, 2N-4\} \\ 5, & i \in \{3, 5, \ldots, 2N-5\} \end{cases}$$

Let t_c denote the Bezier curve parameter corresponding to the crossing point of ${}^j\mathbf{Q}(t)$ on the bisector, such that

$$ {}^j\mathbf{Q}(t_c) = \mathbf{W}_j + d_j \cdot \hat{\mathbf{b}}_j. \tag{12} $$

Let θ_j denote the angle of the tangent vector at the crossing point from X-axis:

$$ {}^j\dot{\mathbf{Q}}(t_c) = (|{}^j\dot{\mathbf{Q}}(t_c)| \cos\theta_j,\ |{}^j\dot{\mathbf{Q}}(t_c)| \sin\theta_j). \tag{13} $$

The notations are illustrated in Fig. 4. Due to the constraint of ${}^j\mathbf{Q}_0^0$ and ${}^j\mathbf{Q}_2^0$ within G_{j-1} and G_j, the feasible scope of θ_j is limited to the same direction as $\mathbf{W}_{j+1}$ is with respect to $\hat{\mathbf{b}}_j$. In other words, if $\mathbf{W}_{j+1}$ is to the right of $\hat{\mathbf{b}}_j$, then θ_j must point to the right of $\hat{\mathbf{b}}_j$, and vice versa.

Given ${}^j\mathbf{Q}_0^0$, ${}^j\mathbf{Q}_2^0$, d_j, and θ_j, the other control point ${}^j\mathbf{Q}_1^0$ is computed such that the crossing point is located at $\mathbf{W}_j + d_j \cdot \hat{\mathbf{b}}_j$ and the angle of the tangent vector at the crossing point is θ_j. Since each control point is two-dimensional, the degrees of freedom of ${}^j\mathbf{Q}(t)$ are six. Since d_j and θ_j are scaler, representing ${}^j\mathbf{Q}(t)$ in terms of ${}^j\mathbf{Q}_0^0$, ${}^j\mathbf{Q}_2^0$, d_j, and θ_j does not affect the degrees of freedom. However, it comes up with an advantage for corridor constraint. If we compute ${}^j\mathbf{Q}_1^0$ such as above, the points computed by applying the de Casteljau algorithm such that two subdivided curves are separated by the j-th bisector are represented as ${}^j\mathbf{Q}_0^0$ and ${}^j\mathbf{Q}_2^0$ as described in the following. The two curves are constructed by $\{{}^j\mathbf{Q}_0^0, {}^j\mathbf{Q}_0^1, {}^j\mathbf{Q}_0^2\}$ and $\{{}^j\mathbf{Q}_0^2, {}^j\mathbf{Q}_1^1, {}^j\mathbf{Q}_2^0\}$. We can test if the convex hull of $\{{}^j\mathbf{Q}_0^0, {}^j\mathbf{Q}_0^1, {}^j\mathbf{Q}_0^2\}$ lies within G_{j-1} and if that of $\{{}^j\mathbf{Q}_0^2, {}^j\mathbf{Q}_1^1, {}^j\mathbf{Q}_2^0\}$ lies within G_j in (26), instead of testing that of $\{{}^j\mathbf{Q}_0^0, {}^j\mathbf{Q}_1^0, {}^j\mathbf{Q}_2^0\}$. (Note that ${}^j\mathbf{Q}_1^0$ is not constrained to lie within corridor as shown in Fig. 4.) So, the convex hull property is tested for tighter conditions against the corridor constraint without increasing the degrees of freedom.

In order to compute ${}^j\mathbf{Q}_1^0$, the world coordinate frame T is transformed and rotated into the local frame ${}^jT$ where the origin is at the crossing point, ${}^j\mathbf{Q}(t_c)$ and X axis is codirectional with the tangent vector of the curve at the crossing point, ${}^j\dot{\mathbf{Q}}(t_c)$.

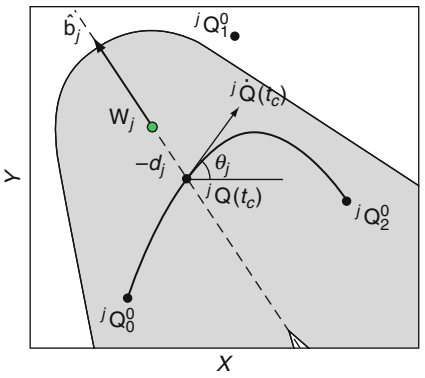

Fig. 4 The geometry of the ${}^j\mathbf{Q}(t)$ in corner area around $\mathbf{W}_j$

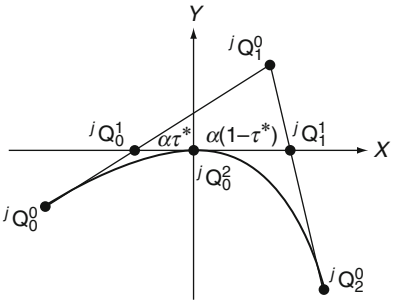

Fig. 5 The geometry of the control points of ${}^j\mathbf{Q}(t)$ with respect to ${}^jT$

Let us consider the subdivision ratio, $\tau^* \in (0, 1)$ such that the location of ${}^j\mathbf{Q}_0^2$ computed by applying the de Casteljau algorithm with it is the crossing point. In other words, τ^* has ${}^j\mathbf{Q}_0^2$ be at the origin with respect to ${}^jT$ frame. Fig. 5 illustrates the control points of ${}^j\mathbf{Q}(t)$ with respect to ${}^jT$ frame. Note that ${}^j\mathbf{Q}_0^2$ is at the origin by the definition of ${}^jT$ and τ^*. ${}^j\mathbf{Q}_0^1$ and ${}^j\mathbf{Q}_1^1$ are on the X axis by the definition of ${}^jT$ and Remark 1. Let the coordinates of the control points be denoted as $\mathbf{Q}_i^0 = (x_i, y_i)$, $i \in \{0, 1, 2\}$, where all coordinates are with respect to ${}^jT$.

Lemma 2. *Given d_j and θ_j, for ${}^j\mathbf{Q}(t)$ to intersect j-th bisector with the crossing point determined by the d_j and (12), and the tangent vector at the point determined by the θ_j and (13), it is necessary that $y_0 y_2 \geq 0$.*

Proof. Let $(x(t), y(t))$ denote the coordinate of ${}^j\mathbf{Q}(t)$ with respect to ${}^jT$. By the definition of ${}^jT$ and Remark 1, $\mathbf{Q}(t)$ passes through the origin with tangent slope of zero with respect to ${}^jT$. That is, $x(t_c) = 0$, $y(t_c) = 0$ and $\dot{y}(t_c) = 0$. Suppose that $y_0 = y(0) < 0$. Since $y(t)$ is a quadratic polynomial, $\dot{y}(t) > 0$ and $\ddot{y}(t) < 0$ for $t \in [0, t_c)$. Subsequently, $\dot{y}(t) < 0$ and $\ddot{y}(t) < 0$ for $t \in (t_c, 1]$. Thus, $y_2 = y(1) < 0$ and $y_0 y_2 > 0$. Similarly, if $y_0 > 0$ then $y_1 > 0$. If $y_0 = 0$ then $\dot{y}(t) = 0$ for $t \in [0, 1]$ and $y_2 = 0$. Thus, $y_0 y_2 = 0$. □

We are to calculate ${}^j\mathbf{Q}_1^0$ depending on whether $y_0 y_2$ is nonzero. For simplicity, superscript j is dropped from now on. Without loss of generality, suppose that $y_0 < 0$ and $y_2 < 0$. $\mathbf{Q}_0^2$ is represented as

$$\mathbf{Q}_0^2 = (1 - \tau^*)\mathbf{Q}_0^1 + \tau^*\mathbf{Q}_1^1$$

by applying (1). So the coordinates of $\mathbf{Q}_0^1$ and $\mathbf{Q}_1^1$ can be represented as

$$\mathbf{Q}_0^1 = (-\alpha\tau^*, 0),\ \mathbf{Q}_1^1 = (\alpha(1 - \tau^*), 0), \quad \alpha > 0 \tag{14}$$

where $\alpha > 0$ is some constant. Applying (1) with $i = 0$ and $j = 1$ and arranging the result with respect to $\mathbf{Q}_1^0$ by using (14) gives

$$\mathbf{Q}_1^0 = \left(-\alpha - \frac{1 - \tau^*}{\tau^*} x_0, -\frac{1 - \tau^*}{\tau^*} y_0 \right) \tag{15}$$

Similarly, applying (1) with $i = 1$ and $j = 1$ yields

$$\mathbf{Q}_1^0 = \left(\alpha - \frac{\tau^*}{1-\tau^*}x_2, -\frac{\tau^*}{1-\tau^*}y_2\right) \tag{16}$$

where α and τ^* are obtained by equations (15) and (16):

$$\tau^* = \frac{1}{1+\sqrt{y_2/y_0}}, \quad \alpha = \frac{x_0 y_2 - y_0 x_2}{2y_0\sqrt{y_2/y_0}} \tag{17}$$

τ^* and α have the square root of y_2/y_0. So, if $y_0 y_2 < 0$ then τ^* and α are not determined, hence, $\mathbf{Q}(t)$ is infeasible. That is, (17) ends up with Lemma 2.

If $y_0 = y_2 = 0$ then all control points of ${}^j\mathbf{Q}$ are on X axis (see proof of Lemma 2). In the geometric relation of control points and the points computed by applying the de Casteljau algorithm as shown in Fig. 6, we obtain

$$\begin{aligned} x_0 &= -(\alpha+\beta)\tau \\ x_2 &= (\alpha+\gamma)(1-\tau) \\ \alpha &= \beta(1-\tau)+\gamma\tau \end{aligned} \tag{18}$$

where $\alpha > 0$, $\beta > 0$, $\gamma > 0$ are some constants. Using (18), $\mathbf{Q}_1^0 = (x_1, 0)$ is represented in terms of arbitrary $\tau \in (0, 1)$:

$$x_1 = -\frac{1}{2}\left(\frac{1-\tau}{\tau}x_0 + \frac{\tau}{1-\tau}x_2\right) \tag{19}$$

The constraints imposed on the planned path are formulated as follows:

- The beginning and end position of λ are $\mathbf{W}_1$ and $\mathbf{W}_N$.

$${}^1\mathbf{P}_0 = \mathbf{W}_1, \quad {}^{2N-3}\mathbf{P}_4 = \mathbf{W}_N \tag{20}$$

- The beginning and end orientation of λ are ψ_0 and ψ_f.

$${}^1\mathbf{P}_1 = \mathbf{W}_1 + l_0(\cos\psi_0, \sin\psi_0), \quad l_0 \in \mathbb{R}^+ \tag{21a}$$

$${}^{2N-3}\mathbf{P}_3 = \mathbf{W}_N - l_f(\cos\psi_f, \sin\psi_f), \quad l_f \in \mathbb{R}^+ \tag{21b}$$

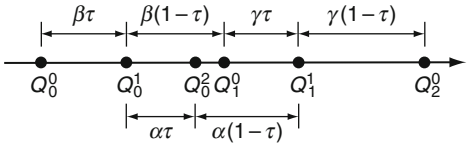

Fig. 6 The geometry of ${}^j\mathbf{Q}(t)$ with respect to ${}^jT$ when $y_0 = y_2 = 0$

- ${}^{i-1}\mathbf{P}$ and ${}^{i}\mathbf{P}$, $\forall i \in \{2, \ldots, 2N-3\}$ are C^2 continuous at the junctions.

$$ {}^{i-1}\mathbf{P}_{n_{i-1}} = {}^{i}\mathbf{P}_0 \tag{22a} $$

$$ n_{i-1}({}^{i-1}\mathbf{P}_{n_{i-1}} - {}^{i-1}\mathbf{P}_{n_{i-1}-1}) = n_i({}^{i}\mathbf{P}_1 - {}^{i}\mathbf{P}_0) \tag{22b} $$

$$ \begin{aligned} &n_{i-1}(n_{i-1}-1)({}^{i-1}\mathbf{P}_{n_{i-1}} - 2\cdot {}^{i-1}\mathbf{P}_{n_{i-1}-1} + {}^{i-1}\mathbf{P}_{n_{i-1}-2}) \\ &\quad = n_i(n_i-1)({}^{i}\mathbf{P}_2 - 2\cdot {}^{i}\mathbf{P}_1 + {}^{i}\mathbf{P}_0) \end{aligned} \tag{22c} $$

- The crossing points are bounded within the corridor.

$$ |d_j| < \frac{1}{2}\min(w_{j-1}, w_j), \quad \forall j \in \{2, \ldots, N-1\} \tag{23} $$

- θ_j has the same direction as $\mathbf{W}_{j+1}$ is with respect to $\hat{\mathbf{b}}_j$.

$$ \begin{aligned} &\text{mod}\ (\angle(\mathbf{W}_{j+1} - \mathbf{W}_j) - \angle\hat{\mathbf{b}}_j, 2\pi) > \pi \\ &\Rightarrow \text{mod}\ (\theta_j - \angle\hat{\mathbf{b}}_j, 2\pi) > \pi, \end{aligned} \tag{24a} $$

$$ \begin{aligned} &\text{mod}\ (\angle(\mathbf{W}_{j+1} - \mathbf{W}_j) - \angle\hat{\mathbf{b}}_j, 2\pi) < \pi \\ &\Rightarrow \text{mod}\ (\theta_j - \angle\hat{\mathbf{b}}_j, 2\pi) < \pi \end{aligned} \tag{24b} $$

- ${}^{j}\mathbf{Q}_0^0$ and ${}^{j}\mathbf{Q}_2^0$ with respect to ${}^{j}T$ satisfies Lemma 2.

$$ y_0 y_2 \geq 0 \tag{25} $$

where y_0 and y_2 are with respect to ${}^{j}T$.
- ${}^{j}\mathbf{Q}_0^0$ and ${}^{j}\mathbf{Q}_0^1$ lie within G_{j-1}. ${}^{j}\mathbf{Q}_2^0$ and ${}^{j}\mathbf{Q}_1^1$ lie within G_j.

$$ {}^{j}\mathbf{Q}_0^0 \in G_{j-1}, {}^{j}\mathbf{Q}_0^1 \in G_{j-1}, {}^{j}\mathbf{Q}_2^0 \in G_j, {}^{j}\mathbf{Q}_1^1 \in G_j \tag{26} $$

- $\{{}^{i}\mathbf{P}_1, \ldots, {}^{i}\mathbf{P}_{ni-1}\}$ always lie within the area of G_i.

$$ {}^{i}\mathbf{P}_1 \in G_i, \ \ldots, {}^{i}\mathbf{P}_{n_i-1} \in G_i, \quad i \in \{1, 3, \ldots, 2N-3\} \tag{27} $$

The free variables are, for $j \in \{2, \ldots, N-1\}$, $\mathbf{Q} = \{{}^{j}\mathbf{Q}_0, {}^{j}\mathbf{Q}_2\}$, $\mathbf{d} = \{d_j\}$, $\boldsymbol{\theta} = \{\theta_j\}$, and $\mathbf{L} = \{l_0, l_f\}$. The degrees of freedom is $6N-10$. The variables are computed by minimizing the constrained optimization problem:

$$ \min_{\mathbf{Q},\mathbf{d},\theta,\mathbf{L}} J = \sum_{i=1}^{N-1} J_i \tag{28} $$

subject to (23), (24,b), (25), (26), and (27), where J_i is the cost function of ${}^{i}\mathbf{P}(t)$, defined in (11). Notice that the convex hull property is tested for ${}^{j}\mathbf{Q}_0^1$ and ${}^{j}\mathbf{Q}_1^1$ of the divided curves instead of ${}^{j}\mathbf{Q}_1^0$ in (26). Thus, it comes up with tighter conditions for curves against the corridor constraint.

5 Simulation Results

Simulations were run for the course with four waypoints and identical corridor width of 8 as illustrated in Fig. 7. The initial and goal orientation are given by $\psi_0 = \psi_f = 0$. For path following, the constant longitudinal velocity $v(t) = 10\ m/s$ is used. The magnitude of ω is bounded within $|\omega|_{max} = 25$ rpm. The PID gains are given by: $k_p = 2$, $k_d = 1$, and $k_i = 0.1$.

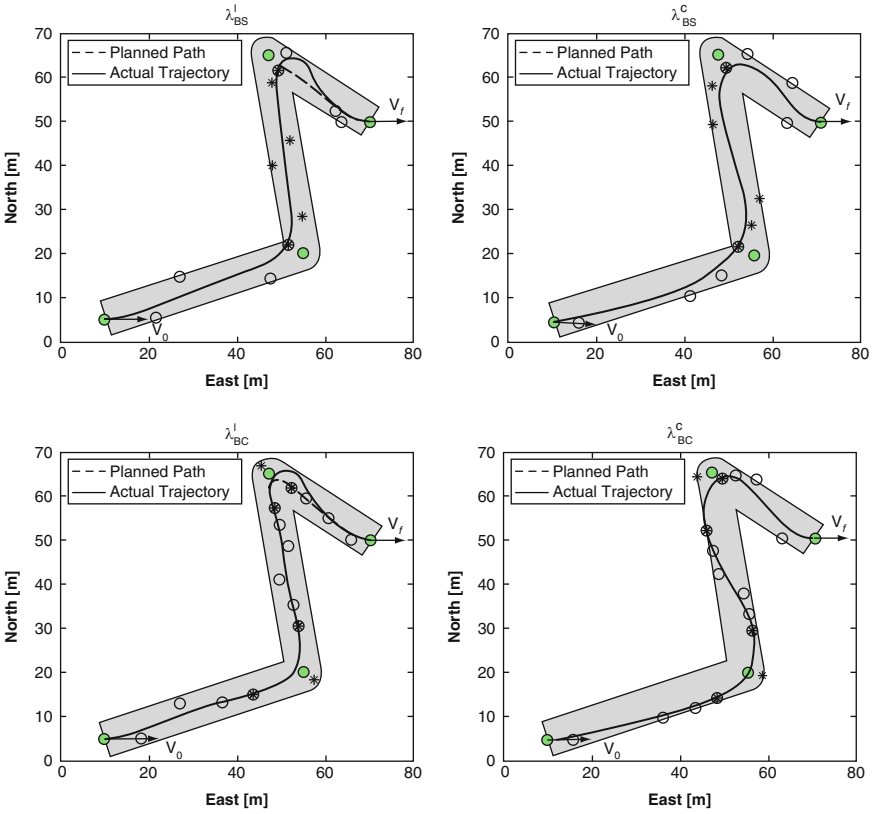

Fig. 7 The planned paths by BS method (*top*) and by BC method (*bottom*). Each pair of results are obtained by minimizing arc lengths (*left*) and by minimizing curvatures (*right*)

In Fig. 7, the reference paths (dashed curves) planned by applying the BS and BC method are placed in top and bottom row, respectively. The actual trajectories (solid curves) are generated by using the proposed tracking method. In the figures, stars indicate the location of control points of Bezier curve segments at an even number index, and empty circles indicate those of the others. The paths in the left column are planned by minimizing the summation of (11) with $a_i = 1$ and $b_i = 0$. Since only arc length is penalized, the cost function leads to paths with minimum arc length, which we denote λ^l_{BS} and λ^l_{BC} for the BS and BC method, respectively. On other hand, the paths in the right column are planned by minimizing the cost function with $a_i = 0$ and $b_i = 1$ so that resulting paths with larger radii of curvature are provided. We denote them λ^c_{BS} and λ^c_{BC} for the BS and BC method, respectively. λ^c_{BS} and λ^c_{BC} generate longer but smoother trajectory guidance than λ^l_{BS} and λ^l_{BC}. Taking a look at the tracking results on λ^l_{BS} and λ^l_{BC}, the vehicle overshoots in sharp turns around $\mathbf{W}_3$ resulting in a large position error (see Fig. 8) due to the limit on steering angle rate. The commanded steering angle rate on λ^l_{BC}, marked by '*' in Fig. 9 undergoes rapid changes and is constrained by the rate limit. However, on λ^c_{BS} and λ^c_{BC}, the vehicle tracks the part of the planned paths accurately thanks to larger radii of curvature. We can verify that more tangibly in cross track errors plot provided in Fig. 8. Also, the steering control signal on λ^c_{BC}, marked by 'o' in Fig. 9 is smoother than that on λ^l_{BC}.

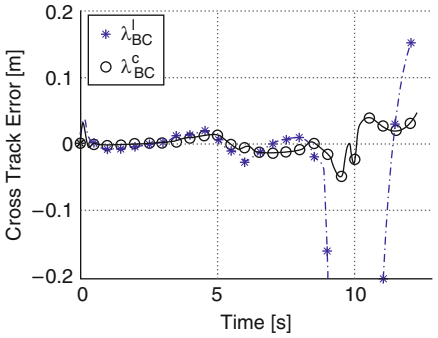

Fig. 8 The cross track errors by BC

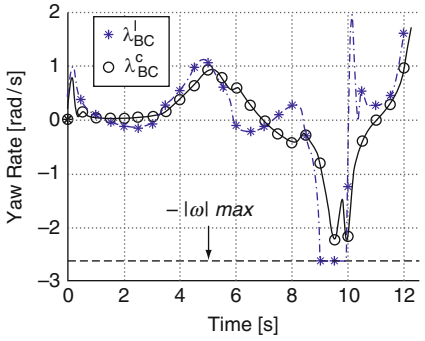

Fig. 9 The steering controls by BC

The main difference of the proposed algorithm from the previous ones of [1] is the degree of continuity at the junctions: C^2. Assuming that the vehicle tracks a reference path perfectly and v is continuous, if κ is continuous then ω is continuous because $\omega = \kappa v$. When v is constant as this simulation, since ω is proportional to κ, the continuity characteristic of ω tends to follow that of κ. Since the previous algorithms imposed only C^1 continuity on junction nodes of the path, κ is discontinuous at the nodes. Hence the path underwent the discontinuity of angular rate, that is, large angular acceleration, which leads to large torque on the vehicle. On other hand, the proposed algorithm has κ continuous along the resulting paths. If the path has curvature small enough so that the vehicle is able to track it accurately given a maximum steering angle rate, then the steering control signal will be smooth, as that of λ_{BC}^{c} in Fig. 9.

6 Conclusions

This paper presents two path planning algorithms based on Bezier curves for autonomous vehicles with waypoints and corridor constraints. Bezier curves provide an efficient way to generate the optimized path and satisfy the constraints at the same time. The simulation results also show that the trajectory of the vehicle follows the planned path within the constraints.

References

1. J. Choi, R.E. Curry, G.H. Elkaim, in *Path Planning Based on Bezier Curve for Autonomous Ground Vehicles* (Chapter 19, IEEE Computer Society, 2009), pp. 158–166
2. J. Choi, G.H. Elkaim, Bezier curves for trajectory guidance. *Proceedings of the World Congress on Engineering and Computer Science, WCECS 2008*, San Francisco, CA (2008)
3. K.G. Jolly, K.R. Sreerama, R. Vijayakumar, A bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. Robot. Autonom. Syst. **57**, 23–33 (2009)
4. J.C. Latombe, in *Robot Motion Planning* (Kluwer, Norwell, MA, 1991)
5. I. Miller, S. Lupashin, N. Zych, P. Moran, B. Schimpf, A. Nathan, E. Garcia, in *Cornell University's 2005 DARPA Grand Challenge Entry*, vol. 36, chapter 12 (Springer, Heidelberg, 2007), pp. 363–405
6. I. Skrjanc, G. Klancar, Cooperative collision avoidance between multiple robots based on bezier curves. *Proceedings of the Information Technology Interfaces, 2007 (ITI 2007)*, pp. 451–456