

Chapter 9

Solutions for Path Planning Using Spline Parameterization

M. Elbanhawi, M. Simic, and Reza N. Jazar

9.1 Introduction

Human drivers display remarkable abilities when controlling vehicles in a highly reactive manner and with impeccable precision. Even more impressive is the implicit consideration of the vehicle and road parameters. Researchers have shown that humans use specific visual cues to identify road curvature and endeavor to match it. Analysis that drew inspiration from the steering commands, used by operators of varying driving experience, is attempted to generate likewise natural paths. This approach can be combined with a path planning algorithm to generate paths with natural smooth trajectories for autonomous vehicles. This will circumvent the need to rely on computationally intensive planning algorithms that are based on forward model integration. Humans have been controlling the steering wheel for the past century. Currently self-driving is emerging as technology that promises to improve our lives greatly. Cars are underactuated systems designed to facilitate their control for operators. This simplification of actuation has adverse effects when attempting to automate such systems, leading to the appearance of nonholonomic constraints (Jazar 2008). Researchers have shown that drivers rely on certain visual landmarks to assess the path curvature prior to attempting to steer toward it (Land and Lee 1994; Land and Horwood 1995). The majority of studies, conducted on human steering, focus on modeling and predicating it, using control system theories (Donges 1978; MacAdam 1981; Prokop 2001). The aim of this chapter is to employ an efficient spline parameterization method to synthesize paths

M. Elbanhawi • M. Simic
School of Engineering, RMIT University, Melbourne, Australia

R.N. Jazar (✉)
School of Engineering, RMIT University, Melbourne, Australia

Xiamen University of Technology, Xiamen, China
e-mail: reza.jazar@rmit.edu.au

that mimic human steering controls, in order to generate motions that feel natural and familiar to human passengers. It is not uncommon for robotics researchers to draw inspiration by observing natural behavior (Donghyun et al. 2014; Lentink 2014). We hope that the results, presented in this chapter, can be employed in robotics to efficiently plan spline-based paths, similar to Yang et al. (2014), in addition to mimicking human steering, improving passenger comfort, and explicitly considering the limitations of the car.

Advances in sensing technology, computer vision, communications, and computational power have contributed toward the development of autonomous agents in a wide range of fields. Self-driving ground vehicles are used in military, urban transportation, and industrial and agricultural applications. Unmanned aerial vehicles (UAVs) and micro aerial vehicles (MAVs) are considered as a cost-effective, safe, and efficient choice for several military and civil applications. Robotic platforms are currently equipped with multiple sensors, which enable them to sense their surroundings and localize themselves in reference to their environment, goals, and obstacles. Path planning is a widely studied, fundamental task for mobile robots. Robot navigation mandates a strategy that steers it from its current location, through the environment while avoiding obstacles toward its goal.

Classical planning algorithms, such as A* algorithm (Hart et al. 1968), Voronoi diagrams (Canny 1985), visibility graphs (Asano et al. 1985), and cell decompositions (Brooks and Lozano-Perez 1985) produce piecewise linear paths. These paths consist of subsequent waypoints joined by straight lines. Potential field methods guide the robot toward its goal by applying attractive forces, toward the goal, and repulsive, away from obstacles (Khatib 1986). Potential field methods tend to produce oscillating paths in narrow passages (Koren and Borenstein 1991). Sampling-based motion planning algorithms, such as rapidly exploring random trees (RRT) (LaValle 2000) and probabilistic roadmap method (PRM) (Kavraki et al. 1996), rely on stochastic sampling to efficiently explore the search space. Resulting paths from randomization are suboptimal and require post-processing to improve their quality (Elbanhawi and Simic 2014c). Motion planning using state lattices is disadvantaged by discretization (Pivtoraiko et al. 2009; Pivtoraiko and Kelly 2011). Coarse discretization leads to loss of completeness, while high-fidelity subdivision increases the computational time of the planner, especially in highly dimensional scenarios. Homotopy class optimization of trajectories is proposed (Zucker et al. 2013). These methods do not discuss curvature continuity, and the performance is dependent on the optimization algorithm. Optimization methods are not immune from local minima and are not guaranteed to converge.

Agile robots, such as omnidirectional, differential-drive robots and quadrotors, are capable of traversing piecewise linear paths. Such paths require stationary turns, at every waypoint, to change heading toward the subsequent waypoint. This approach is inefficient with regard to time, energy, and jerk considerations. The motion of some robots, such as car-like vehicles and fixed-wing UAVs, is highly constrained. Nonholonomic robots must be considered in the planning procedure, as they cannot follow piecewise linear paths. Minimum turning radius constraints

impose further limitations on the path, which is often represented by maximum curvature restrictions.

Traditionally, Dubins paths or Reed's and Shepp's (Xuan-Nam et al. 1994; Reeds and Shepp 1990) are used in path smoothing for vehicles with minimum turning radius constraint in a two-dimensional space. Configurations are joined by sets of primitives consisting of circular arcs and straight lines. The amalgamation of circular arcs and lines results in discontinuities in curvature. Clothoids may appear to be suitable for path smoothing, as they are characterized by continuous curvature (Fraichard and Scheuer 2004). However, clothoid generation is challenging, as they have no closed-form expression. High-order splines (11th order) and polynomials (26th order) have been proposed for clothoid approximation (Wang et al. 2001; Meek and Walton 2004; Walton and Meek 2005; McCrae and Singh 2009; Montes et al. 2008). Recent work has enabled the real-time approximation of clothoids under bounded length and orientation limitations (Brezak and Petrovic 2013). Consequently, they are still not suited for real-time replanning and highly dimensional scenarios.

Curvature discontinuities result in overactuation, slipping, localization errors (Magid et al. 2006), passenger discomfort (Gulati and Kuipers 2008), mechanical wear and failure (Berglund et al. 2010; Maekawa et al. 2010), and control instability (Lau et al. 2009; Roth and Batavia 2002). Subsequently, achieving continuous curvature is advantageous in applications that involve carrying sensitive cargos such as human passengers (Gulati and Kuipers 2008) or heavy loads in mining applications (Berglund et al. 2010; Maekawa et al. 2010) and those which require precise localization such as agricultural applications to minimize the impact of the vehicle on crops (McPhee and Aird 2013; Sabelhaus et al. 2013; Alshaer et al. 2013) or energy loss minimization for MAVs with battery-size restrictions (Myung et al. 2007).

In our earlier work, we proposed an evaluating and bounding B-spline paths approximate solution. We show that humans control vehicles with continuous commands and generate paths that obey the vehicles kinematic constraints. We propose the premise of using a single B-spline curve to generate paths that resemble human driving and obey the vehicle's constraints. This is achieved by defining the curvature of a B-spline segment in terms of the parameters of its corresponding control polygon, which in this case is assumed to be a linear path generated by a path planning algorithm.

In this work we improve B-spline-based motion planning by proposing efficient methods for segment curvature evaluation and analytical bounding. The characteristics of B-splines are exploited to present two solutions for continuous curvature bounding, which can be combined together or used separately. The novelty of our proposal is that it is not limited to a plane or a dimension; it is not subject to orientation, length, or control polygon restrictions. It guarantees continuity throughout the path while preserving real-time performance. We also show that it is possible to plan the trajectory of a robot with nonholonomic constraints and maintain parametric continuity.

This chapter is organized as follows: Sect. 9.2 lists the current related research in path smoothing. The problem is formally described in Sect. 9.3. B-spline curve synthesis is introduced in Sect. 9.4. We address curvature continuity, segment curvature formulation, and curvature bounding in 2D in Sect. 9.5. Our findings are validated and compared with previous work using simulation experiments, as given in Sect. 9.6. The chapter is concluded in Sect. 9.7.

9.2 Related Work

There are two separate problems addressed in this chapter: firstly, planning a geometric curve with curvature bounds given in Sect. 9.4 and, secondly, maintaining parametric continuity of the generated trajectory. The authors could not ascertain any literature that combined these two problems. There are only approaches that address each issue separately. The benefits of synthesizing kinodynamically feasible and continuous paths are well studied in robotics (McPhee and Aird 2013; Magid et al. 2006; Gulati et al. 2009; Gulati and Kuipers 2008; Maekawa et al. 2010; Lau et al. 2009; Myung et al. 2007; Sabelhaus et al. 2013; Alshaer et al. 2013). However, current solutions given in the literature fail to guarantee C^2 continuity with curvature bounds for nonholonomic mobile robots.

Similarly, in trajectory generation literature, there are multiple solutions to optimally generate bounded trajectories for given geometric curves with regard to time (Balkcom and Mason 2002; Wu et al. 2000) and jerk considerations (Guarino Lo Bianco 2013). Other approaches considered curvature and acceleration bounds as parameters in optimization problems (Johnson and Hauser 2012; Kunz and Stilman 2013; Sachin et al. 2014). However these methods, adversely, provided no discussion on the parametric continuity problem and often led to C^1 continuity only. The approach developed by Velenis and Tsiotras (2008) for vehicles is limited to velocity continuity and acceleration bounds and ignored acceleration continuity which will undoubtedly lead to a jerky and uncomfortable ride (Gulati and Kuipers 2008; Guarino Lo Bianco 2013). Control laws proposed for unicycle robots ignored acceleration and curvature bounds (Lapierre et al. 2007; Sgorbissa and Zaccaria 2010; Morro et al. 2011).

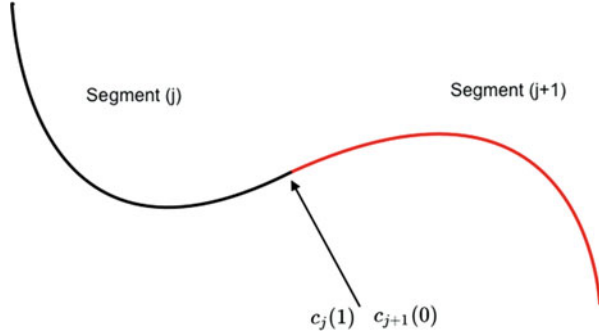
Dubins paths and circular arcs were commonly used for robot planning despite their curvature discontinuity. Dubins paths were generated under the assumption that the vehicle maintains constant linear velocity. Multiple circular segments have been proposed for UAV path smoothing (Anderson et al. 2005). This approach was limited to planar scenarios and produces discontinuous paths. Bézier curves were commonly used for path smoothing. The order of Bézier is dependent on the number of control points, which resulted in limiting them to maintain a low-order curve (Jolly et al. 2009; Lau et al. 2009; Kwangjin and Sukkarieh 2010). A comparative navigational-based analysis showed that Bézier curves poorly interpolate a linear path as opposed to B-splines, (Elbanhawi et al. 2014). Limiting the

number of control points of Bézier curves results in the need to join multiple curve segments. Discontinuities arise at the joint of two Bézier segments.

A condition for Bézier curve geometric, G^2 , continuity was presented in (Walton et al. 2003). However, this condition had no closed-form solution. Kwangjin et al. (Kwangjin and Sukkarieh 2010; Kwangjin et al. 2013) provided a solution for a particular case of the G^2 planar condition. This approach was used for upper-bounded curvature smoothing algorithms. It is still limited to a plane and, in fact, incapable of considering different curvature bounds in horizontal and vertical planes. Fixed-wing UAVs have different turning angle (horizontal plane) and climbing angle (vertical plane) limitations. Barsky and Derose (1990) proposed geometric continuity, G^k , as condition for ensuring that curve endpoints had the same directions, not the values. The work in (Kwangjin and Sukkarieh 2010; Kwangjin et al. 2013) fails to guarantee velocity and acceleration continuity, which are more realistic for robotics than geometric continuity. Pan et al. (2012) have shown that only C^2 parametric continuities of acceleration and velocity are suitable for real robots and provided a shortcutting algorithm that guarantees continuity in most scenarios but fails to address the maximum curvature constraint. Recent studies investigated trajectory planning for trailer cars with continuous velocities (Ghilardelli et al. 2014).

The advantages of B-splines for real-time planning have been shown (Dyllong and Visioli 2003; Elbanhawi and Simic 2012). A genetic algorithm was employed to select the location of a fixed number of control points, for a single B-spline curve (Nikolos et al. 2003). This guaranteed the continuity of the curve. However, having a constant number of control points reduced the robustness of the generated path. B-splines were used for generating smooth paths for passenger transporting robots (Gulati and Kuipers 2008). That approach is limited to a 2D setting and robots with no curvature bounds. Similarly, a 3D B-spline smoothing algorithm was presented that did not consider curvature continuity or upper bounds (Koyuncu and Inalhan 2008). Several optimization algorithms are limited to 2D offline B-spline smoothing and curvature bounding (Berglund et al. 2010; Maekawa et al. 2010). A B-spline shortcutting algorithm was proposed, which used multiple segments; however, it did not guarantee continuity in all segments and did not consider the maximum curvature (Pan et al. 2012). In our earlier work, we provided an approximate B-spline-based approach to the presented problem and did not consider acceleration and velocity bounds (Elbanhawi et al. 2014). Research findings were implemented, in real time, on experimental vehicle (Elbanhawi and Simic 2014b). On the other hand, in here we present solution of the problem analytically and consider kinodynamic constraints.

Fig. 9.1 Joining two path segments



9.3 Problem Statement

We consider front wheel-steered vehicles, referred to as car-like robots. It is common to model the vehicle using the bicycle model, which assumes identical steering angles on both sides (Jazar 2008), as shown in Fig. 9.1. This model has been shown to be adequate for modeling the global kinematic motion of front wheel-steered vehicles (Campion et al. 1996). The advantage of our approach is in solving two closely related problems simultaneously, which are generally decoupled in robotic literature. We aim to synthesize a curve that satisfies conditions given by Eqs. (9.1, 9.2, 9.3, and 9.4) and maintain parametric continuity. The vehicle’s Cartesian coordinates (x, y) and heading angle θ are measured from the center of the rear axle relative to a global frame. The length between the front and back wheel is referred to as wheel base, W . The two actuation commands are linear velocity, v , and steering angle Φ . It is clear that the vehicle is underactuated as it has two controls and three degrees of freedom, i.e., it is not fully controllable (Ogata 2010). The velocity components in the x and y directions, v_x and v_y , are constrained as given in Eq. 9.1. This nonholonomic condition is often referred to as the rolling without slipping constraint.

$$v_x \sin(\theta) - v_y \cos(\theta) = 0 \tag{9.1}$$

Consider a planning algorithm that produces a path consisting of, $n-1$, straight lines joining successive, n , waypoints, $P = [P_1, P_2, \dots, P_n]$, where $P_i = (P_{xi}, P_{yi}, P_{zi})$ for $i = [1, 2, \dots, n-1, n]$. It is required to generate a curve, $c(u)$, which closely follows straight-line path, where u is the normalized path length parameter. It is an independent variable in the range of $[0,1]$ for any curve, $c(u)$. Parameter u takes the value $u = 0$ at the beginning of the segment and reached the value $u = 1$ at the end. The generated curve must satisfy the following imposed constraints.

Path continuity at the endpoints of two curve segments must be addressed; such a situation is illustrated in Fig. 9.1. For two consecutive curve segments $c_j(u)$ and $c_{j+1}(u)$, C^k parametric continuity could be then defined as shown in Eq. 9.2, according to Farin (2002), where k is a positive integer denoting the order of the parametric continuity.

$$\frac{\partial^i c_{j+1}(0)}{\partial u^i} = \frac{\partial^i c_j(1)}{\partial u^i}, \forall i = 1, 2, 3..k \tag{9.2}$$

The curvature of the path must not exceed the maximum curvature, K_{max} , at any point. The curvature, $k(u)$, along a path is defined as Eq. 9.3, where $c(u) = [x(u), y(u)]$ and the first- and second-order derivatives with respect to u are $c'(u) = [x'(u), y'(u)]$ and $c''(u) = [x''(u), y''(u)]$.

$$k(u) = \frac{x'(u)y''(u) - x''(u)y'(u)}{(x'(u)^2 + y'(u)^2)^{3/2}} \tag{9.3}$$

The minimum radius of curvature, r_{min} , in a plane restricts the curvature of the path to K_{max} in that plane. For car-like robots, the curvature constraint is a result of the maximum steering ϕ_{max} angle due to the mechanical construction of the vehicle, as shown in Eq. 9.4. In three-dimensional scenarios for aerial vehicles, the maximum yaw and pitch angles in the horizontal and vertical planes must be considered separately (Fig. 9.2).

$$k_{max} = \frac{1}{r_{min}} = \frac{\tan(\phi_{max})}{W} \tag{9.4}$$

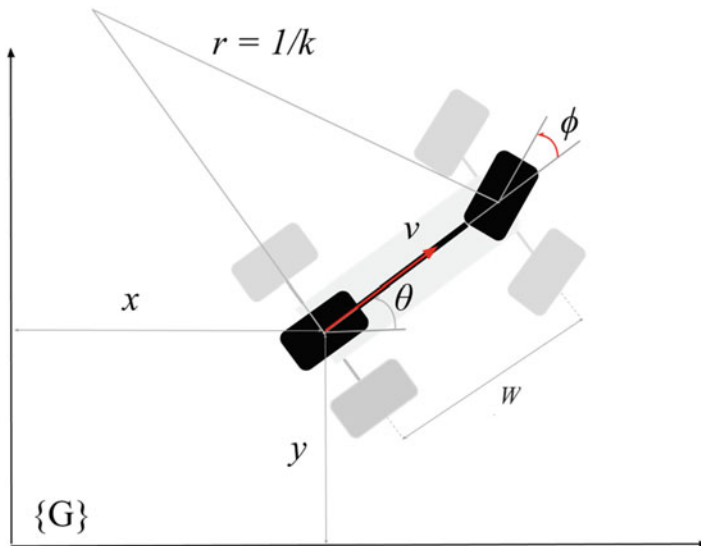


Fig. 9.2 Bicycle model for front wheel-steered vehicles

9.4 Spline Primitives

B-splines are vector-valued parametric curves, initially proposed by Schoenberg (1946). B-splines and nonuniform rational B-splines (NURBs) are commonly used for computer-aided design (CAD) applications as a result of their efficient synthesis and robustness (Farin 1992). In addition to CAD, they have been utilized in reverse engineering (Ma and Kruth 1998; Piegl and Tiller 2001), finite element analysis (Hughes et al. 2008), machining (Cheng et al. 2002; Sungchul and Taehoon 2003), medical imaging (Zhang et al. 2007), computer vision (Biswas and Lovell 2008), bio-inspired data fitting (Jones and Adamatzky 2014), and signal processing (Unser et al. 1993). As discussed in earlier sections, their use in robotics is fairly recent.

A p -th degree B-spline curve, $c(u)$, is defined by n control points and a knot vector \hat{u} , evaluated by Eq. 9.5. The length of the one-dimensional knot vector, m , is equal to $n + p + 1$. Normalized path length parameter, u , is simply referred to as the path parameter (Farin 2002).

$$c(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad (9.5)$$

P_i is the i -th control point, which is in turn influenced by a corresponding basis functions. The number of basis functions therefore mirrors the number of control points, n . $N_{n,i}(u)$ is the i -th B-spline basis function, which is defined using the Cox-de Boor recursive algorithm (De Boor 1972). First-order basis functions are evaluated using Eq. 9.6 based on the predefined knot vector. Higher-order functions are computed by the recursive substitution in Eq. 9.7.

$$N_{i,0}(u) = \begin{cases} 1 & u \in [\hat{u}_i, \hat{u}_{i+1}) \\ 0 & \text{else} \end{cases} \quad (9.6)$$

$$N_{i,p}(u) = \frac{u - \hat{u}_i}{\hat{u}_{i+p} - \hat{u}_i} N_{i,p-1}(u) + \frac{\hat{u}_{i+p+1} - u}{\hat{u}_{i+p+1} - \hat{u}_{i+1}} N_{i+1,p-1}(u) \quad (9.7)$$

We have previously shown B-spline properties that render them as superior to other parametric curves, for the task of robot navigation (Elbanhawi et al. 2014). The curve's degree, p , is independent of the number of control points, n . This allows the possibility of using a single curve for the entire path smoothing without imposing limitations on the number of control points. It is in contrast to Bézier curve methods (Jolly et al. 2009; Kwangjin and Sukkarieh 2010; Lau et al. 2009; Kwangjin et al. 2013) where the number of control points is predefined. Modification of control points affects the curve shape locally and does not change the rest of the path. This enables the local control of the path for smoothing or obstacle avoidance purposes. A clamped B-spline curve follows its control polygon more closely in comparison to a Bézier curve of the same order. Clamping is achieved by having $(p + 1)$ multiplicity of the initial and final knots, \hat{u} (Farin 2002). Knot multiplicity ensures that the curve passes through the initial and final control points.

Despite the beneficial properties that characterize B-splines, maintaining path continuity and controlling its curvature are nontrivial issues. They continue to challenge the use of B-splines in robotic path planning applications (Lau et al. 2009; Elbanhawi and Simic 2014a; Pan et al. 2012). It must be noted that both Beziers and B-spline are essentially combinations of polynomials. In principal, there should exist control laws, or conditions, that are capable of generating parametrically continuous trajectories using Bezier curves as well. The authors could not identify such methods in literature. Consequently, we have utilized the existing benefits of B-splines for motion planning.

9.5 Curvature Bounding

9.5.1 Parametric Continuity

The challenge of path continuity stemmed from the linking of two separate path segments. Primitives such as circular arcs, polynomials, and clothoids were not flexible enough to represent a path using a single segment. The number of control points, which were usually predefined prior to smoothing, governs the order of a Bézier curve. Consequently, multiple Bézier curves must be linked for smoothing a single piecewise linear path.

The order of a B-spline curve is independent of the number of control points in the path, as already mentioned. In theory, it is possible to smooth a path using a single curve of a predefined order. The single B-spline curve approach was adopted for UAV planning; however, the number of control points was fixed (Nikolos et al. 2003). The region in which planning is conducted and path shape robustness are significantly limited by fixing the number of control points. The work by Jolly et al. (2009) is based on rapid replanning with a short planning horizon and relies on four control points. We did not pose any restrictions on the number of control points, apart from that the number of control points must exceed the degree of the curve, p . The local control property of B-spline enables the modification of a curve segment without changing the entire path. The necessity for rerouting commonly results from obstacle detection or smoothing purposes.

Despite the superiority of B-splines over a Bézier curve of the same order, in closely following the shape of a path, they still deviate from the original path (control polygon). Ideally, the curve would follow the original linear path and smoothly cut corners when turning is needed. It is desired to maintain proximity to the originally planned straight-line path as it is more likely to be collision-free. This was achieved by forcing the tangency of the curve to the sides of the control polygon. B-spline tangency to collinear control points is leveraged to ensure the close following of the original path. Systematic midpoint insertion, between every two successive points, effectively transformed control polygon edges into lines connecting three control points, thus forcing the curve's tangency to the edges. The effect of midpoint insertion is illustrated in Fig. 9.3. It is worth highlighting that in

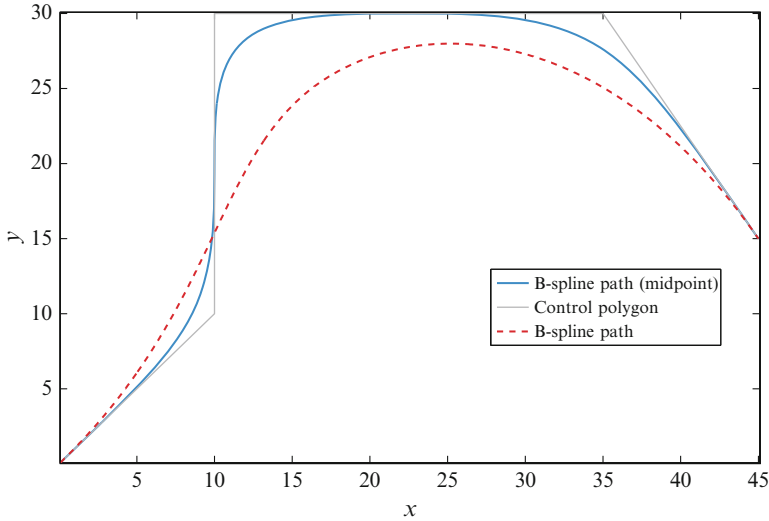


Fig. 9.3 Midpoint insertion improves the path proximity of B-splines without compromising parametric continuity. It forces the curve (blue) tangency to the edge of the control polygon (black) unlike the unmodified B-spline curve (red)

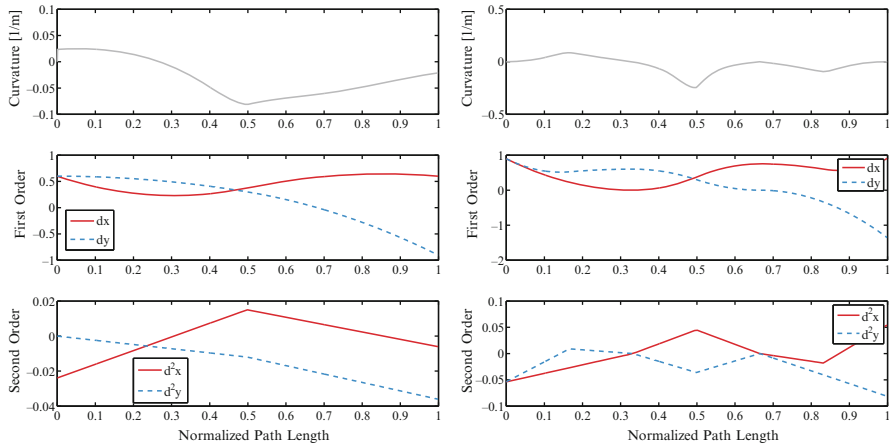


Fig. 9.4 Parametric continuity was maintained before (left) and after (right) midpoint insertion as a result of using a single B-spline segment

both cases, a single curve segment was used for smoothing. That guarantees continuity along the path. This avoids the need to address parametric continuity at union points, as illustrated in Fig. 9.1. The curvature and higher-order derivatives do not exhibit any abrupt changes after adding midpoints. This can be validated by comparing the resulting trajectories given in Fig. 9.4.

9.5.2 Curvature Evaluation

9.5.2.1 Path Segmentation

Our aim was to control the curvature, k , of a B-spline curve. Specifically, it was required to maintain the curvature below the maximum curvature bound, K_{\max} . Systematic midpoint insertion allowed for the definition of a repeated segment throughout the path (see for Fig. 9.5(a) illustration). The segment consists of two intersecting control edges and a total of five control points (including two mid-points). It was required to define B-spline paths curvature in terms of their corresponding segment parameters. This enabled the isolation of each segment and local modifications of its parameters by leveraging the local support property of B-splines. Smoothing modifications will be proposed to ensure maximum curvature bounds are obeyed.

The parameters of the reoccurring control segment are the side length, L , the angle between segment sides, α , and the length ratio of both sides, r , as illustrated in Fig. 9.5(b). In our earlier work (Elbanhawi et al. 2014), segments of equal sides were assumed, $r = 1$, which overestimated the curvature of the path and resulted in attaining approximate solutions. The use of the length ratio parameter, r , is presented to enable a more precise evaluation of the curvature. Position vectors describing the five control points of the segment can be defined with respect to the parameters of the same segment and are given in Eq. 9.8.

$$P \begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} L, \frac{L}{2}, 0, r\frac{L}{2} \cos(\alpha), rL \cos(\alpha) \\ 0, 0, 0, r\frac{L}{2} \sin(\alpha), rL \sin(\alpha) \end{pmatrix} \quad (9.8)$$

The cubic B-spline curve, $p = 3$, has five control points, $n = 5$, and, $m = 9$, knots with four initial and final multiplicity for clamping, $\hat{u} = [0,0,0,0,0.5,1,1,1,1]$. Initial order basis functions were evaluated using Eq. 9.6. Following that, basis functions, $N(u)$, were computed, using the Cox-de Boor algorithm by recursive evaluation of

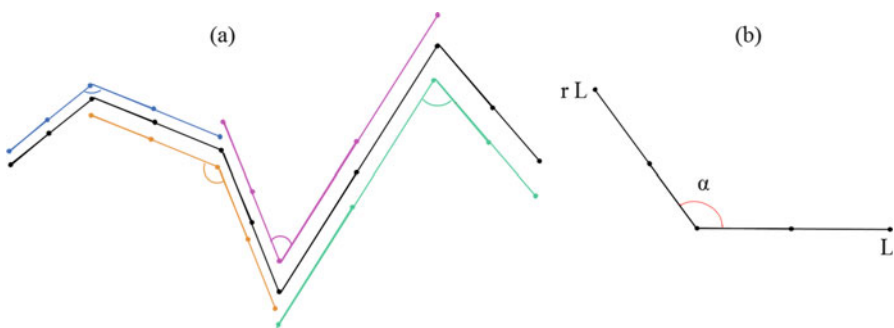


Fig. 9.5 (a) The notion of a reoccurring control segment through the path. A segment consists of two intersecting straight lines and five control points. (b) The parameters of a single segment

Eq. 9.7. The following third-order basis functions can be defined as given in the set of Eq. 9.9.

$$N_{0,3} = \frac{(1 - 2u)^3}{2} \tag{9.9a}$$

$$N_{1,3} = 6u^3 - 6u^2 + 1 \tag{9.9b}$$

$$N_{2,3} = -6u^2 + 6u - 1 \tag{9.9c}$$

$$N_{3,3} = -6u^3 + 12u^2 - 6u + 1 \tag{9.9d}$$

$$N_{4,3} = \frac{(2u - 1)^3}{2} \tag{9.9e}$$

In order to define the curvature of a segment in terms of its parameters, $k = f(r, L, \alpha)$, the position vectors of the segment, Eq. 9.8, and basis functions, Eq. 9.9, were substituted in the curve Eq. 9.5. The curve was defined as a function of its corresponding segment parameters, $c(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \begin{bmatrix} x(r, L, \alpha) \\ y(r, L, \alpha) \end{bmatrix}$; $x(u)$ and $y(u)$ are given in Eq. 9.10.

$$x(u) = \frac{(1-2u)^3}{2} * L + (6u^3 - 6u^2 + 1) * \frac{L}{2} + (-6u^3 + 12u^2 - 6u + 1) * \frac{rL \cos(\alpha)}{2} \frac{(2u - 1)^3}{2} * rL \cos(\alpha) \tag{9.10a}$$

$$y(u) = (-6u^3 + 12u^2 - 6u + 1) * \frac{rL \sin(\alpha)}{2} + \frac{(2u - 1)^3}{2} * rL \sin(\alpha) \tag{9.10b}$$

For a given segment, its parameters, r, L , and α , are constant and known prior to a curvature query. The first- and second-order derivatives with respect to the path parameter, u , are derived below from equations set (Eq. 9.11).

$$x'(u) = 3L(u^2(r \cos(\alpha) - 1) + 2u + 1) \tag{9.11a}$$

$$x''(u) = 6L(u(r \cos(\alpha) - 1) + 1) \tag{9.11b}$$

$$y'(u) = 3Lr \sin(\alpha)u^2 \tag{9.11c}$$

$$y''(u) = 6Lr \sin(\alpha)u \tag{9.11d}$$

The curvature expression, $k = f(r, L, \alpha)$, in Eq. 9.12 was obtained by substituting the curve and its first- and second-order derivatives from Eq. 9.11 into Eq. 9.3. It can be noted that when substituting by $r = 1$, in Eq. 9.12 we get the same expression derived in Elbanhawi et al. (2014). Prior to introducing the parameter, r , curvature evaluations were approximate, and the accuracy of the maneuvers could not be ascertained.

$$k(u) = \frac{2ru(u - 1) \sin(\alpha)}{3L(u^4(r^2 - 2r \cos(\alpha) + 1) + 4u^3(r \cos(\alpha) - 1) - 2u^2(r \cos(\alpha) - 3 - 4u + 1))^{3/2}} \tag{9.12}$$

9.5.2.2 Segment Curvature Evaluation

Midpoint insertion ensured the curve’s tangency to the polygon edges, which resembled paths generated by human operators in the experiments conducted by Elbanhawi et al. (2014). Subsequently, the curvature of the path started at $u = 0$ and finished at $u = 1$, with $k = 0$. Curvature peaked to k_{peak} at some point, u_{peak} , in between, $u = [0,1]$. In order to limit path curvature to the maximum value of K_{max} , the peak curvature, k_{peak} , of the segments must be evaluated first. The point, u_{peak} , along the parametric path length, u , where the curvature peaks, was found by solving Eq. 9.13. Then k_{peak} was computed by substituting u_{peak} in Eq. 9.12.

$$\frac{dk(u)}{du} = 0 \tag{9.13}$$

For every path segment, there exists a singular curvature peak, as shown in Fig. 9.6. The red profiles show the influence of changing the segment angle while maintaining fixed length and ratio. The location of the peak curvature was entirely dependent on the length and ratio. For a large angle (blue) and fixed length, the ratio changed both the position and value of the peak curvature. Similarly, for a much smaller segment angle (gray), the length ratio was still influential on both the peak value and position.

Solving Eq. 9.13 for u_{peak} can prove to be a computationally intensive task, particularly when k_{peak} had to be evaluated multiple times during each query of the path planning procedure. One useful observation is that the location of u_{peak} is dependent on the segment angle, α , and length ratio, r , as highlighted in Fig. 9.7. We note that, while u_{peak} is dependent on r and α only, the peak curvature value, k_{peak} , is still dependent on r , α , and L . It was possible to store u_{peak} values in a lookup table of equal intervals from $r = 1$ to 10 and $\alpha = 0$ to π . The required values

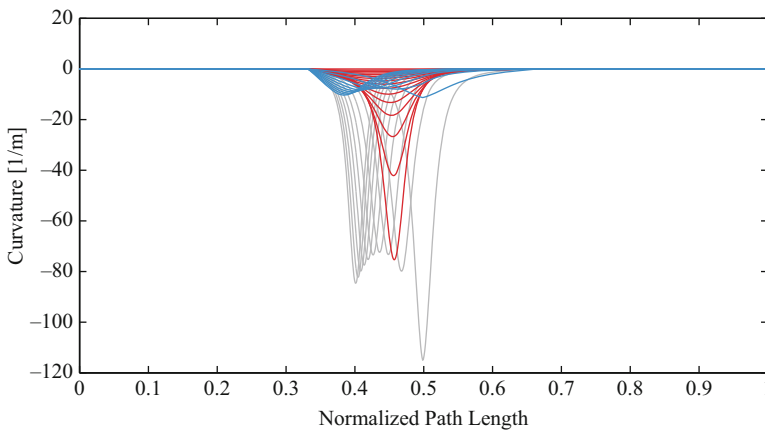


Fig. 9.6 Changing segment parameters shifts the position of the curvature peaks. In all cases, curvature profile is continuous with a singular peak

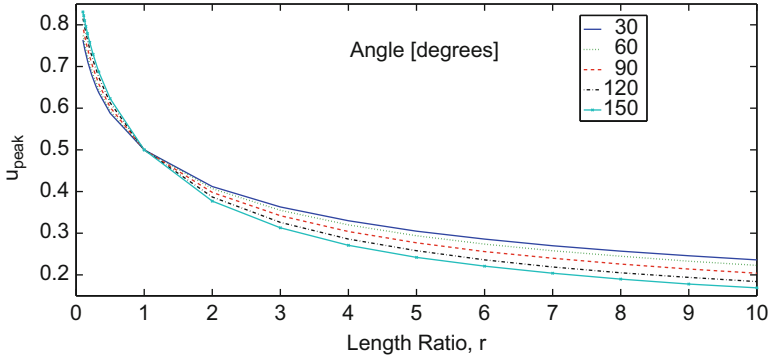


Fig. 9.7 Parametric length location, u_{peak} , of the peak curvature, k_{peak} , is dependent on the segment angle, α , and the length ratio, r . It can be noted that when length ratio is $0 < r < 1$, $u_{\text{peak}} > 0.5$ and when $r > 1$, $u_{\text{peak}} < 0.5$. This results from the observation that u_{peak} is shifted toward the shorter segment edge

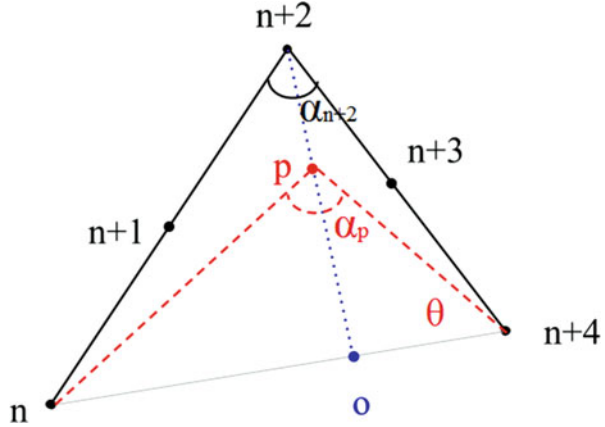
can be interpolated. To maintain a sparse lookup table, we use the property in Eq. 9.14, which can be observed from Fig. 9.7. In our case, retaining a lookup table (*less than 10kB in size*) produced curvature values of 10^{-3} accuracy.

$$\text{If } 0 < r < 1, \text{ then } u_{\text{peak}(r,\alpha)} = 1 - u_{\text{peak}(1/r,\alpha)} \quad (9.14)$$

9.5.3 Curvature Bounding

In this section, two analytical solutions for curvature bounding are presented. They ensured peak segment curvature does not exceed the maximum curvature, $k_{\text{peak}} \leq K_{\text{max}}$. This confirms that the path is feasible, having shown in the previous section that each path segment has a single peak. The first solution was relaxed ensuring a smooth curvature. The second solution was strict to minimize deviation from the original control polygon. It was possible to combine both conditions in different segments, on account of B-spline local support property, with minimal effect on other segments. Both conditions were designed to make certain that the path was contained within the convex hull of the original control polygon to reduce the probability of the obstacle collision. Both solutions are essential homotopy class transformation to ensure feasibility. Nonetheless, the guarantee that the path is collision-free was not addressed in this work. We assume that this work will eventually be combined within a planning framework and will not be restricted to path smoothing.

Fig. 9.8 First smoothing solution; it is required to find the point (P) along the line (dotted blue line), joining point ($n + 2$) and point (o), that ensures the curvature, k_{peak} , does not exceed K_{max}



9.5.3.1 Single-Peak Solution

Consider the single control segment, shown in Fig. 9.8, whose corresponding B-spline curvature violates the maximum curvature condition. The segment consists of two lines $l_{n,n+2}$, joining point (n) and point ($n + 2$), and $l_{n+2,n+4}$, joining point ($n + 2$) and point ($n + 4$), shown as solid black lines. Point (o) is the intersection point between $l_{n,n+4}$ (thin gray line) and line $l_{o,n+2}$ (dotted blue line) which is passing through point ($n + 2$) and is orthogonal to $l_{n,n+4}$.

The current curvature, k_{n+2} , and segment angle, α_{n+2} , are known, and $k_{n+2} > K_{max}$. Assume that point ($n + 2$) is shifted toward point (o), along the line, $l_{o,n+2}$, while points (n) and ($n + 4$) are unchanged and the midpoints ($n + 1$) and ($n + 3$) are recomputed accordingly. Finally at $\alpha_{n+2} = \pi$, $k_o = 0$. It is required to find the nearest point (p), at which $k_p = K_{max}$, as point ($n + 2$) is being shifted toward (o) along $l_{o,n+2}$. The minimum angle α_p lies between $\alpha_o = \pi$ and α_{n+2} as given by Eq. 9.15. We define $\overline{l_i, j}$ as the Euclidean distance between two points (i) and (j) whose Cartesian coordinates are known.

Assuming line $l_{o,n+2}$ is parameterized between P_{n+2} and P_o using $\hat{l}=[0,1]$, the value of \hat{l} is required where the point (p) satisfies the curvature requirement. Firstly, P_p is given as follows:

$$P_p = P_{n+2}(1 - \hat{l}) + P_o\hat{l} \tag{9.15}$$

In every iteration, the curvature is evaluated until the $k_p = K_{max}$ condition is satisfied. To optimize the search, we can estimate the initial point where the curvature may be equal to K_{max} . This is achieved by knowing that, at $\hat{l} = 0$, $k = k_p$ and, at $\hat{l} = 1$, $k = 0$.

Fig. 9.9 Bounding using single-peak solution. The original path is *blue* and new path is *red*

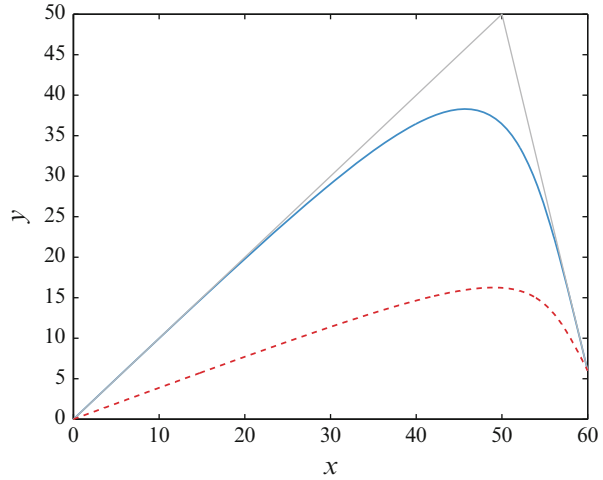
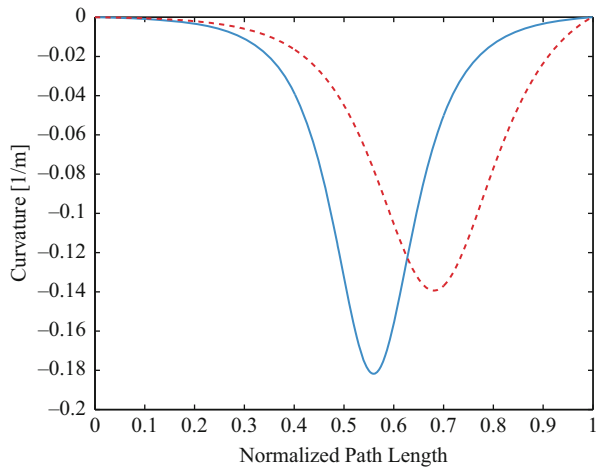


Fig. 9.10 Resulting curvature profiles before (*blue*) and after (*red*) bounding



$$\hat{l}_{init} = 1 - \left| \frac{K_{max}}{K_p} \right| \tag{9.16}$$

An example of curvature bounding is shown in Fig. 9.9 using this solution. The resulting curvature has a single segment as shown in Fig. 9.10 and was bound to 0.14 m^{-1} . Curvature continuity was maintained in both cases.

9.5.3.2 Double-Peak Solution

In this section we proposed a different approach for the same problem considered in the previous section. The curvature of a control segment, P_1 , P_0 , P_5 and their

Fig. 9.11 Second smoothing solution; it is required to find the value of β that ensures curvature bounding in both segments and minimizes the total path length

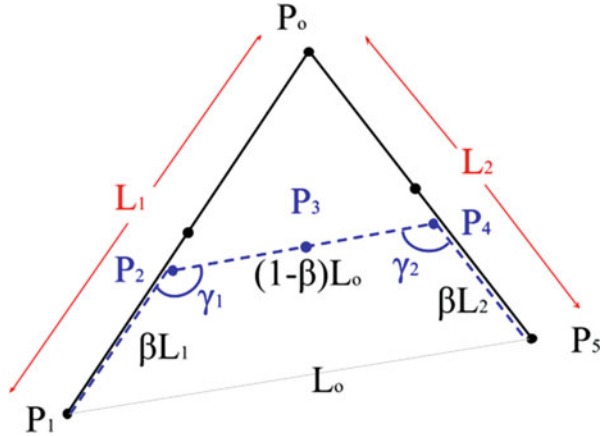


Table 9.1 Comparing related methods

Method	Curvature bounds	Curve	Path	Continuity
Nikolos et al. (2003)	Yes	B-spline	3D	C^2
Anderson et al. (2005)	Yes	Arcs/line	2D	No
Gulati and Kuipers (2008)	No	B-spline	2D	No
Koyuncu and Inalhan (2008)	No	B-spline	3D	No
Jolly et al. (2009)	No	Bézier	2D	C^2
Lau et al. (2009)	No	Bézier	2D	No
Berglund et al. (2010)	Yes	B-spline	2D	No
Maekawa et al. (2010)	Yes	B-spline	2D	No
Pan et al. (2012)	No	B-spline	3D/2D	C^2/C^1
Kwangjin (2013)	Yes	Bézier	2D	G^2
Huh and Chang (2014)	No	Polynomial	2D	G^2
Proposed	Yes	B-spline	2D/3D	C^2

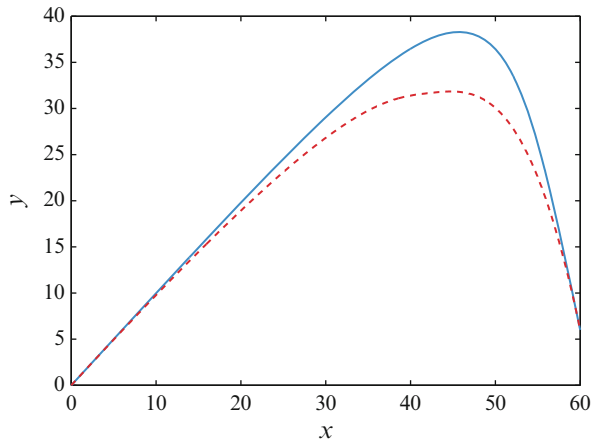
midpoints in Fig. 9.1 exceeds K_{max} . Segment P_1, P_0, P_5 is decomposed into two segments, P_1, P_2, P_4 (segment 1) and P_2, P_4, P_5 (segment 2). Line segment $\overline{P_2P_4}$ is constructed to be parallel to edge $\overline{P_1P_5}$. As a result, triangles $\Delta P_1P_0P_5$ and $\Delta P_2P_0P_4$ are similar, and the ratio between their side lengths is $(1-\beta)$, where $0 < \beta < 1$ (Fig. 9.11). Segment 1 and 2 parameters can be described in terms of β , where segment angles are constant, as given in Table 9.1.

By substituting the segment parameters, given in Table 9.2, in Eq. 9.12, it is possible to find a range for β , subset of set $[0,1]$, in which both segment curvatures are less than K_{max} . Firstly, we compute a separate range for each segment 1 and 2 $[\beta_{min1}, \beta_{max1}]$ and $[\beta_{min2}, \beta_{max2}]$. These computations are efficient by virtue of using the lookup table in the previous section. The allowable range for β is $[\max(\beta_{min1}, \beta_{min2}), \min(\beta_{max1}, \beta_{max2})]$.

Table 9.2 Segment parameter

Parameter	Symbol	Segment 1	Segment 2
Length ratio	r	$\frac{(1-\beta)L_0}{\beta L_1}$	$\frac{(1-\beta)L_0}{\beta L_2}$
Edge length	L	βL_1	βL_2
Segment angle	α	γ_1	γ_2

Fig. 9.12 Bounding using double-peak solution. The original path is *blue* and the feasible path is *red*



We nominate the β value that minimizes the total length. Now, new segment control points P_2, P_3, P_4 can be computed, where for any control point we have $P_i = (x_i, y_i)$.

$$\beta = \operatorname{argmin}(\beta(L_1 + L_2 - L_0) + L_0), \beta \in [\beta_{\min}, \beta_{\max}] \tag{9.17}$$

$$P_2 = \beta P_1 + (1 - \beta)P_0 \tag{9.18}$$

$$P_4 = \beta P_5 + (1 - \beta)P_0 \tag{9.19}$$

A midpoint is inserted between the two added points based on the ratio between the lengths of both, such that if both lines are equal, $r = 1$; the midpoint is equidistant between them.

$$P_3 = \frac{r}{r + 1}(P_4 - P_2) + P_2 \tag{9.20}$$

An example of curvature bounding is shown in Fig. 9.12 using this solution. The resulting curvature has two segments as shown in Fig. 9.13 and was bound to 0.14 m^{-1} . Curvature continuity was maintained in both cases.

Fig. 9.13 Resulting curvature profiles before (blue) and after (red) bounding

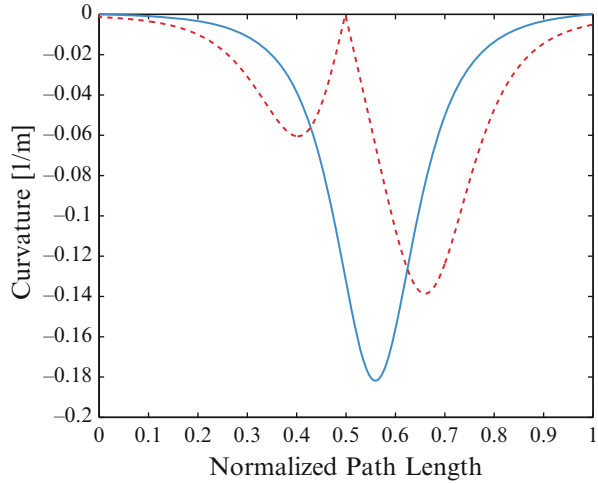


Table 9.3 Curvature evaluation time performance for 1000 queries

Query time	Analytical solution	Lookup table
Mean (ms)	122.16	0.91
Standard deviation (ms)	6.95	0.34

9.6 Results

9.6.1 Curvature Evaluation

To efficiently evaluate a segment’s curvature, we proposed storing the peak curvature position u_{peak} in a sparse lookup table and evaluating the curvature using the segment parameters. We conducted 1000 queries, for a range of segment parameters where r and $L = [1 \text{ m}, 10 \text{ m}]$ in steps of 1 m and α was $= [30^\circ, 180^\circ]$ in steps of 15° . The time performance of this evaluation method was compared with solving Eq. 9.13. From the results, given in Table 9.3, it is clear that this method is more efficient. Comparing with previously published research results Elbanhawi et al. (2014), which assumed equal segment length, we show that this approach has better accuracy. The results are illustrated in Fig. 9.14 and given in Table 9.4.

9.6.2 Curvature Bounding

In this section we compared the presented bounding solutions to our earlier work in Elbanhawi et al. (2014). Two different examples were used as shown in Figs. 9.15 and 9.18. The linear reference paths are assumed to result from a planning

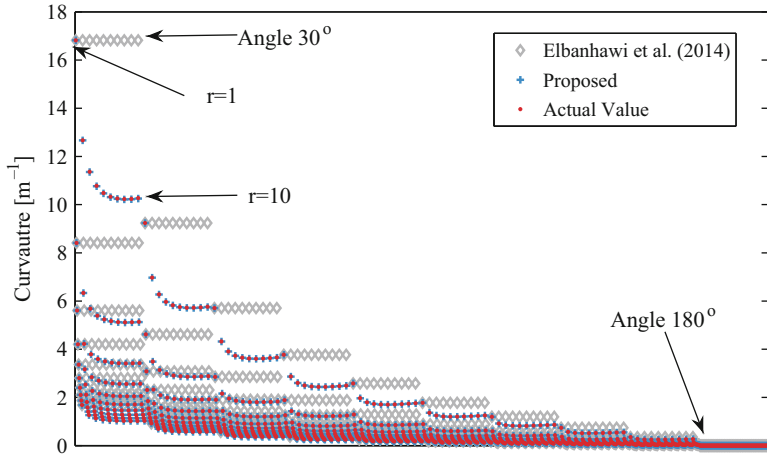


Fig. 9.14 Curvature evaluation errors of proposed lookup table compared to Elbanhawi et al. (2014)

Table 9.4 Curvature evaluation errors

Error [mm ⁻¹]	Proposed	Elbanhawi et al. (2014)
Mean	0.87	384.25
Standard deviation	1.738	788.45
Maximum	19.34	6592.10

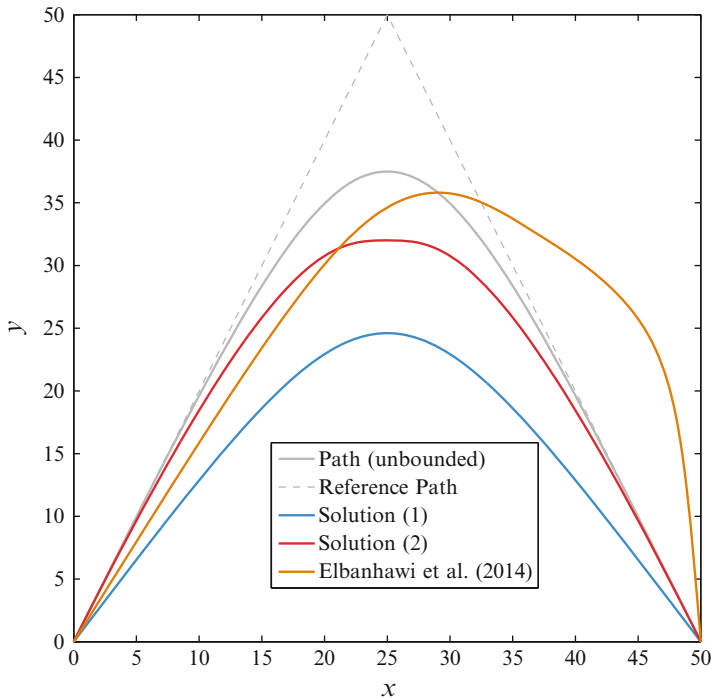


Fig. 9.15 Example 1: Bounding paths using different methods

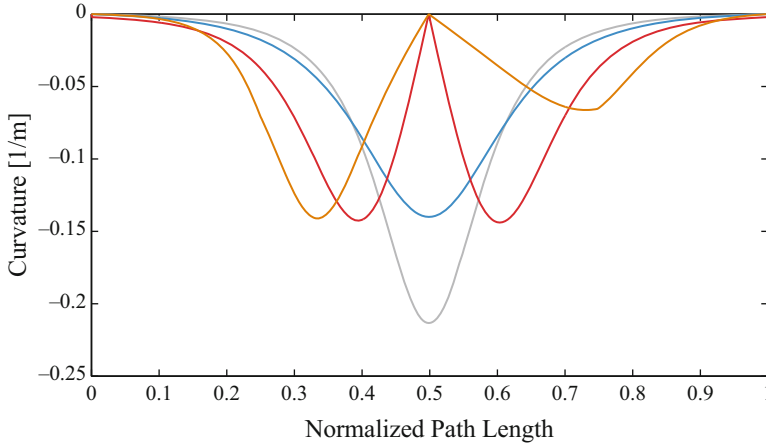


Fig. 9.16 Example 1: Resulting curvature profiles

algorithm. It can be noted that the proposed solutions maintain the curve within the convex hull of the original reference path. In both cases, the curvature is successfully bounded to 0.2 and 0.15 m^{-1} successively, and its continuity is maintained as shown in Figs. 9.16, 9.17, and 9.18. The proposed solutions reduce the deviation from the original path and the total path length, outperforming our earlier work as detailed in Tables 9.5 and 9.6. Solution (1) results in a low-frequency single-peak curvature profile as opposed to solution (2), which may have a better impact on passenger comfort in autonomous cars, as suggested in (Gulati and Kuipers 2008; Turner and Griffin 1999). On the other hand, solution (2) minimizes deviation from the reference paths and as a consequence minimizing the risk of collision.

Example (3) highlights the ability of the proposed method to generate a feasible path among obstacles. The benefit of maintaining the curve within the convex hull of the path is apparent in this example. The linear path was generated from a rapidly exploring random tree (RRT) algorithm (Elbanhawi and Sivic 2014b). The resulting B-spline path among obstacles is illustrated in Fig. 9.19. Post-processing RRT algorithms have been shown to improve path quality and produce fairly consistent results. Nonetheless these methods do not guarantee that the path is collision-free. The resulting trajectory is given in Fig. 9.20. It is clear that the multi-segment path maintains curvature and parametric continuity.

9.7 Conclusion

An approach to continuous curvature robot path smoothing that satisfies the maximum curvature bounds and parametric continuity is presented here. B-spline curves have been proposed for this task. In this chapter we offer the following contributions:

Fig. 9.17 Example 2: Bounding paths using different methods

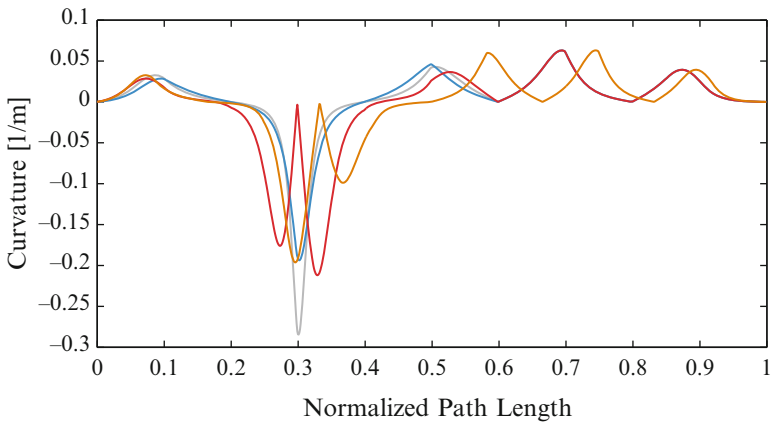
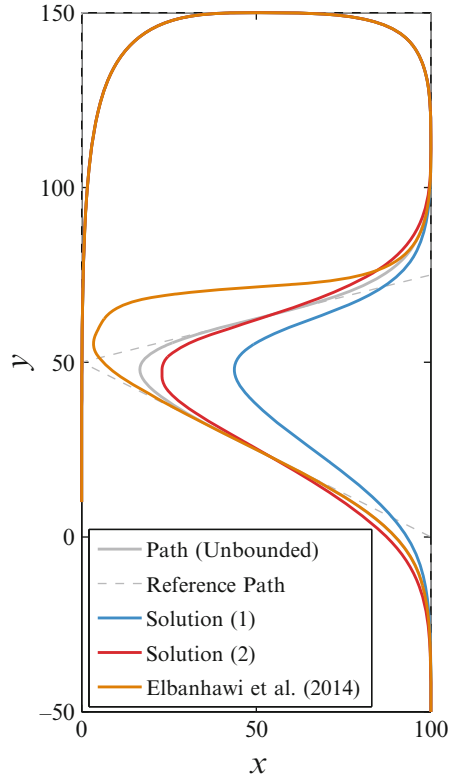


Fig. 9.18 Example 2: Resulting curvature profiles

Table 9.5 Example 1: Resulting path lengths and deviation

Path	Length [m]	Deviation mean [m]	Deviation maximum [m]
Linear path	94.33	–	–
Reference B-spline	92.28	–	–
Elbanhawi et al. (2014)	112.59	12.48	17.19
Solution (1)	71.78	8.58	12.89
Solution (2)	84.42	2.94	5.50

Table 9.6 Example 2: Resulting path lengths and deviation

Path	Length [m]	Deviation mean [m]	Deviation maximum [m]
Linear path	579.88	–	–
Reference B-spline	507.35	–	–
Elbanhawi et al. (2014)	535.28	20.15	45.04
Solution (1)	464.47	6.48	27.03
Solution (2)	506.27	6.04	23.99

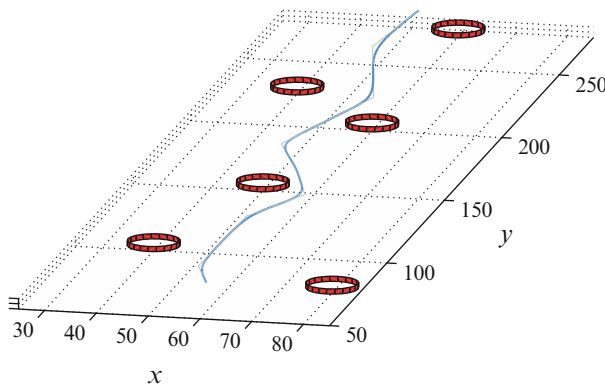


Fig. 9.19 Example 3: Kinodynamic motion among obstacles

- Maintaining path parametric C^2 continuity, by using a single B-spline curve segment with midpoint insertion, to generate more realistic robot paths (Pan et al. 2012). No limitations were posed on the number of control points, for the B-spline curve, enabling a more robust representation of the path, unlike the work in (Nikolos et al. 2003; Jolly et al. 2009).
- Two analytical solutions are offered, formulating the path curvature in terms of a predefined path segment’s parameters. They modify the path to limit its curvature to the maximum kinodynamic curvature and satisfy the vehicle’s constraints. Our previous publications presented an introduction to the more advanced solutions (Elbanhawi et al. 2014; Elbanhawi and Simic 2014b).

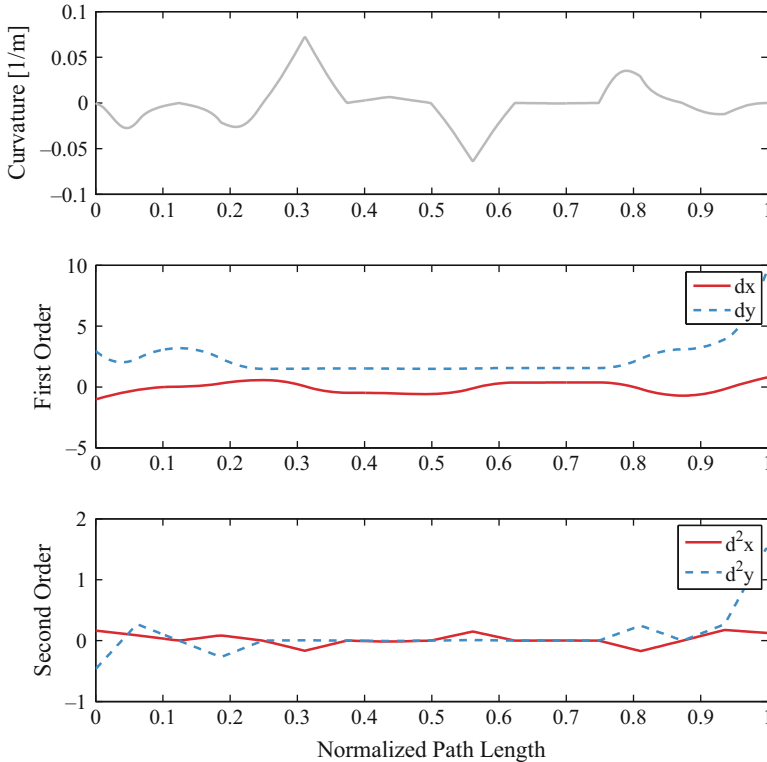


Fig. 9.20 Example 3: Resulting path maintains parametric continuity

Based on the presented numerical and experimental results, we show that this approach:

- Improved accuracy of segment curvature evaluation
- Accelerated segment curvature evaluation
- Decreased path length compared to reference spline and linear path
- Decreased deviation from reference path
- Bounded curvature to desired value while maintaining parametric continuity

The proposed method results in paths that lie within the convex hull of the linear path, with no undesirable oscillations in the path. This produced realistic commands with continuous velocity and acceleration.

This approach relies on smoothing a path defined by successive waypoints, which are generated by a planning algorithm. As presented here, smoothing is considered as a post-planning procedure. Consequently, obtaining an obstacle-free smooth path cannot be guaranteed. In many cases, when collision is detected, replanning is required (Koyuncu and Inalhan 2008). Several researchers examine collision detection for parametric curves (Kwangjin and Sukkarieh 2010; Pan et al. 2012). Circumventing the need for replanning can be achieved by incorporating the

smoothing process within the planning framework. The benefits of guiding the search by the reachability of the robot have been revealed (Shkolnik et al. 2009; Jaillet et al. 2011). In this context, the reachable set can be computed using the efficient curvature evaluation method presented here. Several comments have to be made regarding the presented contributions.

The benefit of solving C^2 parametric continuity problem with maximum curvature constraint and employing the results to mimic human steering is the possibility to combine this parameterization, within any planning framework, such as an RRT or A* algorithm for autonomous vehicles. We predict improvements in human comfort as a result of mimicking human steering, in addition to other claims made by researchers, with regard to continuous curvature paths.

The proposed midpoint insertion algorithm is used to simplify the smoothing algorithm; a more generalized approach would include the location of the inserted point, as a function of the segment angle, but the benefits of doing that are not clear. That could be the subject of other investigation.

In the practical implementation through experiments, we demonstrated that the closely following control polygon method is advantageous over the other algorithms. This benefit comes from the assumption that the path planning algorithm generates a collision-free piecewise linear path, which is then used by our smoothing algorithm, as shown in (Elbanhawi and Simic 2014b). The results can be developed within the context of a recently developed sampling-based algorithm (Elbanhawi and Simic 2014c), which employs efficient collision-checking procedures.

We expect that the outcomes of presented research can be integrated within an efficient planning framework, in which the spline-parameterized motions feel natural to passengers and improve their comfort. Passenger comfort and natural paths are obviously subjective terms that require a large sample of human volunteers for validation. The promising simulations' results, presented here, will be followed by field tests using prototype ground vehicles and UAVs. We plan to validate the concept of graceful motions and curvature continuity, with regard to passenger comfort, by conducting full-scale field experiments.

Acknowledgments M. Elbanhawi acknowledges the financial support of the Australian Post-graduate Award (APA) and Research Training Scheme (RTS).

References

- Alshaer, B. J., Darabseh, T. T., & Alhanouti, M. A. (2013). Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle. *Applied Mathematical Modelling*, 37(7), 5315–5325. doi:http://dx.doi.org/10.1016/j.apm.2012.10.042.
- Anderson, E. P., Beard, R. W., & McLain, T. W. (2005). Real-time dynamic trajectory smoothing for unmanned air vehicles. *IEEE Transactions on Control Systems Technology*, 13(3), 471–477. <https://doi.org/10.1109/TCST.2004.839555>.

- Asano, T., Guibas, L., Hershberger, J., & Imai, H. (1985). Visibility-polygon search and euclidean shortest paths. In: *26th Annual Symposium on Foundations of Computer Science*, 21–23 Oct 1985, pp 155–164. doi:<https://doi.org/10.1109/SFCS.1985.65>.
- Balkcom, D. J., & Mason, M. T. (2002). Time optimal trajectories for bounded velocity differential drive vehicles. *The International Journal of Robotics Research*, 21(3), 199–217. <https://doi.org/10.1177/027836402320556403>.
- Barsky, B. A., & Derosé, T. D. (1990). Geometric continuity of parametric curves: Constructions of geometrically continuous splines. *Computer Graphics and Applications, IEEE*, 10(1), 60–68. <https://doi.org/10.1109/38.45811>.
- Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M., & Soderkvist, I. (2010). Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles. *IEEE Transactions on Automation Science and Engineering*, 7(1), 167–172. <https://doi.org/10.1109/TASE.2009.2015886>.
- Biswas, S., & Lovell, B. C. (2008). *Bézier and splines in image processing and machine vision*. London: Springer.
- Brezak, M., & Petrovic, I. (2013). Real-time approximation of clothoids with bounded error for path planning applications. *Robotics, IEEE Transactions on PP*, 99, 1–9. <https://doi.org/10.1109/TRO.2013.2283928>.
- Brooks, R. A., & Lozano-Perez, T. (1985). A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man and Cybernetics, SMC-15*(2), 224–233. <https://doi.org/10.1109/TSMC.1985.6313352>.
- Campion, G., Bastin, G., & Dandrea-Novet, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1), 47–62. <https://doi.org/10.1109/70.481750>.
- Canny, J. (1985). Voronoi method for the piano-movers problem. In: *1985 I.E. International Conference on Robotics and Automation*, Mar 1985, pp. 530–535. doi:<https://doi.org/10.1109/ROBOT.1985.1087297>.
- Cheng, M. Y., Tsai, M. C., & Kuo, J. C. (2002). Real-time NURBS command generators for CNC servo controllers. *International Journal of Machine Tools and Manufacture*, 42(7), 801–813. doi:[http://dx.doi.org/10.1016/S0890-6955\(02\)00015-9](http://dx.doi.org/10.1016/S0890-6955(02)00015-9).
- De Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6(1), 50–62.
- Donges, E. (1978). A Two-Level Model of Driver Steering Behavior. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 20(6), 691–707. <https://doi.org/10.1177/001872087802000607>.
- Donghyun, K., Cheongjae, J., & Frank, C. P. (2014). Kinematic feedback control laws for generating natural arm movements. *Bioinspiration & Biomimetics*, 9(1), 016002.
- Dyllong, E., & Visioli, A. (2003). Planning and real-time modifications of a trajectory using spline techniques. *Robotica*, 21(05), 475–482. <https://doi.org/10.1017/S02635747030005009>.
- Elbanhawi, M., & Simic, M. (2012). Robotics application in remote data acquisition and control for solar ponds. *Applied Mechanics and Materials*, 253–255, 705–715. <https://doi.org/10.4028/http://www.scientific.net/AMM.253-255.705>.
- Elbanhawi, M., & Simic, M. (2014a). Examining the use of B-splines in Parking Assist Systems. *Applied Mechanics and Materials*, 490–491(1), 1025–1029. <https://doi.org/10.4028/http://www.scientific.net/AMM.490-491.1025>.
- Elbanhawi, M., & Simic, M. (2014b). Randomised kinodynamic motion planning for an autonomous vehicle in semi-structured agricultural areas. *Biosystems Engineering*, 126(0), 30–44. doi:<http://dx.doi.org/10.1016/j.biosystemseng.2014.07.010>.
- Elbanhawi, M., & Simic, M. (2014c). Sampling-based robot motion planning: A review. *Access, IEEE*, 2, 56–77. <https://doi.org/10.1109/ACCESS.2014.2302442>.
- Elbanhawi, M., Simic, M., & Jazar, R. (2014). Continuous path smoothing for car-like robots using B-spline curves. *Journal of Intelligent and Robotic Systems*. (in press). <https://doi.org/10.1007/s10846-014-0172-0>.

- Farin, G. (1992). From conics to NURBS: A tutorial and survey. *Computer Graphics and Applications, IEEE*, 12(5), 78–86. <https://doi.org/10.1109/38.156017>.
- Farin, G. (2002). *Curves and surfaces for CAGD. Computing*. San Francisco: Morgan Kaufmann.
- Fraichard, T., & Scheuer, A. (2004). From Reeds and Shepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, 20(6), 1025–1035. <https://doi.org/10.1109/TRO.2004.833789>.
- Ghilardelli, F., Lini, G., & Piazzzi, A. (2014). Path generation using η 4-Splines for a truck and trailer vehicle. *IEEE Transactions on Automation Science and Engineering*, 11(1), 187–203. <https://doi.org/10.1109/TASE.2013.2266962>.
- Guarino Lo Bianco, C. (2013). Minimum-Jerk velocity planning for mobile robot applications. *IEEE Transactions on Robotics*, 29(5), 1317–1326. <https://doi.org/10.1109/TRO.2013.2262744>.
- Gulati, S., & Kuipers, B. (2008). High performance control for graceful motion of an intelligent wheelchair. In: *ICRA 2008, IEEE International Conference on Robotics and Automation*, 19–23 May 2008, pp. 3932–3938. doi:<https://doi.org/10.1109/ROBOT.2008.4543815>.
- Gulati, S., Jhurani, C., Kuipers, B., & Longoria, R. (2009). A framework for planning comfortable and customizable motion of an assistive mobile robot. In: *IROS 2009, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10–15 Oct 2009, pp. 4253–4260. doi:<https://doi.org/10.1109/IROS.2009.5354172>.
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the Heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107. <https://doi.org/10.1109/TSSC.1968.300136>.
- Hughes, T. J. R., Reali, A., & Sangalli, G. (2008). Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197(49–50), 4104–4124. doi:<http://dx.doi.org/10.1016/j.cma.2008.04.006>.
- Huh, U.-Y., & Chang, S.-R. (2014). A G^2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation. *International Journal of Advanced Robotic Systems*, 11(25). <https://doi.org/10.5772/57340>.
- Jaillet, L., Hoffman, J., van den Berg, J., Abbeel, P., Porta, J. M., & Goldberg, K. (2011). EG-RRT: Environment-guided random trees for kinodynamic motion planning with uncertainty and obstacles. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 25–30 Sept 2011, pp. 2646–2652. doi:<https://doi.org/10.1109/IROS.2011.6094802>.
- Jazar, R. N. (2008). *Vehicle dynamics: Theory and application*. New York: Springer.
- Johnson, J., & Hauser, K. (2012). Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path. In: *2012 I.E. International Conference on Robotics and Automation (ICRA)*, 14–18 May 2012, pp. 2035–2041. doi:<https://doi.org/10.1109/ICRA.2012.6225233>.
- Jolly, K. G., Sreerama Kumar, R., & Vijayakumar, R. (2009). A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. *Robotics and Autonomous Systems*, 57(1), 23–33. doi:<http://dx.doi.org/10.1016/j.robot.2008.03.009>.
- Jones, J., & Adamatzky, A. (2014). Material approximation of data smoothing and spline curves inspired by slime mould. *Bioinspiration & Biomimetics*, 9(3), 036016. <https://doi.org/10.1088/1748-3182/9/3/036016>.
- Kavraki, L. E., Svestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580. <https://doi.org/10.1109/70.508439>.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1), 90–98. <https://doi.org/10.1177/027836498600500106>.
- Koren, Y., & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In: *1991 I.E. International Conference on Robotics and Automation*, 9–11 Apr 1991, (Vol. 1392, pp. 1398–1404). doi:<https://doi.org/10.1109/ROBOT.1991.131810>.

- Koyuncu, E., & Inalhan, G. (2008). A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments. In: *IROS 2008, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 22–26 Sept 2008, pp. 815–821. doi: <https://doi.org/10.1109/IROS.2008.4651122>.
- Kunz, T., & Stilman, M. (2013). Time-optimal trajectory generation for path following with bounded acceleration and velocity. *Robotics: Science and Systems*, 8, 209. Paper 27, July 9–13, University of Sydney, Sydney, NSW, Australia, 2012.
- Kwangjin, Y. (2013). An efficient Spline-based RRT path planner for non-holonomic robots in cluttered environments. In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 28–31 May 2013, pp. 288–297. doi: <https://doi.org/10.1109/ICUAS.2013.6564701>.
- Kwangjin, Y., & Sukkarieh, S. (2010). An analytical continuous-curvature path-smoothing algorithm. *Robotics, IEEE Transactions on*, 26(3), 561–568. <https://doi.org/10.1109/TRO.2010.2042990>.
- Kwangjin, Y., Jung, D., & Sukkarieh, S. (2013). Continuous curvature path-smoothing algorithm using cubic Bezier spiral curves for non-holonomic robots. *Advanced Robotics*, 27(4), 247–258. <https://doi.org/10.1080/01691864.2013.755246>.
- Land, M., & Horwood, J. (1995). Which parts of the road guide steering? *Nature*, 377(6547), 339–340.
- Land, M. F., & Lee, D. N. (1994). Where we look when we steer. *Nature*, 369(6483), 742–744.
- Lapierre, L., Zapata, R., & Lepinay, P. (2007). Combined path-following and obstacle avoidance control of a wheeled robot. *The International Journal of Robotics Research*, 26(4), 361–375. <https://doi.org/10.1177/0278364907076790>.
- Lau, B., Sprunk, C., & Burgard, W. (2009). Kinodynamic motion planning for mobile robots using splines. In: *IROS 2009, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10–15 Oct 2009, pp. 2427–2433. doi: <https://doi.org/10.1109/IROS.2009.5354805>.
- LaValle, S. M., & Kuffner, J. J. (2000). *Rapidly-exploring random trees: A new tool for path planning*. In: Proceedings Workshop on the Algorithmic Foundations of Robotics, Iowa State University.
- Lentink, D. (2014). Bioinspired flight control. *Bioinspiration & Biomimetics*, 9(2), 020301. <https://doi.org/10.1088/1748-3182/9/2/020301>.
- Ma, W., & Kruth, J. P. (1998). NURBS curve and surface fitting for reverse engineering. *International Journal of Advanced Manufacturing Technology*, 14(12), 918–927. <https://doi.org/10.1007/BF01179082>.
- MacAdam, C. C. (1981). Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(6), 393–399. <https://doi.org/10.1109/TSMC.1981.4308705>.
- Maekawa, T., Noda, T., Tamura, S., Ozaki, T., & Machida, K. I. (2010). Curvature continuous path generation for autonomous vehicle using B-spline curves. *Computer-Aided Design*, 42(4), 350–359. doi: <http://dx.doi.org/10.1016/j.cad.2009.12.007>.
- Magid, E., Keren, D., Rivlin, E., & Yavneh, I. (2006). Spline-based robot navigation. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 9–15 Oct 2006, pp. 2296–2301. doi: <https://doi.org/10.1109/IROS.2006.282635>.
- McCrae, J., & Singh, K. (2009). Sketching piecewise clothoid curves. *Computers & Graphics*, 33(4), 452–461. doi: <http://dx.doi.org/10.1016/j.cag.2009.05.006>.
- McPhee, J. E., & Aird, P. L. (2013). Controlled traffic for vegetable production: Part 1. Machinery challenges and options in a diversified vegetable industry. *Biosystems Engineering*, 116(2), 144–154. <https://doi.org/10.1016/j.biosystemseng.2013.06.001>.
- Meek, D. S., & Walton, D. J. (2004). An arc spline approximation to a clothoid. *Journal of Computational and Applied Mathematics*, 170(1), 59–77. <https://doi.org/10.1016/j.cam.2003.12.038>.
- Montes, N., Herraiz, A., Armesto, L., & Tornero, J. (2008). Real-time clothoid approximation by Rational Bezier curves. In: *ICRA 2008, IEEE International Conference on Robotics and*

- Automation*, 19–23 May 2008, pp. 2246–2251. doi:<https://doi.org/10.1109/ROBOT.2008.4543548>.
- Morro, A., Sgorbissa, A., & Zaccaria, R. (2011). Path following for unicycle robots with an arbitrary path curvature. *Robotics, IEEE Transactions on*, 27(5), 1016–1023. <https://doi.org/10.1109/TRO.2011.2148250>.
- Myung, H., Kuffner, J., & Kanade, T. (2007). Efficient two-phase 3D motion planning for small fixed-wing UAVs. In: *2007 I.E. International Conference on Robotics and Automation*, 10–14 April 2007, pp. 1035–1041. doi:<https://doi.org/10.1109/ROBOT.2007.363121>.
- Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., & Kostaras, A. N. (2003). Evolutionary algorithm based offline/online path planner for UAV navigation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 33(6), 898–912. <https://doi.org/10.1109/TSMCB.2002.804370>.
- Ogata K (2010) *Modern Control Engineering*. Prentice Hall,
- Pan, J., Zhang, L., & Manocha, D. (2012). Collision-free and smooth trajectory computation in cluttered environments. *The International Journal of Robotics Research*, 31(10), 1155–1175. <https://doi.org/10.1177/0278364912453186>.
- Piegl, L. A., & Tiller, W. (2001). Parametrization for surface fitting in reverse engineering. *Computer-Aided Design*, 33(8), 593–603. doi:[http://dx.doi.org/10.1016/S0010-4485\(00\)00103-2](http://dx.doi.org/10.1016/S0010-4485(00)00103-2).
- Pivtoraiko, M., & Kelly, A. (2011). Kinodynamic motion planning with state lattice motion primitives. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 25–30 Sept 2011, pp. 2172–2179. doi:<https://doi.org/10.1109/IROS.2011.6094900>.
- Pivtoraiko, M., Knepper, R. A., & Kelly, A. (2009). Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3), 308–333.
- Prokop, G. (2001). Modeling human vehicle driving by model predictive online optimization. *Vehicle System Dynamics*, 35(1), 19–53. <https://doi.org/10.1076/vesd.35.1.19.5614>.
- Reeds, J. A., & Shepp, L. A. (1990). Optimal paths for a car that goes both forward and backward. *Pacific Journal of Mathematics*, 145(2), 367–393.
- Roth, S., & Batavia, P. (2002). *Evaluating path tracker performance for outdoor mobile robots* (pp. 26–27/07). Chicago, Illinois, USA: Paper presented at the Automation Technology for Off-Road Equipment.
- Sabelhaus, D., Röben, F., Meyer zu Helligen, L. P., & Schulze Lammers, P. (2013). Using continuous-curvature paths to generate feasible headland turn manoeuvres. *Biosystems Engineering*, 116(4), 399–409. doi: <http://dx.doi.org/10.1016/j.biosystemseng.2013.08.012>.
- Sachin, P., Jia, P., Abbeel, P., & Goldberg, K. (2014). Planning curvature and torsion constrained ribbons in 3D with application to intracavitary brachytherapy. Paper presented at the *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Boğaziçi University, İstanbul, Turkey, 3–5 August 2014.
- Schoenberg, I. J. (1946). Contributions to the problem of approximation of equidistant data by analytic functions – A. On the problem of smoothing or graduations a 1st class of approximation formulae. *Quarterly of Applied Mathematics*, 4(1), 45–99.
- Sgorbissa, A., & Zaccaria, R. (2010). 3D path following with no bounds on the path curvature through surface intersection. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 18–22 Oct 2010. pp. 4029–4035. doi:<https://doi.org/10.1109/IROS.2010.5653235>.
- Shkolnik, A., Walter, M., & Tedrake, R. (2009). Reachability-guided sampling for planning under differential constraints. In: *ICRA '09, IEEE International Conference on, Robotics and Automation*, 12–17 May 2009, pp 2859–2865. doi:<https://doi.org/10.1109/ROBOT.2009.5152874>.
- Sungchul, J., & Taehoon, K. (2003). Tool-path generation for NURBS surface machining. In: *American Control Conference*, 2003. Proceedings of the 2003, 4–6 June 2003 (Vol. 2613, pp. 2614–2619). doi:<https://doi.org/10.1109/ACC.2003.1243471>.

- Turner, M., & Griffin, M. J. (1999). Motion sickness in public road transport: The effect of driver, route and vehicle. *Ergonomics*, 42(12), 1646–1664. <https://doi.org/10.1080/001401399184730>.
- Unser, M., Aldroubi, A., & Eden, M. (1993). B-spline signal processing. I. Theory. *Signal Processing, IEEE Transactions on*, 41(2), 821–833. <https://doi.org/10.1109/78.193220>.
- Velenis, E., & Tsiotras, P. (2008). Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding-horizon implementation. *Journal of Optimization Theory and Applications*, 138(2), 275–296. <https://doi.org/10.1007/s10957-008-9381-7>.
- Walton, D. J., & Meek, D. S. (2005). A controlled clothoid spline. *Computers & Graphics*, 29(3), 353–363. doi:<http://dx.doi.org/10.1016/j.cag.2005.03.008>.
- Walton, D. J., Meek, D. S., & Ali, J. M. (2003). Planar G2 transition curves composed of cubic Bézier spiral segments. *Journal of Computational and Applied Mathematics*, 157(2), 453–476. [https://doi.org/10.1016/s0377-0427\(03\)00435-7](https://doi.org/10.1016/s0377-0427(03)00435-7).
- Wang, L. Z., Miura, K. T., Nakamae, E., Yamamoto, T., & Wang, T. J. (2001). An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves. *Computer-Aided Design*, 33(14), 1049–1058. doi:[http://dx.doi.org/10.1016/S0010-4485\(00\)00142-1](http://dx.doi.org/10.1016/S0010-4485(00)00142-1).
- Wu, W., Chen, H., & Woo, P.-Y. (2000). Time optimal path planning for a wheeled mobile robot. *Journal of Robotic Systems*, 17(11), 585–591. [https://doi.org/10.1002/1097-4563\(200011\)17:11<585::AID-ROB1>3.0.CO;2-7](https://doi.org/10.1002/1097-4563(200011)17:11<585::AID-ROB1>3.0.CO;2-7).
- Xuan-Nam, B., Boissonnat, J. D., Soueres, P., & Laumond, J. P. (1994). Shortest path synthesis for Dubins non-holonomic robot. In: *IEEE International Conference on Robotics and automation*, 8–13 May 1994, (Vol.1, pp. 2–7). doi:<https://doi.org/10.1109/ROBOT.1994.351019>.
- Yang, K., Moon, S., Yoo, S., Kang, J., Doh, N., Kim, H., & Joo, S. (2014). Spline-Based RRT Path Planner for Non-Holonomic Robots. *Journal of Intelligent and Robotic Systems*, 73(1–4), 763–782. <https://doi.org/10.1007/s10846-013-9963-y>.
- Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C. L., & Hughes, T. J. R. (2007). Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29–30), 2943–2959. doi:<http://dx.doi.org/10.1016/j.cma.2007.02.009>.
- Zucker, M., Ratliff, N., Dragan, A. D., Pivtoraiko, M., Klingensmith, M., Dellin, C. M., Bagnell, J. A., & Srinivasa, S. S. (2013). CHOMP: Covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9–10), 1164–1193. <https://doi.org/10.1177/0278364913488805>.