



Review

Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges

Abhijeet Ravankar ^{1,*}, Ankit A. Ravankar ^{2,†}, Yukinori Kobayashi ², Yohei Hoshino ¹ and Chao-Chung Peng ³

¹ School of Regional Innovation and Social Design Engineering, Faculty of Engineering, Kitami Institute of Technology, Kitami, Hokkaido 090-8507, Japan; hoshinoy@mail.kitami-it.ac.jp

² Division of Human Mechanical Systems and Design, Faculty of Engineering, Hokkaido University, Sapporo, Hokkaido 060-8628, Japan; ankit@eng.hokudai.ac.jp (A.A.R.); kobay@eng.hokudai.ac.jp (Y.K.)

³ Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan 701, Taiwan; ccpeng@mail.ncku.edu.tw

* Correspondence: aravankar@mail.kitami-it.ac.jp

† These authors contributed equally to this work.

Received: 8 August 2018; Accepted: 17 September 2018; Published: 19 September 2018



Abstract: Robot navigation is an indispensable component of any mobile service robot. Many path planning algorithms generate a path which has many sharp or angular turns. Such paths are not fit for mobile robot as it has to slow down at these sharp turns. These robots could be carrying delicate, dangerous, or precious items and executing these sharp turns may not be feasible kinematically. On the contrary, smooth trajectories are often desired for robot motion and must be generated while considering the static and dynamic obstacles and other constraints like feasible curvature, robot and lane dimensions, and speed. The aim of this paper is to succinctly summarize and review the path smoothing techniques in robot navigation and discuss the challenges and future trends. Both autonomous mobile robots and autonomous vehicles (outdoor robots or self-driving cars) are discussed. The state-of-the-art algorithms are broadly classified into different categories and each approach is introduced briefly with necessary background, merits, and drawbacks. Finally, the paper discusses the current and future challenges in optimal trajectory generation and smoothing research.

Keywords: robot trajectory smoothing; robot navigation; path planning; autonomous vehicle motion planning

1. Introduction

The advent of ubiquitous sensors [1], artificial intelligence, and decrease in the cost of computing [2] has set the stage for an increase in the number of service mobile robots [3]. Today, these robot are used for cleaning, industry automation, and moving stuff in the warehouse [4]. Service robots are also perfect for tasks which are dull, dirty, dangerous, and difficult. Most of these robots have the ability to navigate autonomously in an environment. For large environments and work on a large scale, multiple mobile robots are often used. These robots autonomously move between various locations of the environment to provide their services (like deliver items, or clean).

In order to navigate between different parts of the environment, mobile robots need a map of the environment. They also need to localize themselves in the map to estimate their current position. To do this, mobile service robots are generally equipped with exteroceptive sensors like monocular and stereo cameras, laser range finders, inertial measurement units (IMUs), and RGBD (RGB color information with depth) sensors. A SLAM (Simultaneous Localization and Mapping) [5,6] module builds the map of the environment and simultaneously localizes the robot in the map. The localized

position of the robot is the start location, and the goal location is specified for the robot for it to navigate towards the goal using any of the state-of-the-art path planning algorithms. SLAM is considered to be a core module for autonomous mobile robots [7] because it is often a prerequisite to path planning, navigation, and manipulation for single and multi-robot systems [8–11].

In order to navigate towards its goal in a given map, a robot first requires to find the overall path from the start to the goal location. This involves considering the static obstacles of the map. The dynamic obstacles in the map (like moving people, other robots) are not considered at this stage. Various algorithms for path planning of autonomous robots have been proposed for planning this overall path. Each algorithm has its own advantages and disadvantages and a review of these algorithms can be found in [12–14]. For example, Dijkstra’s algorithm [15] with uniform cost search has widely been used for global planning. However, it is computationally expensive with a time complexity of $\mathcal{O}(N^2)$, where N is the number of nodes. Optimization of the Dijkstra’s algorithm has been proposed in [16] and variants like [17] have been proposed. Using a good initial heuristic, A-star algorithm [18] performs better than Dijkstra’s algorithm. On the other hand, D-star (or Dynamic A*) algorithm [19,20] finds an optimal path in real-time by incrementally updating paths to the robot’s state as new information is discovered, and is more efficient than A* algorithm. Rapidly exploring random tree (RRT) algorithm [21–23] efficiently searches high dimensional spaces by building space-filling trees randomly. It plans a path with a biased growth towards the goal with high probability. Other widely used path planning algorithms include potential fields [24] algorithm, and probabilistic road map (PRM) [25] planner. During the traversal of this overall path a robot might encounter dynamic obstacles like moving people or other robots and must change its trajectory accordingly. The dynamic obstacles are considered using various approaches like the “dynamic window approach” proposed in [26], and a smooth path is generated from the global path.

Most of the path planning algorithms generate a path consisting of straight lines and sharp turns.

As an example, Figure 1 shows a robot and its goal location. The green path is a path consisting of straight lines and sharp turns at points A, B, C, D, and E. Such path is not desired for robot motion as the robot cannot make these sharp turns suddenly, and needs to slow down. In the case of a robotic wheelchair operated by an injured person or patient, such sudden stops and sharp turns are not favorable and may even cause injuries or induce unpleasant feelings in the person. Moreover, from an external person’s point of view in the same environment such erratic robot motion is unnatural, and it is difficult to predict the robot’s next position to avoid collision. In some cases, depending on the kinematics of the robot, it might even be difficult to execute some of these sharp turns. Such paths are also not suitable for robots delivering delicate, precious, or dangerous items.

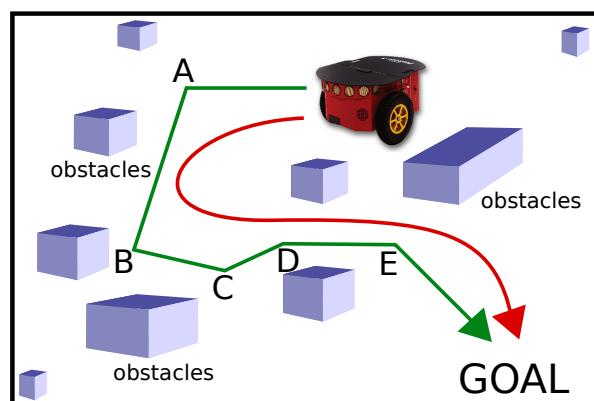


Figure 1. The green path is a path consisting of straight lines and sharp turns at points A, B, C, D, and E. A smooth and continuous path is shown in red color.

A smooth and continuous path is desirable for robot navigation as shown in red color in Figure 1. Such path avoids abrupt and sharp turns, and the robot can maneuver without stopping. Path

smoothing is an important problem in service robots and the smooth paths must satisfy certain constraints like continuity and safety. The continuity problem mainly refers to the geometric continuity in terms of tangential or curvature continuity. The safety check ensures that the smooth path is sufficiently far from the obstacles. Robot motion kinematics is another factor which must be considered while smoothing the paths.

In this paper, we review the state-of-the-art approaches used for smoothing the paths of the mobile robots. We not only summarize the various algorithms, but also provide a concise mathematical description about the particular techniques. We discuss the novelty, advantages, and limitations of each method. At last we discuss some of the main challenges in path smoothing.

2. A Short Note on Path Continuity

The smoothness of a path is generally expressed in terms of continuity [27]. Continuity is of two types: (a) Geometric continuity (G^i), and (b) Parametric continuity (C^i). Geometric continuity ensures that the endpoints of the various segments of the path meet, and the tangent vector's directions are equal. Parametric continuity ensure that the endpoints of the various path segments meet, and tangent vector's direction and magnitudes both are equal. Roughly speaking, parametric continuity (C^i) implies geometric continuity (G^i), but not vice-versa.

Two curves are C^i continuous at a point p if the i th derivatives of the two curves are also equal at the point p . If segments of the two curves are C^i continuous at point p , then they are also C^k continuous $\forall k \leq i$. In general, a curve s has C^n continuity if its n th derivative $\frac{d^n s}{dt^n}$ is also continuous. Figure 2a shows a discontinuous path. A C^0 continuous path is shown in Figure 2b and it connects all the points between the start and the goal location such that there is no discontinuity. However, there is a slope discontinuity at the point joining the two path segments. A C^1 continuous path is shown in Figure 2c is also C^0 continuous and also preserves the tangency at the point of joint. In other words, the path matches first differential values at each point in the path. However, the curvature of the two segments are not continuous. It is evident that in Figure 2c that the straight segment has no curvature, while the curved segment has a finite curvature. Hence, the curvature suddenly bumps from zero to a finite value at the point joining the two segments shown in Figure 2d. A C^2 continuous path is shown in Figure 2d and preserves the second order differential values at each point in the path. Therefore, C^1 continuous path is smoother than C^0 continuous at the point joining the two curves. Similarly, C^2 continuous is smoother than C^1 continuous curve at the joint.

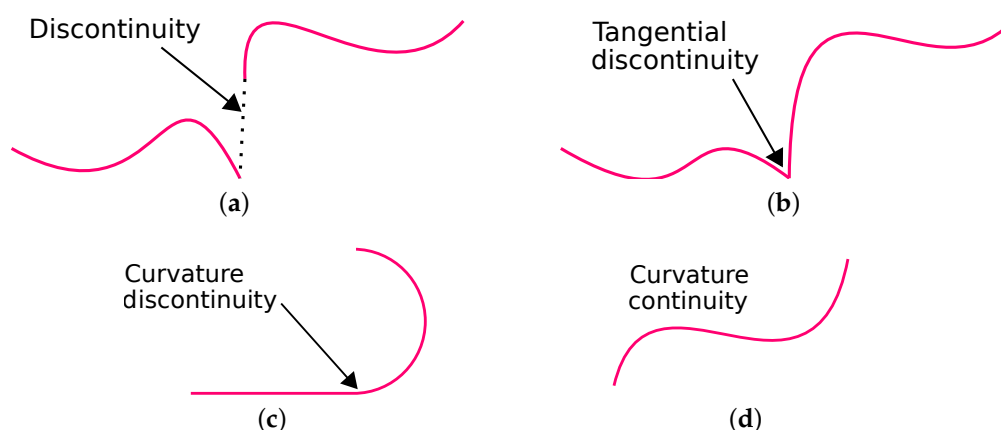


Figure 2. Parametric continuity. (a) Discontinuous curve segments. (b) C^0 continuity. (c) C^1 continuity. (d) C^2 continuity.

In terms of robot path planning and motion, the terms parametric continuity (C^i) and geometric continuity (G^i) are both found in the literature. As discussed in [28–30], parametric continuity means

smoothness both of the curve and of its parameterization. Geometric continuity simply means the smoothness of the track that the robot traverses. For example, C^1 continuity means continuity of the tangent vector, while G^1 continuity means continuity of slope; C^2 continuity means continuity of the acceleration vector, while G^2 continuity means continuity of the curvature. In terms of robot motion, the C^1 continuous motion preserves velocity, whereas C^2 continuous path preserves acceleration. For robot path planning, what matters is mainly the path's C^1 or C^2 continuity. Higher orders of continuity like C^3 continuity or higher deals with surface continuity and find usage mainly in CAD/CAM based design applications.

3. Interpolation Based Path Smoothing

In literature, interpolation technique was first proposed by E. Warning [31,32]. Precisely, given $m + 1$ pairs (x_i, y_i) , the problem consists of finding a function $\psi = \psi(x)$ such that $\psi(x_i) = y_i$ for $i = 0, \dots, m$, y_i being some given values, and say that ψ interpolates $\{y_i\}$ at the nodes $\{x_i\}$. We speak about polynomial interpolation if ψ is an algebraic polynomial, trigonometric approximation if ψ is a trigonometric polynomial, or piecewise polynomial interpolation (or spline interpolation) if ψ is only locally a polynomial.

3.1. Polynomial Interpolation

In literature, the two major polynomial interpolation techniques found for path smoothing use Lagrange's interpolating polynomial and Hermite's interpolating polynomial which are discussed below.

The Lagrange interpolating polynomial [33] is the polynomial $P(x)$ of degree $\leq (n - 1)$ that passes through the n points $(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), \dots, (x_n, y_n = f(x_n))$, and is given by,

$$P(x) = \sum_{j=1}^n P_j(x), \quad (1)$$

where,

$$P_j(x) = y_j \prod_{k=1, k \neq j}^n \frac{x - x_k}{x_j - x_k}. \quad (2)$$

Another interpolation technique is the Hermite's interpolating polynomial. Let $p(x)$ be an n th degree polynomial with zeros at x_1, \dots, x_n . Then, the fundamental Hermite interpolating polynomials [34,35] of the first and second kinds are defined by,

$$h_i^{(1)}(x) = \left[1 - \frac{p''(x_i)}{p'(x_i)}(x - x_i) \right] [p_i(x)]^2 \quad (3)$$

and

$$h_i^{(2)}(x) = (x - x_i) [p_i(x)]^2, \quad (4)$$

for $i = 1, 2, \dots, n$, where the fundamental polynomials of Lagrange interpolation are defined by,

$$p_i(x) = \frac{p(x)}{p'(x_i)(x - x_i)}. \quad (5)$$

Path smoothing using interpolation is very old class of algorithms [36]. Two major limitations of interpolation techniques are: (a) high computational costs, and (b) Runge's phenomenon [37] which is a classic illustration of polynomial interpolation non-convergence.

In a recent work ([38,39]) proposed by S.R. Chang and U.Y. Huh, a QPMI (Quadratic Polynomial and Membership Interpolation) algorithm was proposed, which avoids Runge's phenomenon [37] and the weakness of spline interpolation, by creating a G^2 continuous path using just the quadratic

polynomials and membership functions. In [39], S.R. Chang et al. propose a collision free continuous G^2 path using interpolation. However, their methods required explicit collision detection checks, and reprogramming of smooth paths which can be expensive in case of crowded environments. Other approaches involve using quintic polynomials [40]. A summary of work related to interpolation can be found in [29].

3.2. Bézier Curve

These parametric curves make use of ‘control points’ to define the shape as shown in Figure 3. At their core, they make use of Bernstein polynomial functions [41]. Given a set of $n + 1$ control points P_0, P_1, \dots, P_n , the corresponding Bézier curve (or Bernstein-Bézier curve) [42,43] is given by,

$$C(t) = \sum_{i=0}^n P_i B_{i,n}(t), \quad (6)$$

where, $B_{i,n}(t)$ is a Bernstein polynomial [41] and $t \in [0, 1]$. A Bernstein polynomial of degree n is defined by,

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad (7)$$

where,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \quad (8)$$

Work in [42] proposes two path planning algorithms which makes use of Bezier curves generated using proper positioning of the control points for a mobile robot to navigate in a corridor with constraints. In [44], a smooth path generation for automated vehicle using Bezier curve is proposed for both 2D and 3D case. Work in [42,45] discusses the optimized placement of control points. An interesting work in [46] uses Bezier curves as transition curves (described in Section 6) to join a straight segment with a curved segment, and also in generating clothoidal paths [47,48] demonstrating their modularity and extensibility. Trajectory generation for vehicles in urban environments has been discussed in [49–51]. Some researchers have used Bezier curves for automatic parking of vehicles [52].

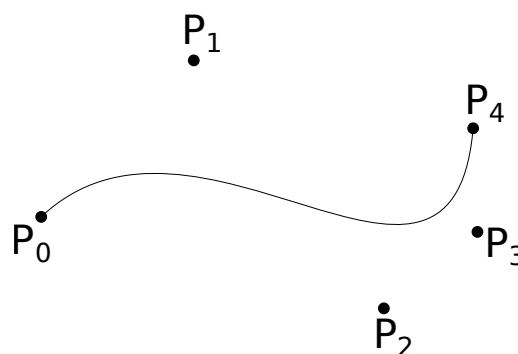


Figure 3. Bezier curve through control points P_0, P_1, \dots, P_4 .

3.3. Cubic Splines

A spline [53,54] is a piecewise polynomial function that can have a locally very simple form, yet at the same time be globally flexible and smooth. Splines are very useful for modeling arbitrary functions, and are used extensively in computer graphics. In interpolating problems, spline interpolation is often preferred to polynomial interpolation because it yields similar results, even when using low degree polynomials, while avoiding Runge’s phenomenon for higher degrees.

The two important characteristics of interpolatory cubic splines [35] are: (1) they are the splines of minimum degree that yield C^2 approximations; and (2) they are sufficiently smooth in

the presence of small curvatures. Let us thus consider, in $[a, b]$, $n + 1$ ordered nodes $a = x_0 < x_1 < \dots < x_n = b$ and the corresponding evaluations $f_i, i = 0, \dots, n$. The aim is to provide an efficient procedure for constructing the cubic spline interpolating those values. Since the spline is of degree 3, its second-order derivative must be continuous. By introducing the following notation $f_i = s_3(x_i)$, $m_i = s'_3(x_i)$, $M_i = s''_3(x_i)$, $i = 0, \dots, n$, the cubic spline interpolation is given by,

$$s_{3,i-1}(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + C_{i-1}(x - x_{i-1}) + \tilde{C}_{i-1} \quad (9)$$

where, $h_i = x_i - x_{i-1}, i = 0, \dots, n$ and,

$$\tilde{C}_{i-1} = f_{i-1} - M_{i-1} \frac{h_i^2}{6}, C_{i-1} = \frac{f_i - f_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}), \quad (10)$$

M_i can be calculated from the M-continuity system:

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i, i = 0, \dots, n. \quad (11)$$

3.4. B-Spline

A B-spline is a generalization of the Bézier curve. Given m real values $x_i, x_0 \leq x_1 \leq \dots \leq x_{m-1}$, called knots, B-spline parametric curve of degree n ,

$$S : [x_0, x_{m-1}] \rightarrow \mathbb{R}^2$$

is composed of a linear combination of basis B-splines $b_{i,n}$ of degree n ,

$$S(x) : \sum_{i=0}^{m-n-2} P_i B_{i,n}(x), x \in [x_n, x_{m-n-1}], \quad (12)$$

where, P_i are called control points, and the $m - n - 1$ control points form a convex hull. The $m - n - 1$ basis B-spline of degree n is defined [35] as below with $j = 0, \dots, m - 2$.

$$B_{j,n}(x) = \frac{x - x_j}{x_{j+n} - x_j} B_{j,n-1}(x) + \frac{x_{j+n+1} - x}{x_{j+n+1} - x_{j+1}} B_{j+1,n-1}(x). \quad (13)$$

Note that, when the number of control points is one more than the degree and $x \in [0, 1]$, i.e., $x_0 = \dots = x_n = 0$ and $x_{n+1} = \dots = x_{2n} = 1$, the B-spline degenerates into Bezier curve [35]. Figure 4 shows B-Spline curve generated in red color along with the basis splines.

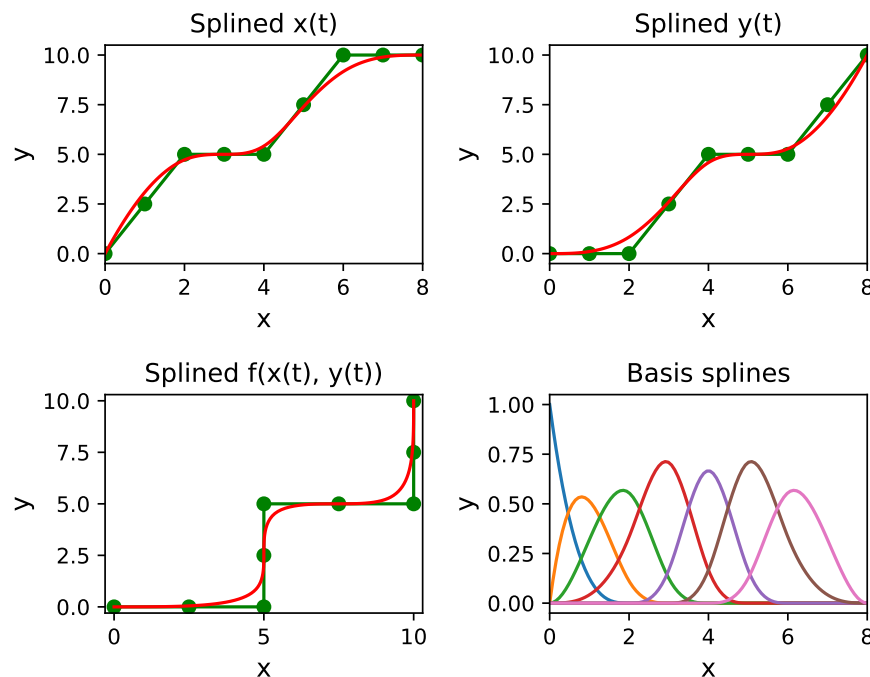


Figure 4. An example of path smoothing using B-Spline. The green path is the global path while the red path is the smooth path. The green dots are the control points.

Komoriya et al. discusses trajectory design and control of a mobile robot using B-spline curves [55]. B-splines were chosen for path planning algorithm in ‘KAT-5’ vehicle [56,57] to complete the 2005 DARPA challenge [58] primarily because of the ease in which the shape of their resulting curves can be controlled. Spline interpolation has been used in the autonomous driving car ‘Stanley’ which won the DARPA challenge [59] in 2006, and ‘Odin’ [60] which secured third place in the same challenge in 2007. Berglund et al. [61] has used B-Splines for planning smooth and obstacle free paths for autonomous mining vehicles (more than 100 tonnes) operating at speeds of around 20 km/h. For vision based autonomous vehicles, a Quintic G^2 -spline-based steering algorithm is proposed in [62]. B-Splines have also been used [63] for curvature continuous trajectory generation with energy minimization as the goal. In order to recover from the problems of complicated curvature functions which require many parameters improved methods using spline-based smoothing has been proposed in [64] allowing better visual representation and better data compression compared to traditional methods. In [65], a motion planner tailored for particular requirements for robotic car navigation is proposed which leverages B-spline curve properties to include vehicle’s constraint requirements, thus lowering the search dimensionality.

B-Splines are powerful tools for path smoothing as desired trajectory can be generated for different degrees and obstacle configuration. For example, Figure 5 shows various trajectories based on different number of control points and curve degrees. Without smoothing, the robot will encounter sharp turns at the corners. Figure 5a shows B-Spline-based trajectories with nine control points and curve degrees from 1 to 8, in a closed form (i.e., the start and end points are the same). In this configuration, the generated curves are too far from the actual points. However, by doubling the number of control points by introducing extra control points between two control points, the turns are smoothed out and the smoothed trajectories are shown in Figure 5c for various degrees. Similarly, Figure 5e shows the smooth trajectory with 36 control points. In this case, it can be seen that the smoothed trajectory keeps the straight segments straight and only smooths the sharp turns for varying degrees of the curve. Figure 5b,d,f shows the open loop versions of their respective figures on left column of Figure 5.

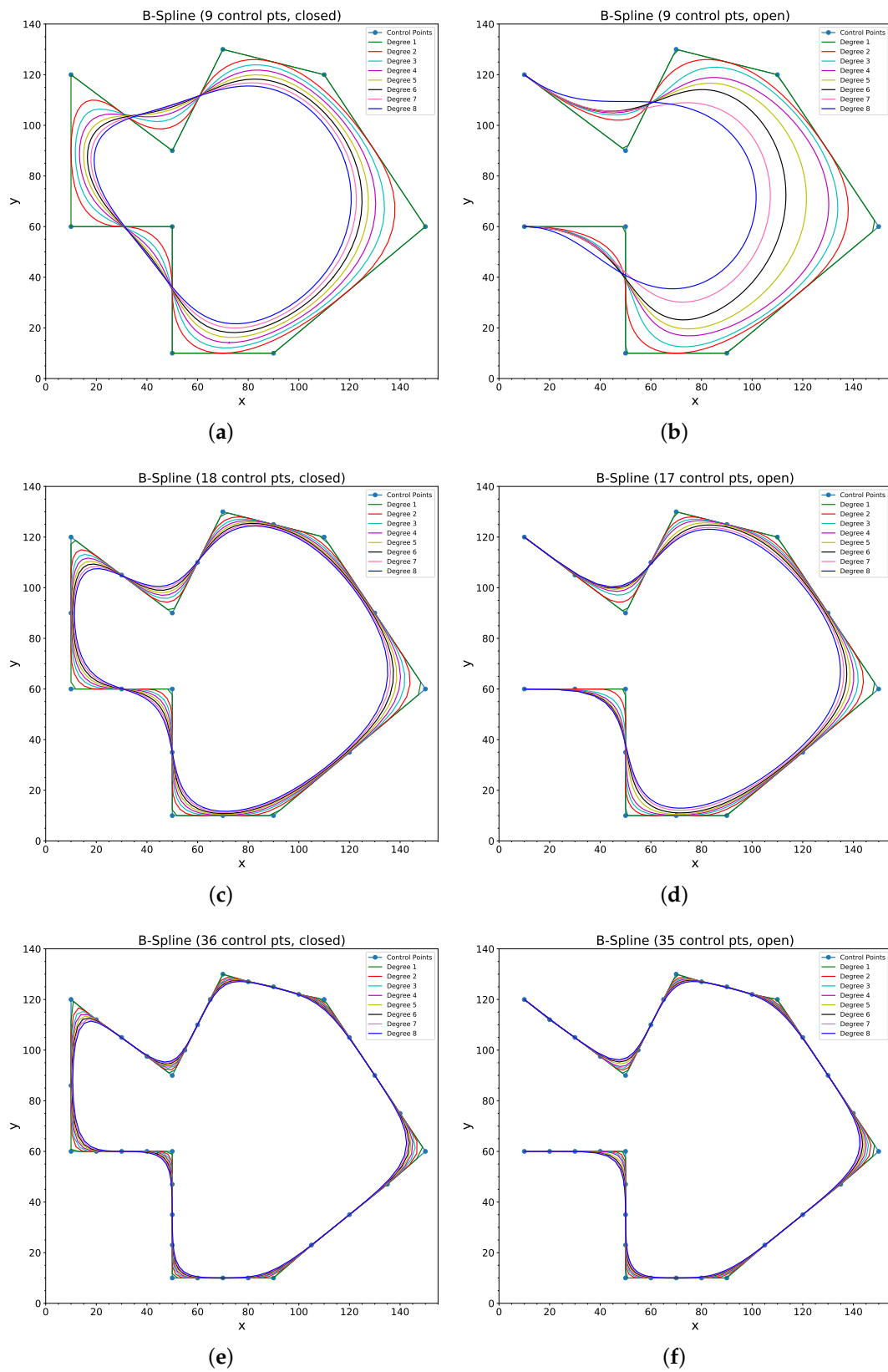


Figure 5. B-Spline-based robot path smoothing with varying number of control points and degrees. (a,c,e) shows closed path (robot returns to the same location) while (b,d,f) shows open paths. (a) Closed path, 9 control points. (b) Open path, 9 control points. (c) Closed path, 18 control points. (d) Open path, 17 control points. (e) Closed path, 36 control points. (f) Open path, 35 control points.

3.5. NURBS Curve

A non-uniform rational B-spline (NURBS) curve [66–68] is defined by,

$$C(t) = \frac{\sum_{i=0}^n N_{i,p}(t)w_i P_i}{\sum_{i=0}^n N_{i,p}(t)w_i} \quad (14)$$

where p is the order, $N_{i,p}$ are the B-spline basis functions, P_i are control points, and the weight w_i of P_i is the last ordinate of the homogeneous point P_i^w .

Thus, by manipulating both the control points and the weights, NURBS prove to be a very flexible tool to generate desired trajectories. Moreover, NURBS are invariant under shear, translation, rotation, scaling, as well as parallel and perspective projection [69]. NURBS based path smoothing has been applied in [70] for trajectory generation of an industrial five axis needle winding robot with optimal winding pattern in the stator slots. Apart from trajectory planning, some researchers have used NURBS curves for 3D map smoothing [71]. Work in [72] proposes an arm trajectory planner with obstacle avoidance. Similarly, Quintic NURBS has also been used for the optimal trajectory planning of manipulators in [73]. In [74–76], NURBS based trajectory generation for an autonomous mobile robot navigating in 3D environment has been proposed with obstacle avoidance. NURBS has also been used in swarm robotics [77], path planning of humanoid robots [78], and even segmenting unknown objects in RGB-D images for robotics tasks such as object search, grasping and manipulation [79].

The main drawbacks of using NURBS is that they require extra storage, and improper initialization of weights can lead to bad parametrization [69]. However, NURBS are still powerful with clear geometric interpretations, fast evaluation, and stable computation [80].

4. Path Smoothing Using Special Curves

The other segment of robot path smoothing algorithms utilize one or more curves from the family of curves which includes: parabola, ellipse, hyperbola, cardioid, limaçon, hypocycloid, cycloid, pedal curves, and spirals. A detailed mathematical description of these curves is given in [81]. Here, we discuss some of the most prominent curves found in the state-of-the-art.

4.1. Dubin's Curve

Given two points and in a plane and a specified direction of motion, Dubins [82] in 1957 used circular arcs and straight line segments to find the shortest smooth path of bounded curvature that joins the points [13,83]. Figure 6 shows an example of path smoothing using Dubin's curves. Segments shown in red color i.e., \overline{AB} , \overline{CD} , and \overline{EF} are the straight segments which are combined with circular arcs BC and DE shown in green color.

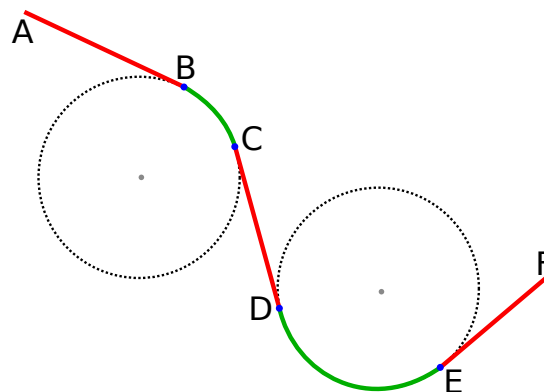


Figure 6. Path smoothing using Dubin's curve. Straight segments \overline{AB} , \overline{CD} , and \overline{EF} in red color are combined with circular arcs BC and DE shown in green color.

Since then, a number of modifications and improvements have been proposed. In [84], authors have considered using Dubin's curve with obstacle avoidance. Unmanned Aerial Vehicle (UAVs) navigation using predictive vector field control is proposed in [85] which uses Dubin curves. Similarly, work in [86] proposes a path planning algorithm based on 3D Dubins curves for UAVs to avoid both static and moving obstacles using a variation of Rapidly-exploring Random Tree (RRT) [21] as the planner. In [87], authors have proposed an efficient two-phase approach to motion planning for small fixed-wing UAVs navigating in complex 3D environments. Path smoothing for UAV with similar goals is proposed in [88]. First, a kinematically feasible obstacle-free coarse global path is computed in a discretized 3D environment. Then, the optimal trajectory based on a set of admissible paths based on the task tree approach are generated using Dubin's curve. Work in [89] proposes an algorithm for planning C^∞ paths with bound curvature and curvature derivative linking two fixed (initial and final) configurations and passing through a given number of intermediate via-points. In [90] authors have used Dubins curves extended to 3D case that model the properties of turn and straight flight for wind-aware emergency landing. In agricultural robotics, Dubin's curves have been used for coverage path planning of autonomous robotic lawn mower equipped with GPS in [91].

Dubin's curves provide a simple yet powerful technique for real-time path smoothing as they are generally not computationally expensive. They can be combined with clothoids (discussed in Section 4.2) to satisfy different constraints. For example, Chen et.al. proposed to use Dubin's curve in conjunction with Fermat's spiral [92] to design a curvature continuous path in [93]. Like clothoids, Fermat's spiral curvature changes continuously with length.

4.2. Clothoid

Clothoid [94], also known as Euler's spiral or Cornu's spiral is the curve parameterized in the complex plane of points by,

$$B(t) = S(t) + iC(t), \quad (15)$$

where,

$$\begin{aligned} x(t) = C(t) &= \int_0^t \cos\left(\frac{\pi}{2}s^2\right) ds \\ y(t) = S(t) &= \int_0^t \sin\left(\frac{\pi}{2}s^2\right) ds \end{aligned} \quad (16)$$

where C and S are the Fresnel functions [95], sometimes called the Fresnel cosine integral and Fresnel sine integral.

Both Fresnel functions approach $\frac{1}{2}$ as $t \rightarrow \infty$ and so the curve slowly spirals toward $(\frac{1}{2}, \frac{1}{2})$ in the first quadrant. And by symmetry, because both functions are odd, the curve spirals toward $(-\frac{1}{2}, -\frac{1}{2})$ in the third quadrant. Cornu's curve has the property that its curvature is proportional to the distance along the path of the curve. Hence, a vehicle traveling at constant speed will experience a constant rate of angular acceleration as it travels around the curve—this means that the driver can turn the steering wheel at a constant rate which makes the junction safer.

These properties make clothoids an attractive option for path smoothing of trajectories. In [96], path smoothing for car-like vehicle navigation has been proposed in which a continuous curvature path is generated by multiple clothoids composition and parametric adjustment. Clothoid based path smoothing has been presented in [97–101]. Recently, clothoids have been used in trajectory generation of autonomous Audi TTS car [102], and vehicles in VisLab Intercontinental Autonomous Challenge [103]. Some researchers have used clothoids for autonomous parallel parking with geometric continuous-curvature path planning in [104–106].

Brezak et al. proposed [107] a method computation of clothoid coordinates that guarantees bounded approximation error over a wide range of clothoid parameters in real-time. Clothoids

have also been used in trajectory planning of vehicles at relatively large speeds than mobile robots. For instance, work in [108] proposes using dynamically feasible clothoid trajectories which provide a reliable method for representing the vehicle's path for the next few seconds of driving. In [109] authors have considered a case of trajectory generation of vehicle consisting of a robotic walking assistant pushed by a user. In [110] motion planning algorithm is proposed for a mobile robot that reduces not only the path length, but also the curvature change along the path using clothoids. In [111] authors have considered the bounded-curvature path of the of the three-wheeled omni-directional mobile robot based on a smooth road which is described as a clothoid. Connecting the straight segments by symmetric clothoid curves at the junction has been proposed in [112]. An overtaking maneuver of mobile robot using clothoids is undertaken in [113].

Work in [114] has proposed to integrate clothoid based trajectory into the Robot Operating System (ROS) [115] framework. It should be noted that ROS is a very popular framework for developing robot applications. Integration of such path smoothing algorithms in the ROS framework should facilitate mobile robot navigation based applications.

4.3. Hypocycloid

A hypocycloid is a geometrical curve which is produced by a fixed point P which lies on the circumference of a small circle of radius r_s rolling inside a larger circle of radius $R_L > r_s$ [81,116]. The rolling circle has a radius of r_s and the large circle has a radius of R_L , and the curve is defined in general by,

$$\begin{aligned} x(\theta) &= (R_L - r_s) \cos(\theta) + r_s \cdot \cos\left(\frac{R_L - r_s}{r_s} \theta\right) \\ y(\theta) &= (R_L - r_s) \sin(\theta) - r_s \cdot \sin\left(\frac{R_L - r_s}{r_s} \theta\right) \end{aligned} \quad (17)$$

where, η is the number of cusps and ξ is the ratio of the radius of the rolling circle to the radius of the large circle, i.e., $\xi = \frac{r_s}{R_L}$. A cusp is defined as the sharp corner where the curve is not differentiable. Figure 7b shows a 3-cusped hypocycloid called as a deltoid or tricuspoid. A 4-cusped hypocycloid called as an astroid is shown in Figure 7a.

In general, to obtain n cusps in a hypocycloid, the radius of smaller circle is set to $r_s = \frac{R_L}{n}$, as n rotations of the smaller circle brings it back to the original position, generating n cusps while traversal [81]. For a deltoid, $R_L = 3 \times r_s$, $\xi = \frac{1}{3}$ and for astroid, $R_L = 4 \times r_s$, and $\xi = \frac{1}{4}$.

Recently, a new approach of Smooth Hypocycloidal Paths (SHP) has been proposed in [117]. An important characteristic of hypocycloids is that they can be generated for any angle as shown in Figure 8. Figure 8 shows hypocycloid curve generation for various angles. This property is used in [117] for path smoothing with an aim to keep the straight segments straight and only smooth the points of sharp turns. In [118], an extension of SHP has been proposed in which SHP has been proposed as an 'extension' or a 'plug-in'. Any of the traditional path planning algorithms can be used for overall planning and the sharp turns are detected and smoothed using hypocycloidal curves. Similarly, other curves like circle involute (Figure 7c), cycloid (Figure 7d), logarithmic spiral (Figure 7e), and clothoids (Figure 7f) can also be used for smoothing and a detailed mathematical description can be found in [81].

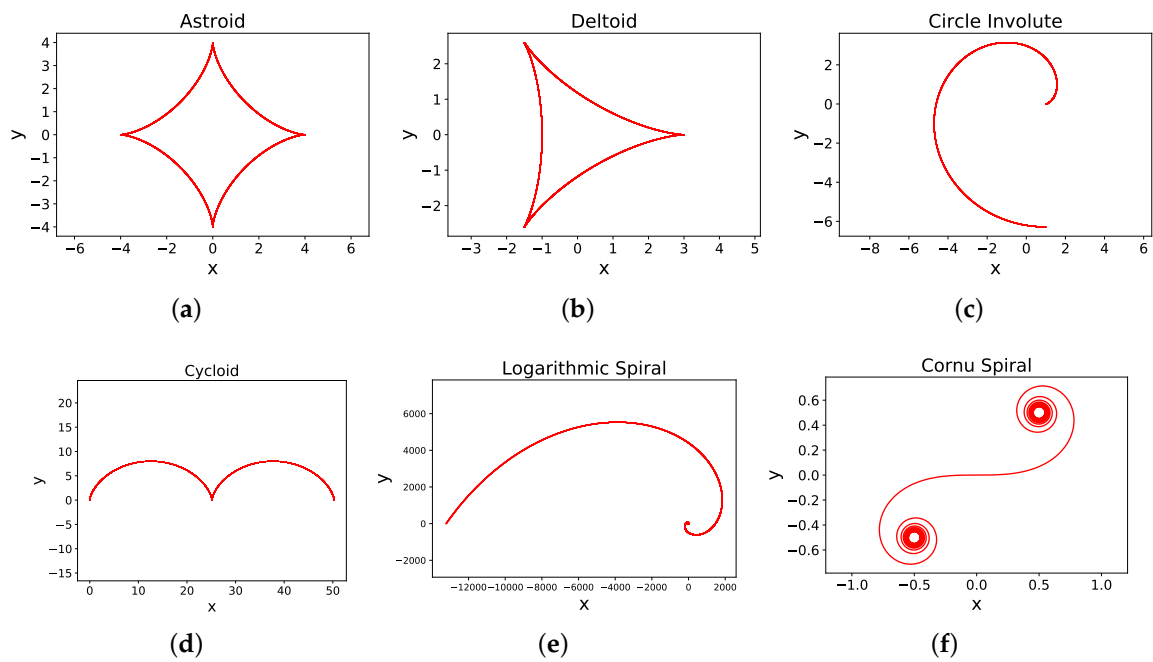


Figure 7. Curves for path smoothing. (a) Astroid. (b) Deltoid. (c) Circle Involute. (d) Cycloid. (e) Logarithmic Spiral. (f) Clothoid or Cornu Spiral (Euler's Spiral).

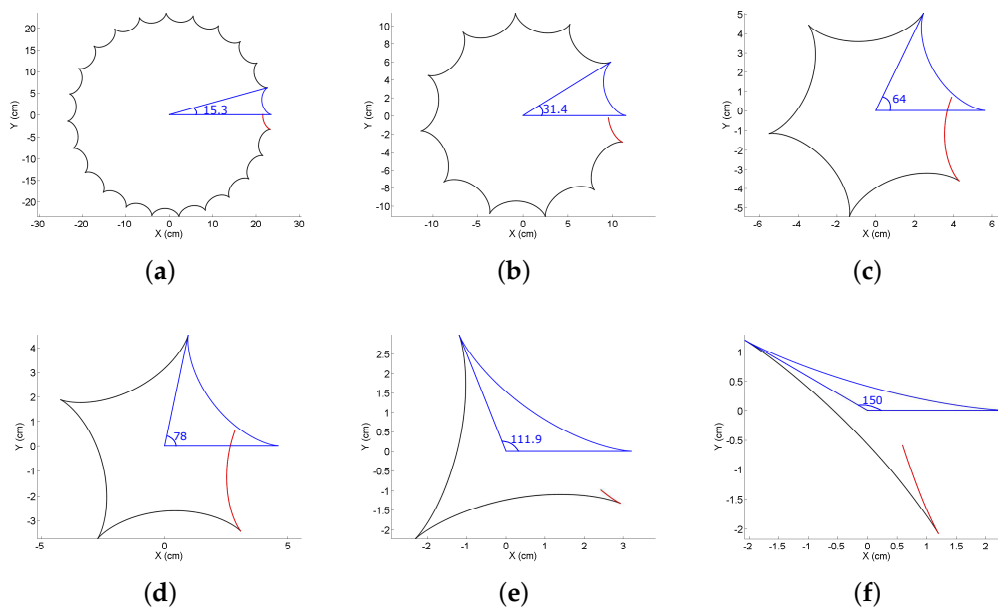


Figure 8. SHP [117] generation with different angles. The first segment of the curve in blue is taken to generate SHP, whereas other curve segments in black and red are ignored. (a) $\theta = 15.3^\circ$. (b) $\theta = 31.4^\circ$. (c) $\theta = 64^\circ$. (d) $\theta = 78^\circ$. (e) $\theta = 111.9^\circ$. (f) $\theta = 150^\circ$.

5. Optimization-Based Path Smoothing

Path smoothing has also been considered as a function optimization problem guiding a mobile robot on the lane and avoiding obstacles through minimizing several parameter constraints like speed, rollover constraints, acceleration, jerk, and others [119–122].

Work in [123] proposes a convex elastic smoothing heuristic algorithm for trajectory smoothing and speed optimization for mobile robots with car-like dynamics. The key feature of the algorithm is that the optimization problem can be solved via convex programming, making it fast. In [124], authors have proposed an algorithm for a non-holonomic wheeled vehicle operating in a semi-structured environment. First, the algorithm computes offline a finite set of feasible motions connecting discrete robot states to construct a search graph. Later, the motion primitives based on Bezier curves are generated by solving the constrained optimization problem.

In this regard “Time Elastic Band” planner (TEB Planner) it is worth mentioning [125–129] as it is available with ROS integration [130,131]. Timed Elastic Band locally optimizes the robot’s trajectory with respect to trajectory execution time, separation from obstacles and compliance with kinodynamic constraints at runtime. The TEB planner optimizes robot trajectories by subsequent modification of an initial trajectory generated by a global planner. The objectives considered in the trajectory optimization include the overall path length, trajectory execution time, separation from obstacles, passing through intermediate way points, compliance with the robots dynamic, kinematic and geometric constraints, and dynamic obstacles. It also allows efficient online motion planning of car-like robots.

Some of these optimization based approaches have been used practically in the DARPA Grand Challenge. Work in [132,133] proposes an optimization based navigation algorithm for the DARPA Grand Challenge. The authors address the path planning problem with a nonlinear optimization method running in real-time. An optimization problem is continually solved to find a time-optimal, dynamically feasible trajectory from the vehicle’s position to some receding horizon ahead (20 m–70 m forward).

Similarly, in [134] authors describe a practical path-planning algorithm for an autonomous vehicle operating in an unknown semi-structured (or unstructured) environment, where obstacles are detected online by the robot’s sensors. The first phase uses a variant of A* search to obtain a kinematically feasible trajectory. The second phase then improves the quality of the solution via numeric non-linear optimization, leading to a local (and frequently global) optimum. Work in [135,136] presents an algorithm for trajectory planning used on-board the vehicle that completed the 103 km of the Bertha-Benz-Memorial-Route fully autonomously. In the algorithm, the constraints are carefully designed to ensure that the solution converges to a single, global optimum. In [137] authors propose a motion planner for autonomous vehicles based on the idea of a on-road state lattice. A focused search is performed in the previously identified region in which the optimal trajectory is most likely to exist.

A comparison of various trajectory smoothing methods in terms of advantages and disadvantages is given in Table 1.

Table 1. A comparison of various trajectory smoothing techniques.

Classification	Main Advantages (+) and Disadvantages (–)
Dubins Curve	<ul style="list-style-type: none"> + Fast to compute for given configuration of obstacles. + Dubin’s curves are easy to compute even on low spec hardware. + Shortest paths are assured. – These curves do not have curvature continuity. – Robot will experience a jerk at the point of transition of straight line and circle.
Bezier Curve	<ul style="list-style-type: none"> + Bezier curves have low computational cost. + Control points can generate curve of desired characteristic. + Bezier curves can be connected with each other to get desired shape. – With increasing degree of curves, computation costs increase. – Difficult to adjust for curves with higher degrees. – Global waypoints affect the entire curve – It might be difficult to place control points.

Table 1. Cont.

Classification	Main Advantages (+) and Disadvantages (–)
Splines	<ul style="list-style-type: none"> + Splines have low computational cost. + They can easily provide C^2 continuity desired for robots. + Knots can easily control the shape of splines.
	<ul style="list-style-type: none"> – It might be difficult to balance the trade-off between continuity and desired shape.
NURBS	<ul style="list-style-type: none"> + NURBS are easy to compute, with fast and stable computation. + They can be very flexible to generate desired trajectories. + They are invariant under shear, translation, rotation, or scaling. + They are powerful tools used in CAD/CAM applications.
	<ul style="list-style-type: none"> – NURBS require more memory storage. – Improper initialization of weights can lead to bad parametrization.
Clothoids	<ul style="list-style-type: none"> + Clothoids curvature changes linearly. + Curvature continuity is easy to obtain. + Clothoids can be used as transition curves in conjunction to other curves. + Heavily used in railway track and highway road designs.
	<ul style="list-style-type: none"> – Fresnel’s integral might be difficult to compute. – Clothoid based planning uses global waypoints.
Interpolation Methods	<ul style="list-style-type: none"> + Generally easy to compute. + Curves can be concatenated to get desired shape. + Fit for local planning for safety.
	<ul style="list-style-type: none"> – Difficult to control coefficients of curves of higher order (> 4). – Curves of higher order are time consuming and not suitable for high speeds.
Hypocycloids	<ul style="list-style-type: none"> + Easy to compute. + Can be generated for desired angles.
	<ul style="list-style-type: none"> – C^2 continuity is not guaranteed. – Requires using transition curves (clothoids) for curvature continuity. – Not suitable for robots at high speeds.
Optimization Methods	<ul style="list-style-type: none"> + Various constraints can be taken into account while optimizing. + Can be combined with other approaches.
	<ul style="list-style-type: none"> – Depends on global pathways. – Optimization is time consuming and might not necessarily converge.

6. A Note on Transition Curves

There are many algorithms in the literature (for example Dubin’s curves [82], SHP [117], etc.) which ensure a C^1 continuity in which the tangential constraint is satisfied, however, higher orders of continuity like C^2 curvature continuity is not guaranteed. As explained in [117], although C^1 continuity is enough for mobile robots navigating at low speed, in case of a robot traveling at high speed with acceleration, it is not enough. In these cases, a transition between the straight line and the curved section is required. This is a well studied problem and various solutions are available. In fact, smoothing out a straight section of track to a curved section while maintaining curvature continuity using transition curves [138] has extensively been studied for building railway tracks and highway roads. We briefly discuss transition curves, as their solutions in parametric forms can easily be obtained and computed.

Transition curves [138] connect a straight segment of the path at one end to the curve at the other end. Hence, radius of curvature changes from zero on the straight segment to a finite value of the curve, at a uniform rate. It eliminates the kink generated by directly connecting the straight and the curve section [117]. Transition curves have the following important properties desirable for robot motion: (a) They are tangential to the straight line of the path, i.e., curvature at start is zero. (b) It joins the circular curve tangentially. (c) It’s curvature increases at the same rate. Transition curves have rigorously been studied for C^2 continuity in many works [139–141], and can also provide smooth C^2 transition between two straight lines [117].

Figure 9a shows a curved path in red meeting the straight segment at point A. To generate transition curve, point A is shifted back to A' which is the starting point of the transition curve shown in blue. Although different types of curves can be used to generate transition curves, clothoids (i.e., Euler's curve or cornu spiral shown in Figure 9c) are the most common [138]. Cubical parabola shown in Figure 9b can also be used. Taking into account linear increase of curvature from zero on the straight segment to the curvature of the hypocycloid, rate of change of angle at any point (P) on the transition curve of length L (shown in blue in Figure 9a) is given by,

$$\frac{d\theta}{ds} = \frac{s}{R_L \cdot L'} \quad (18)$$

where s is the length of the transition curve from point A' to P in Figure 9a. Integrating Equation (18), we obtain,

$$\theta = \frac{s^2}{2R_L \cdot L}. \quad (19)$$

During the initial condition (on straight segment) when $s = 0$, θ is also 0. As explained in [117], apart from polar coordinates, the equation of transition curve in parametric form is given as [138,142],

$$\begin{aligned} x &= s - \frac{s^5}{40R_L^2L^2}, \\ y &= \frac{s^3}{6R_L} - \frac{s^7}{336R_L^3L^3}. \end{aligned} \quad (20)$$

Similarly, Figure 9c shows using a clothoid in red color for smooth transition from point A to B to meet the curve in blue color. Authors in [117] have described using transition curves for mobile robot navigation applications.

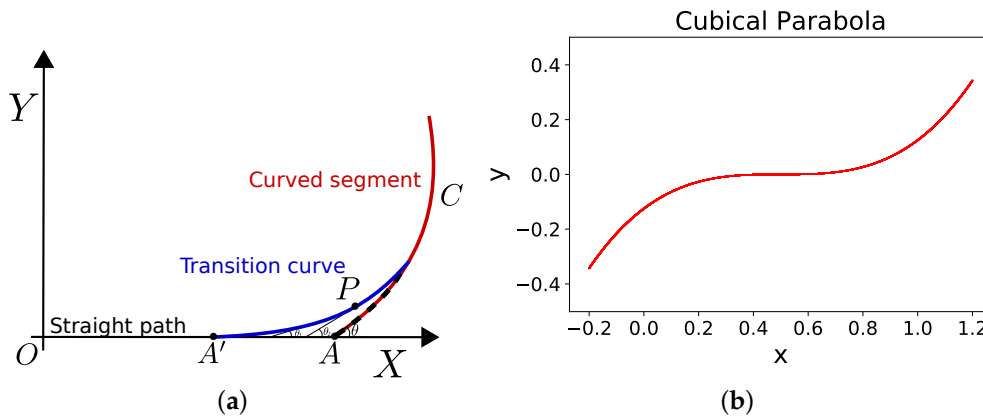


Figure 9. Cont.

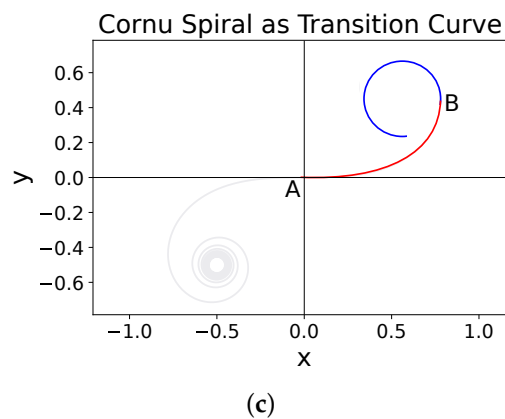


Figure 9. Transition Curve. (a) Point A is shifted back to A' which is the starting point of the transition curve shown in blue. There is a gradual increase in curvature so that there is no sudden kink or jerk. (b) Cubical parabola as a transition curve. (c) Cornu Spiral or clothoid as a transition curve.

7. Robot Trajectories with Obstacle Avoidance

There is a huge plethora of research work in static and dynamic obstacle avoidance of mobile robots. This section limits the review to obstacle avoidance in the context of smooth trajectory generation. Many works in the state-of-the-art consider a static scenario in which the positions of the obstacles are well known and smooth paths are generated. However, in real environments there are moving entities like people and other robots in vicinity and smoothing must be done in real-time to avoid collision while still traversing a smooth path. Even in the case of outdoor robots and autonomous vehicles, there is a constant challenge from multiple entities like pedestrians, cyclists, and other vehicles. Thus, there is a gap between global path planning and real-time sensor-based robot control. This is graphically explained by taking an example in Figure 10 in which the gray rectangle represents an obstacle while the green path is the non-optimized global path. B-spline-based smooth trajectory is generated and shown in red color. In Figure 10a, the smoothed path collides with the obstacle and is not suitable. Hence by introducing additional control points at $x = 30$, $x = 35$, and $x = 40$, the smooth path can be adjusted to avoid collision as shown in Figure 10b,c,d, respectively. This is easy in case of static obstacles. However, dynamic obstacles have to be tracked for speed and orientation, and the control points need to be fixed at correct places to generate the trajectory (for example in case of using B-Splines).

In this regard, one of the significant early works was proposed by S. Quinlan and O. Khatib in “Elastic Bands” planner [143,144]. An elastic band is a deformable collision-free path whose initial shape is the planner’s smooth global path. The elastic band continues to deform as changes in the environment are detected by sensors, enabling the robot to accommodate uncertainties and react to unexpected and moving obstacles. This is done in real time and the smooth paths maintain a safe threshold distance from the obstacles [143]. Later this approach was extended to non-holonomic kinematics [145–147], robotic systems with many degrees of freedom [148], and dynamics obstacles [149]. However, these works do not take any dynamic constraints of the underlying robot into account directly. An improved method built upon [143] called timed elastic band [126,127] considers temporal aspects of the motion in terms of dynamic constraints such as limited robot velocities and accelerations, and the problem is formulated in a weighted multi-objective optimization framework, and is suitable for high dimensional state spaces. The disadvantage of timed elastic band local planner is that it often gets stuck in a locally optimal trajectory as they are unable to transit across obstacles. This limitation was addressed in [125,128], in which, a subset of admissible trajectories of distinctive topologies are optimized in parallel. The local planner is able to switch to the current globally optimal trajectory from the candidate set. The extension to robots with car-like motion model was proposed in [129,130].

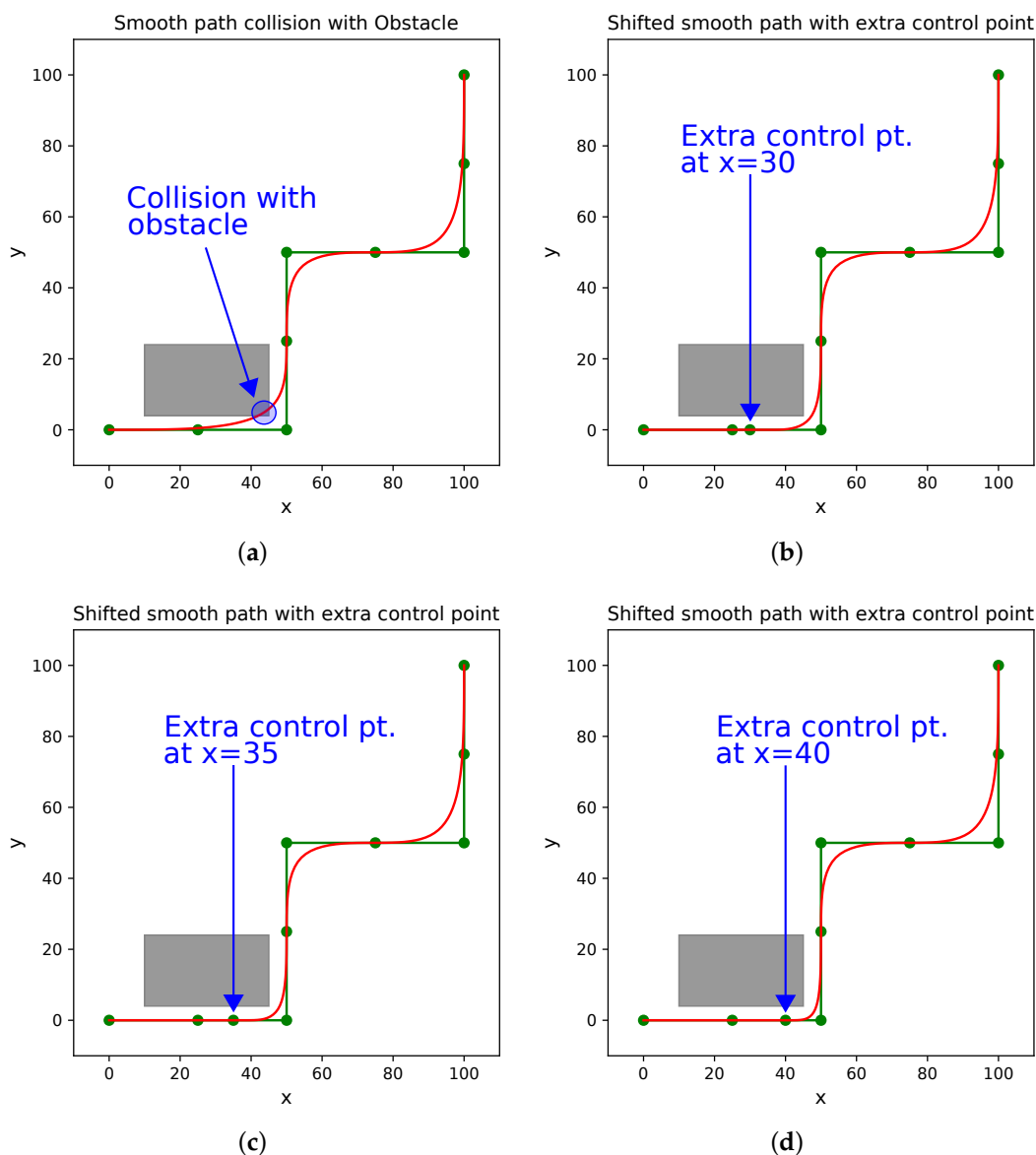


Figure 10. Problem of smooth trajectory generation while avoiding collision with obstacles with the case of B-Spline-based smoothing. (a) Smooth path in red color collides with the obstacle. (b) Inclusion of extra control point at $x = 30$ avoids collision. (c) Smooth path with extra control point at $x = 35$. (d) Smooth path with extra control point at $x = 40$.

The controllability and ease of smooth trajectory generation while avoiding the obstacles is also important. Splines reviewed in Section 3.4 are potential tools in this respect. The *culebra* (Spanish for “snake”) algorithm proposed in [57] for the KAT-5 vehicle [56]. In it, a centerline is first generated from the waypoints. Additional equidistant control points are inserted between the original waypoints and a cubic B-spline is drawn fitting the control points until no obstacles were found along the resulting path, iteratively. In the vehicle Stanley [59], if an obstacle is encountered, the algorithm plans a smooth change in lateral offset that avoids the obstacle and the trajectory can be safely executed. Planning in lateral offset space also has the advantage that it gracefully handles GPS error.

A log-space solution for robotic path planning with harmonic functions is proposed in [150]. The log-space solution rapidly produces smooth obstacle-avoiding trajectories, and supports planning in exponentially larger real-world robotic applications. Similarly, work in [151] proposes a real-time

path planner for a smart wheelchair using harmonic potentials. Voronoi and potential maps have been used for safe trajectory generation in [152]. In [153], collision avoidance considering the shape, kinematics, and dynamics of a mobile robot is presented. A Virtual Force Field (VFF) method was proposed in [154] for real-time obstacle avoidance approach for mobile robots. The virtual force field method integrates certainty grids for obstacle representation and potential fields for navigation, suitable for inaccurate sensor data and sensor fusion. While the VFF method provides superior real-time obstacle avoidance for fast mobile robots, some limitations concerning fast travel among densely cluttered obstacles were addressed in Vector Field Histogram [155] method which was further developed in VFH+ algorithm [156] which provides smoother robot trajectories and greater reliability, VFH* algorithm [157] (and variants [158]) which also provides a verification that the selected trajectory avoids obstacles, and VFH*TDT (VFH* with Time Dependent Tree) algorithm [159].

In [160], a curvature velocity method based on a probabilistic 3D occupancy and velocity grid is proposed for dynamic scenarios. Apart from curvature velocity and potential methods, dynamic window approach based obstacle avoidance [26,161,162], and nearness diagram [163–167] based algorithms have also shown promising results for obstacle avoidance. It is worth mentioning that apart from smooth and safe trajectory generation avoiding obstacles, other factors like path length, accuracy, control, computational cost, and reliability in uncertain scenarios are also important. A review of obstacle avoidance algorithms for mobile robots can be found in [168].

8. Challenges

The research in trajectory smoothing has been successful at various fronts, nevertheless, several challenges still remain and are summarized below:

1. **Trajectory smoothing in dynamic environments:** One of the biggest constraint in these terms is the speed of detection of dynamic entities which is directly affected by the total number of entities tracked. Hence, in an environment like an open public space, there are a lot of dynamic entities for the mobile robot to track which consumes a lot of time. This has an adverse effect on the trajectory smoothing process as the robot ends up stopping suddenly or considerably reducing the speed while trying to select the best possible alternate trajectory among the potential candidates. It is important to accurately track these dynamic entities in conjunction to path smoothing. This is more challenging in autonomous self driving cars at high speeds. Work in [169] has succinctly summarized the challenges in pedestrian detection considering the resolution, range, and field-of-view of various on-board sensors like radar, lidar, or omni-directional cameras with various types of hardware. In addition, work in [170] has focused the survey of challenges of pedestrian detection with vision based sensors which are very prominent recently. In adverse conditions like low illumination, night, snowfall, and rain, it is further difficult to detect the dynamic entities while work has been done in this regard by using thermal images [171]. In both indoor and outdoor environments, occlusion is another big hindrance with dynamic entity detection and solutions have been proposed [172,173], although it still is an open problem.
2. **Fusing trajectory smoothing into SLAM process:** SLAM is an indispensable module for any mobile robot. Even autonomous vehicles need to localize themselves in the environment and build or update their map using perceptive sensors like GPS, cameras, or range sensors. SLAM models the uncertainty of the robot motion and the sensor errors to come up with the most optimum state estimation [5,174]. Trajectory generation is directly linked with currently estimated state and perception and hence fusing it in the SLAM module is beneficial. Currently, it seems like SLAM module is decoupled from the trajectory smoothing module. However, the two modules must work side-by-side to update the map with the new obstacles or entities and generate real-time smooth trajectories on the fly. Work in [175] presents a motion planning algorithm considering both the uncertainty caused by robot and dynamic entities. The motion of dynamic entities are predicted using a local planner, and the uncertainty along the predicted trajectory is computed based on Gaussian propagation. In this regard, a relative continuous-time SLAM

has been proposed by Anderson et al. [176] which uses weights on cubic B-splines to represent continuous state variables. Only the local weights are adjusted during optimization, while implicit trajectory prior is arbitrary. In [177], a Simultaneous Trajectory Estimation and Mapping (STEAM) is proposed which uses Gaussian Process (GP) regression instead of cubic B-splines. It interpolates between conventional state parameterizations at certain key times. When applied in SE3, this parameterization can represent realistic probabilistic trajectories obeying nonlinear, nonholonomic motion models. Although slow for dense kernels, a careful selection can result in realistic sparse GP kernels that are very fast. A non-uniform sampling of the trajectory representation over the sliding window with continuous correction is presented in [178]. In [179], a dense map-centric SLAM method based on a continuous-time trajectory is proposed which removes the need for global trajectory optimization by introducing map deformation. Some other recent significant works in continuous-time SLAM are [180–183], while a broad overview of challenges in SLAM can be found in [184].

3. **Operator in the loop, Safety, and User Experience:** In case of tele-operated robots, currently, the trajectory generation and control lacks the input from the operator. This is more important in case of autonomous vehicles to feedback the planned trajectory to the driver, and generate smooth trajectories based on driver's intentions. This requires active feedback mechanisms and integration of human-robot integration [185–187]. Researchers have proposed work in this regard in [188,189] by proposing a trajectory planning algorithm that 'adapts' to traffic on a lane-structured infrastructure such as highways. In many researches, the emphasis has been on the mathematical completeness of the system while the user-experience seems to get ignored. For instance, in case of an autonomous robot wheelchair used in hospitals, safety is important and the wheelchair must not come close to either of the walls. Hence, in this particular scenario, it is important that the robot wheelchair moves in nearly a straight line in the center lane of the passage. However, aiming for C^2 or higher continuities, many algorithms generate a path which brings the robot close to either of the walls. For example, Figure 11a shows the results of paths generated by D^* [19], PRM [25], QPMI [39], and SHP [117] in dotted green, red, black, and blue colors, respectively (results reproduced from [117]). The curve in black is C^2 continuous, however, it brings the robot too close to the obstacles at points Q_2 and Q_3 . On the other hand, SHP [117] curve in blue keeps the robot sufficiently far from the obstacles and keeps straight segments of the path straight considering the input from operators of robot wheelchair. Such safe paths are easy to generate particularly on grid-maps by using Voronoi paths [190], or using thinning algorithms [191,192] or skeleton maps [193,194]. Figure 11b shows the skeleton path of the environment shown in Figure 11a. In conjunction to the previous point, the feedback from the operator must be fused, and operator intentions must be anticipated to generate smooth paths. With the advent of autonomous cars and platforms like fully autonomous robotic wheelchair, the concept of user experience especially in terms of *passenger comfort* is being re-evaluated. This is a relatively new area of research with a strong correlation with smooth trajectory generation, and some promising research has been proposed in [195–200].

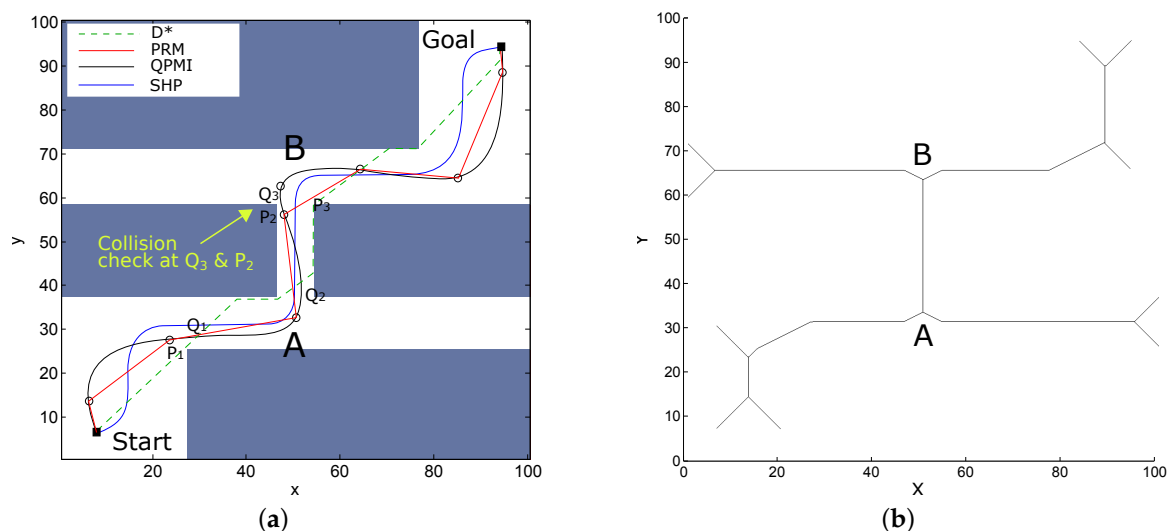


Figure 11. Safety and user experience while smoothing paths. (a) Paths generated by D* [19], PRM [25], QPMI [39], and SHP [117] in dotted green, red, black, and blue colors, respectively. (b) The skeleton path of the environment of Figure 11a.

9. Discussion and Conclusions

In this paper, we summarized the various path smoothing techniques and algorithms present in the state-of-the-art. With the advent of autonomous robots, and self driving cars, optimal trajectory generation and smoothing becomes an important field of research. We presented the actual algorithm with concise mathematical description and then discussed the work presented by various researchers along with their novel points. The paper discussed the advantages and disadvantages of the various approaches. Since safe traversal of smooth trajectory is important, we reviewed important obstacle avoidance methodologies in mobile robotics. Finally, since there is an ongoing research in this important field, the paper discussed some of the important challenges in the field in terms of trajectory smoothing in dynamic environments, fusing trajectory smoothing in SLAM process, the importance of operator in the control loop, and safety and use experience. The review covered topics in continuous-time SLAM and simultaneous trajectory estimation and mapping with collision detection. This review did not cover the actual integration of path smoothing algorithms with the planning algorithms like A* or PRM planner. Actually executing the smooth trajectory by a robot depends on many factors like the motion model (ex. differential drive or car-like model) of the robot, its kinematic constraints and others. Moreover, the actual execution of obstacle detection and avoidance while executing the smooth trajectories also depends on the sensors attached to the robot. The smooth paths generated by various methods in the state-of-the-art are generally evaluated in terms of the continuity parameters which were briefly covered in this review.

It is anticipated that we will see an influx of more and more autonomous mobile robots and vehicles working in human-centric environments. In such scenarios, research in optimal trajectory generation and smoothing is expected to see solutions for the current problems and advance further. In that context, the present work is expected to provide readers with a thorough overview of the state-of-the-art and challenges to actively pursue research and propose novel solutions.

Author Contributions: A.R. and A.A.R. conceived the idea, designed, performed experiments, and summarized the research; Y.K. made valuable suggestions to analyze the data and improve the manuscript. Y.H. and C.-C.P. provided important feedback to improve the manuscript. The manuscript was written by A.R.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ITU. Ubiquitous Sensor Networks (USN). 2018. Available online: http://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000040001PDFE.pdf (accessed on 1 June 2018).
2. Cost of Computing. Wikipedia. 2016. Available online: <https://en.wikipedia.org/wiki/FLOPS> (accessed on 30 December 2016).
3. Ortigoza, R.S.; Marcelino-Aranda, M.; Ortigoza, G.S.; Guzman, V.M.H.; Molina-Vilchis, M.A.; Saldana-Gonzalez, G.; Herrera-Lozada, J.C.; Olguin-Carbajal, M. Wheeled Mobile Robots: A review. *IEEE Lat. Am. Trans.* **2012**, *10*, 2209–2217. [\[CrossRef\]](#)
4. Correll, N.; Bekris, K.E.; Berenson, D.; Brock, O.; Causo, A.; Hauser, K.; Okada, K.; Rodriguez, A.; Romano, J.M.; Wurman, P.R. Analysis and Observations from the First Amazon Picking Challenge. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 172–188. [\[CrossRef\]](#)
5. Ravankar, A.A.; Hoshino, Y.; Ravankar, A.; Jixin, L.; Emaru, T.; Kobayashi, Y. Algorithms and a Framework for Indoor Robot Mapping in a Noisy Environment using Clustering in Spatial and Hough Domains. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 27. [\[CrossRef\]](#)
6. Ravankar, A.; Ravankar, A.A.; Hoshino, Y.; Emaru, T.; Kobayashi, Y. On a Hopping-points SVD and Hough Transform Based Line Detection Algorithm for Robot Localization and Mapping. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 98. [\[CrossRef\]](#)
7. Thrun, S.; Burgard, W.; Fox, D.; Hexmoor, H.; Mataric, M. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Mach. Learn.* **1998**, *31*, 29–53. [\[CrossRef\]](#)
8. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Emaru, T. Avoiding blind leading the blind. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1729881416666088. [\[CrossRef\]](#)
9. Ravankar, A.; Ravankar, A.; Kobayashi, Y.; Emaru, T. Hitchhiking Robots: A Collaborative Approach for Efficient Multi-Robot Navigation in Indoor Environments. *Sensors* **2017**, *17*, 1878. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Ravankar, A.; Ravankar, A.; Kobayashi, Y.; Hoshino, Y.; Peng, C.C.; Watanabe, M. Hitchhiking Based Symbiotic Multi-Robot Navigation in Sensor Networks. *Robotics* **2018**, *7*, 37. [\[CrossRef\]](#)
11. Ravankar, A.; Ravankar, A.; Kobayashi, Y.; Emaru, T. Symbiotic Navigation in Multi-Robot Systems with Remote Obstacle Knowledge Sharing. *Sensors* **2017**, *17*, 1581. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Delling, D.; Sanders, P.; Schultes, D.; Wagner, D. Engineering Route Planning Algorithms. In *Algorithmics of Large and Complex Networks*; Lerner, J., Wagner, D., Zweig, K., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5515, pp. 117–139.
13. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006. Available online: <http://planning.cs.uiuc.edu/> (accessed on 11 February 2016).
14. Latombe, J.C. *Robot Motion Planning*; Kluwer Academic Publishers: Norwell, MA, USA, 1991.
15. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [\[CrossRef\]](#)
16. Shu-Xi, W. The Improved Dijkstra's Shortest Path Algorithm and Its Application. 2012 International Workshop on Information and Electronics Engineering. *Procedia Eng.* **2012**, *29*, 1186–1190. [\[CrossRef\]](#)
17. Fujita, Y.; Nakamura, Y.; Shiller, Z. Dual Dijkstra Search for paths with different topologies. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03), Taipei, Taiwan, 14–19 September 2003; Volume 3, pp. 3359–3364.
18. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
19. Stentz, A.; Mellon, I.C. Optimal and Efficient Path Planning for Unknown and Dynamic Environments. *Int. J. Robot. Autom.* **1993**, *10*, 89–100.
20. Stentz, A. The Focussed D* Algorithm for Real-Time Replanning. In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995; pp. 1652–1659.
21. Lavalle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report; Computer Science Department, Iowa State University: Ames, IA, USA, 1998.
22. LaValle, S.M.; Kuffner, J.J. Randomized Kinodynamic Planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. Available online: <http://journals.sagepub.com/doi/pdf/10.1177/02783640122067453> (accessed on 19 September 2018). [\[CrossRef\]](#)

23. Lavalley, S.M.; Kuffner, J.J., Jr. Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics: New Directions*; CRC Press: Boca Raton, FL, USA, 2000; pp. 293–308.
24. Hwang, Y.; Ahuja, N. A potential field approach to path planning. *IEEE Trans. Robot. Autom.* **1992**, *8*, 23–32. [[CrossRef](#)]
25. Kavraki, L.; Svestka, P.; Latombe, J.C.; Overmars, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [[CrossRef](#)]
26. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [[CrossRef](#)]
27. Shene, C.K. Continuity Issues. 2018. Available online: <http://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/curves/continuity.html> (accessed on 1 June 2018).
28. Guibas, L. Geometric Modeling. 2018. Available online: <http://graphics.stanford.edu/courses/cs348a-17-winter/ReaderNotes/handout27.pdf> (accessed on 1 June 2018).
29. Farin, G. *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2002.
30. De Boor, C. *A Practical Guide to Splines*; Springer Verlag: New York, NY, USA, 1978.
31. Waring, E. Problems concerning Interpolations. *Philos. Trans. R. Soc.* **1779**, *69*, 59–67. [[CrossRef](#)]
32. Waring, E. *Problems Concerning Interpolations*; The Royal Society Publishing: London, UK, 2015. Available online: <http://rstl.royalsocietypublishing.org/content/69/59.full.pdf+html> (accessed on 15 January 2016).
33. Weisstein, E.W. Lagrange Interpolating Polynomial from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html> (accessed on 1 June 2018).
34. Weisstein, E.W. Hermite’s Interpolating Polynomial from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/HermiteInterpolatingPolynomial.html> (accessed on 1 June 2018).
35. Song, B.; Tian, G.; Zhou, F. A comparison study on path smoothing algorithms for laser robot navigated mobile robot path planning in intelligent space. *J. Inf. Comput. Sci.* **2010**, *7*, 2943–2950.
36. Mathematical Interpolation. Wikipedia. 2016. Available online: <https://en.wikipedia.org/wiki/Interpolation> (accessed on 11 February 2016).
37. Epperson, J.F. On the Runge Example. *Am. Math. Mon.* **1987**, *94*, 329–341. [[CrossRef](#)]
38. Huh, U.Y.; Chang, S.R. A G^2 Continuous Path-smoothing Algorithm Using Modified Quadratic Polynomial Interpolation. *Int. J. Adv. Robot. Syst.* **2013**, *11*. [[CrossRef](#)]
39. Chang, S.R.; Huh, U.Y. A Collision-Free G^2 Continuous Path-Smoothing Algorithm Using Quadratic Polynomial Interpolation. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 194. [[CrossRef](#)]
40. Takahashi, A.; Hongo, T.; Ninomiya, Y.; Sugimoto, G. Local Path Planning and Motion Control for Agv in Positioning. In Proceedings of the Autonomous Mobile Robots and Its Applications, IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS ’89), Tsukuba, Japan, 4–6 September 1989; pp. 392–397.
41. Weisstein, E.W. Bernstein Polynomial from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/BernsteinPolynomial.html> (accessed on 1 June 2018).
42. Choi, J.W.; Curry, R.; Elkaim, G. Path Planning Based on Bezier Curve for Autonomous Ground Vehicles. In Proceedings of the Advances in Electrical and Electronics Engineering—IAENG Special Edition of the World Congress on Engineering and Computer Science 2008 (WCECS ’08), San Francisco, CA, USA, 22–24 October 2008; pp. 158–166.
43. Weisstein, E.W. Bezier Curve from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/BezierCurve.html> (accessed on 1 June 2018).
44. Kawabata, K.; Ma, L.; Xue, J.; Zhu, C.; Zheng, N. A Path Generation for Automated Vehicle Based on Bezier Curve and Via-points. *Robot. Auton. Syst.* **2015**, *74*, 243–252. [[CrossRef](#)]
45. Rastelli, J.P.; Lattarulo, R.; Nashashibi, F. Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 510–515.
46. Walton, D.; Meek, D.; Ali, J. Planar G^2 transition curves composed of cubic Bézier spiral segments. *J. Comput. Appl. Math.* **2003**, *157*, 453–476. [[CrossRef](#)]

47. Montes, N.; Herraiz, A.; Armesto, L.; Tornero, J. Real-time clothoid approximation by Rational Bezier curves. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 2246–2251.
48. Montes, N.; Mora, M.C.; Tornero, J. Trajectory Generation based on Rational Bezier Curves as Clothoids. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 505–510.
49. Han, L.; Yashiro, H.; Nejad, H.T.N.; Do, Q.H.; Mita, S. Bezier curve based path planning for autonomous vehicle in urban environment. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 1036–1042.
50. Pérez, J.; Godoy, J.; Villagrà, J.; Onieva, E. Trajectory generator for autonomous vehicles in urban environments. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 409–414.
51. González, D.; Pérez, J.; Lattarulo, R.; Milanés, V.; Nashashibi, F. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1430–1435.
52. Liang, Z.; Zheng, G.; Li, J. Automatic parking path optimization based on Bezier curve fitting. In Proceedings of the 2012 IEEE International Conference on Automation and Logistics, Zhengzhou, China, 15–17 August 2012; pp. 583–587.
53. Weisstein, E.W. Spline from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/Spline.html> (accessed on 1 June 2018).
54. Quarteroni, A.; Sacco, R.; Saleri, F. *Numerical Mathematics (Texts in Applied Mathematics)*; Springer-Verlag: Berlin/Heidelberg, Germany, 2006.
55. Komoriya, K.; Tanie, K. Trajectory Design and Control of a Wheel-type Mobile Robot Using B-spline Curve. In Proceedings of the Autonomous Mobile Robots and Its Applications, IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89), Tsukuba, Japan, 4–6 September 1989; pp. 398–405.
56. Trepagnier, P.G.; Nagel, J.; Kinney, P.M.; Koutsougeras, C.; Dooner, M. KAT-5: Robust Systems for Autonomous Vehicle Navigation in Challenging and Unknown Terrain. In *The 2005 DARPA Grand Challenge: The Great Robot Race*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 103–128.
57. Nagel, J.; Trepagnier, P.G.; Koutsougeras, C.; Kinney, P.M.; Dooner, M. The Culebra Algorithm for Path Planning and Obstacle Avoidance in Kat-5. In Proceedings of the 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '06), Arlington, VA, USA, 13–15 November 2006; pp. 247–253.
58. DARPA Grand Challenge. Wikipedia. 2018. Available online: https://en.wikipedia.org/wiki/DARPA_Grand_Challenge (accessed on 5 May 2018).
59. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. Stanley: The Robot That Won the DARPA Grand Challenge: Research Articles. *J. Robot. Syst.* **2006**, *23*, 661–692.
60. Bacha, A.; Bauman, C.; Faruque, R.; Fleming, M.; Terwelp, C.; Reinholtz, C.; Hong, D.; Wicks, A.; Alberi, T.; Anderson, D.; et al. Odin: Team VictorTango's Entry in the DARPA Urban Challenge. *J. Field Robot.* **2008**, *25*, 467–492. [[CrossRef](#)]
61. Berglund, T.; Brodnik, A.; Jonsson, H.; Staffanson, M.; Soderkvist, I. Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2010**, *7*, 167–172. [[CrossRef](#)]
62. Piazza, A.; Bianco, C.G.L.; Bertozzi, M.; Fascioli, A.; Broggi, A. Quintic G2-splines for the iterative steering of vision-based autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 27–36. [[CrossRef](#)]
63. Delingette, H.; Hebert, M.; Ikeuchi, K. Trajectory generation with curvature constraint based on energy minimization. In Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, Intelligence for Mechanical Systems (IROS '91), Osaka, Japan, 3–5 November 1991; Volume 1, pp. 206–211.
64. Yang, K.; Sukkarieh, S. An Analytical Continuous-Curvature Path-Smoothing Algorithm. *IEEE Trans. Robot.* **2010**, *26*, 561–568. [[CrossRef](#)]
65. Elbanhawi, M.; Simic, M.; Jazar, R. Randomized Bidirectional B-Spline Parameterization Motion Planning. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 406–419. [[CrossRef](#)]
66. Piegl, L.; Tiller, W. *The NURBS Book*, 2nd ed.; Springer-Verlag: New York, NY, USA, 1996.
67. Weisstein, E.W. NURBS from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/NURBSCurve.html> (accessed on 1 June 2018).

68. Bingol, O.R. *NURBS-Python*; 2016; Available online: <https://nurbs-python.readthedocs.io/en/latest/> (accessed on 19 September 2018).
69. Piegl, L. On NURBS: A Survey. *IEEE Comput. Graph. Appl.* **1991**, *11*, 55–71. [[CrossRef](#)]
70. Herrmann, P.; Gerngroß, M.; Endisch, C. NURBS based trajectory generation for an industrial five axis needle winding robot. In Proceedings of the 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 20–23 April 2018; pp. 31–36.
71. Ravari, A.N.; Taghirad, H.D. NURBS-based representation of urban environments for mobile robots. In Proceedings of the 2016 4th International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 26–28 October 2016; pp. 20–25.
72. Lai, T.C.; Xiao, S.R.; Aoyama, H.; Wong, C.C. Path planning and obstacle avoidance approaches for robot arm. In Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, Japan, 19–22 September 2017; pp. 334–337.
73. Shi, X.; Fang, H.; Guo, L. Multi-objective optimal trajectory planning of manipulators based on quintic NURBS. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 759–765.
74. Belaidi, H.; Hentout, A.; Bouzouia, B.; Bentarzi, H.; Belaidi, A. NURBS trajectory generation and following by an autonomous mobile robot navigating in 3D environment. In Proceedings of the 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, Hong Kong, China, 4–7 June 2014; pp. 168–173.
75. Tatematsu, N.; Ohnishi, K. Tracking motion of mobile robot for moving target using NURBS curve. In Proceedings of the IEEE International Conference on Industrial Technology, Maribor, Slovenia, 10–12 December 2003; Volume 1, pp. 245–249.
76. Aleotti, J.; Caselli, S.; Maccherozzi, G. Trajectory reconstruction with NURBS curves for robot programming by demonstration. In Proceedings of the 2005 International Symposium on Computational Intelligence in Robotics and Automation, Espoo, Finland, 27–30 June 2005; pp. 73–78.
77. Guo, H.; Meng, Y.; Jin, Y. Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3205–3210.
78. Schmid, A.J.; Woern, H. Path planning for a humanoid using NURBS curves. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Edmonton, AB, Canada, 1–2 August 2005; pp. 351–356.
79. Richtsfeld, A.; Mörwald, T.; Prankl, J.; Zillich, M.; Vincze, M. Segmentation of unknown objects in indoor environments. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 4791–4796.
80. Farin, G. From conics to NURBS: A tutorial and survey. *IEEE Comput. Graph. Appl.* **1992**, *12*, 78–86. [[CrossRef](#)]
81. Lockwood, E.H. *Book of Curves*; Cambridge Books; Cambridge University Press: Cambridge, UK, 1961.
82. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *Am. J. Math.* **1957**, *79*, 497–516. [[CrossRef](#)]
83. Reeds, J.A.; Shepp, L.A. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [[CrossRef](#)]
84. Yang, D.; Li, D.; Sun, H. 2D Dubins Path in Environments with Obstacle. *Math. Probl. Eng.* **2013**, *2013*, 291372. [[CrossRef](#)]
85. Gerlach, A.R.; Kingston, D.; Walker, B.K. UAV navigation using predictive vector field control. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 4907–4912.
86. Lin, Y.; Saripalli, S. Path planning using 3D Dubins Curve for Unmanned Aerial Vehicles. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 296–304.
87. Hwangbo, M.; Kuffner, J.; Kanade, T. Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 1035–1041.

88. Lugo-Cárdenas, I.; Flores, G.; Salazar, S.; Lozano, R. Dubins path generation for a fixed wing UAV. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 339–346.
89. Parlangeli, G.; Ostuni, L.; Mancarella, L.; Indiveri, G. A motion planning algorithm for smooth paths of bounded curvature and curvature derivative. In Proceedings of the 2009 17th Mediterranean Conference on Control and Automation, Thessaloniki, Greece, 24–26 June 2009; pp. 73–78.
90. Klein, M.; Klos, A.; Lenhardt, J.; Schiffmann, W. Wind-Aware Emergency Landing Assistant Based on Dubins Curves. In Proceedings of the 2017 5th International Symposium on Computing and Networking (CANDAR), Aomori, Japan, 19–22 November 2017; pp. 546–550.
91. Hameed, I.A. Coverage path planning software for autonomous robotic lawn mower using Dubins' curve. In Proceedings of the 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), Okinawa, Japan, 14–18 July 2017; pp. 517–522.
92. Fermat's Spiral. Wikipedia. 2018. Available online: https://en.wikipedia.org/wiki/Fermat's_spiral (accessed on 1 May 2018).
93. Chen, X.; Zhang, J.; Yang, M.; Zhong, L.; Dong, J. Continuous-Curvature Path Generation Using Fermat's Spiral for Unmanned Marine and Aerial Vehicles. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 4911–4916.
94. Weisstein, E.W. Cornu Spiral from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/CornuSpiral.html> (accessed on 1 June 2018).
95. Weisstein, E.W. Fresnel Integrals from MathWorld—A Wolfram Web Resource. 2018. Available online: <http://mathworld.wolfram.com/FresnelIntegrals.html> (accessed on 1 June 2018).
96. Gim, S.; Adouane, L.; Lee, S.; Déruvin, J.P. Clothoids Composition Method for Smooth Path Generation of Car-Like Vehicle Navigation. *J. Intell. Robot. Syst.* **2017**, *88*, 129–146. [CrossRef]
97. Fraichard, T.; Scheuer, A. From Reeds and Shepp's to continuous-curvature paths. *IEEE Trans. Robot.* **2004**, *20*, 1025–1035. [CrossRef]
98. Liscano, R.; Green, D. Design and Implementation of a Trajectory Generator for an Indoor Mobile Robot. In Proceedings of the Autonomous Mobile Robots and Its Applications, IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89), Tsukuba, Japan, 4–6 September 1989; pp. 380–385.
99. Dai, J.; Wang, Y.; Bortoff, S.A.; Burns, D.J. From reeds-shepp's paths to continuous curvature paths-Part I: Transition schemes algorithms. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Mauna Lani, HI, USA, 27–30 August 2017; pp. 355–362.
100. Behringer, R.; Muller, N. Autonomous road vehicle guidance from autobahnen to narrow curves. *IEEE Trans. Robot. Autom.* **1998**, *14*, 810–815. [CrossRef]
101. Kim, C.H.; Jeong, K.M.; Jeong, T.W. Semi-autonomous navigation of an unmanned ground vehicle for bird expellant in an airport. In Proceedings of the 2012 12th International Conference on Control, Automation and Systems, JeJu Island, Korea, 17–21 October 2012; pp. 2063–2067.
102. Funke, J.; Theodosis, P.; Hindiyeh, R.; Stanek, G.; Kritatakirana, K.; Gerdes, C.; Langer, D.; Hernandez, M.; Müller-Bessler, B.; Huhnke, B. Up to the limits: Autonomous Audi TTS. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Spain, 3–7 June 2012; pp. 541–547.
103. Broggi, A.; Medici, P.; Zani, P.; Coati, A.; Panciroli, M. Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. *Annu. Rev. Control* **2012**, *36*, 161–171. [CrossRef]
104. Vorobieva, H.; Minoiu-Enache, N.; Glaser, S.; Mammar, S. Geometric continuous-curvature path planning for automatic parallel parking. In Proceedings of the 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), Evry, France, 10–12 April 2013; pp. 418–423.
105. Vorobieva, H.; Glaser, S.; Minoiu-Enache, N.; Mammar, S. Automatic parallel parking with geometric continuous-curvature path planning. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 465–471.
106. Fuji, H.; Xiang, J.; Tazaki, Y.; Levedahl, B.; Suzuki, T. Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 407–413.
107. Brezak, M.; Petrović, I. Real-time Approximation of Clothoids With Bounded Error for Path Planning Applications. *IEEE Trans. Robot.* **2014**, *30*, 507–515. [CrossRef]

108. Coombs, D.; Murphy, K.; Lacaze, A.; Legowik, S. Driving autonomously off-road up to 35 km/h. In Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511), Dearborn, MI, USA, 5 October 2000; pp. 186–191.
109. Bevilacqua, P.; Frego, M.; Fontanelli, D.; Palopoli, L. Reactive Planning for Assistive Robots. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1276–1283. [[CrossRef](#)]
110. Kim, Y.H.; Park, J.B.; Son, W.S.; Yoon, T.S. Modified turn algorithm for motion planning based on clothoid curve. *Electron. Lett.* **2017**, *53*, 1574–1576. [[CrossRef](#)]
111. Kim, K.B.; Kim, B.K. Minimum-Time Trajectory for Three-Wheeled Omnidirectional Mobile Robots Following a Bounded-Curvature Path With a Referenced Heading Profile. *IEEE Trans. Robot.* **2011**, *27*, 800–808. [[CrossRef](#)]
112. Liu, C.A.; Cheng, W.G.; Hong, Z. A Trajectory Generator for a Mobile Robot in 3D Pathplanning. In Proceedings of the 2007 IEEE International Conference on Automation and Logistics, Jinan, China, 18–21 August 2007; pp. 1247–1251.
113. Alia, C.; Reine, T.; Ali, C. Maneuver planning for autonomous vehicles, with clothoid tentacles for local trajectory planning. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
114. Otto, M.; Kramer, M. Implementation of a clothoid based trajectory into the ROS framework. In Proceedings of the 2016 International Conference on Control, Decision and Information Technologies (CoDIT), St. Julian's, Malta, 6–8 April 2016; pp. 345–348.
115. Quigley, M.; Conley, K.; Gerkey, B.P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
116. Weisstein, E.W. Hypocycloid from MathWorld—A Wolfram Web Resource. 2016. Available online: <http://mathworld.wolfram.com/Hypocycloid.html> (accessed on 3 March 2016).
117. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Emaru, T. SHP: Smooth Hypocycloidal Paths with Collision-Free and Decoupled Multi-Robot Path Planning. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 133. [[CrossRef](#)]
118. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Emaru, T. Path Smoothing Extension for Various Robot Path Planners. In Proceedings of the 16th IEEE International Conference on Control, Automation and Systems, Gyeongju, Korea, 16–19 October 2016.
119. Campana, M.; Lamiroux, F.; Laumond, J.P. *A Simple Path Optimization Method for Motion Planning*; Working Paper or Preprint. HAL Archives; Rapport LAAS n° 15108. 2015. <hal-01137844v2>. Available online: https://hal.archives-ouvertes.fr/hal-01137844/file/paper_icra2016_hal.pdf (accessed on 19 September 2018).
120. Park, C.; Pan, J.; Manocha, D. ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments. In Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012), Atibaia, Sao Paulo, Brazil, 25–29 June 2012; pp. 207–215.
121. Garber, M.; Lin, M.C. Constraint-Based Motion Planning Using Voronoi Diagrams. In *Algorithmic Foundations of Robotics V*; Boissonnat, J.D., Burdick, J., Goldberg, K., Hutchinson, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 541–558.
122. Richardson, A.; Olson, E. Iterative path optimization for practical robot planning. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3881–3886.
123. Zhu, Z.; Schmerling, E.; Pavone, M. A convex optimization approach to smooth trajectories for motion planning with car-like robots. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 835–842.
124. Choi, J.; Huhtala, K. Constrained path optimization with Bézier curve primitives. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 246–251.
125. Rösmann, C.; Hoffmann, F.; Bertram, T. Integrated online trajectory planning and optimization in distinctive topologies. *Robot. Auton. Syst.* **2017**, *88*, 142–153. [[CrossRef](#)]
126. Roesmann, C.; Feiten, W.; Woesch, T.; Hoffmann, F.; Bertram, T. Trajectory modification considering dynamic constraints of autonomous robots. In Proceedings of the 7th German Conference on Robotics (ROBOTIK 2012), Munich, Germany, 21–22 May 2012; pp. 1–6.

127. Rösmann, C.; Feiten, W.; Wösch, T.; Hoffmann, F.; Bertram, T. Efficient trajectory optimization using a sparse model. In Proceedings of the 2013 European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 138–143.
128. Rösmann, C.; Hoffmann, F.; Bertram, T. Planning of multiple robot trajectories in distinctive topologies. In Proceedings of the 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2–4 September 2015; pp. 1–6.
129. Rösmann, C.; Hoffmann, F.; Bertram, T. Kinodynamic trajectory optimization and control for car-like robots. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 5681–5686.
130. Rösmann, C. TEB Planner. 2018. Available online: http://wiki.ros.org/teb_local_planner (accessed on 1 May 2018).
131. Rösmann, C. TEB Planner. 2018. Available online: https://github.com/rst-tu-dortmund/teb_local_planner (accessed on 1 May 2018).
132. Kogan, D.; Murray, R.M. Optimization-Based Navigation for the DARPA Grand Challenge. In Proceedings of the 2006 Conference on Decision and Control (CDC), San Diego, CA, USA, 13–15 December 2006.
133. Cremean, L.B.; Foote, T.B.; Gillula, J.H.; Hines, G.H.; Kogan, D.; Kriechbaum, K.L.; Lamb, J.C.; Leibs, J.; Lindzey, L.; Rasmussen, C.E.; et al. Alice: An Information-Rich Autonomous Vehicle for High-Speed Desert Navigation. In *The 2005 DARPA Grand Challenge: The Great Robot Race*; Buehler, M., Iagnemma, K., Singh, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 437–482.
134. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments. *Int. J. Robot. Res.* **2010**, *29*, 485–501. [CrossRef]
135. Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making Bertha Drive; An Autonomous Journey on a Historic Route. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 8–20. [CrossRef]
136. Ziegler, J.; Bender, P.; Dang, T.; Stiller, C. Trajectory planning for Bertha—A local, continuous method. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 450–457.
137. Gu, T.; Dolan, J.M. On-Road Motion Planning for Autonomous Vehicles. In *Intelligent Robotics and Applications*; Su, C.Y., Rakheja, S., Liu, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 588–597.
138. Howard, C.R. *The Transition-Curve Field-Book*; Palala Press: Warsaw, Poland, 2015; ISBN 1347030956.
139. Habib, Z.; Sakai, M. Family of G2 cubic transition curves. In Proceedings of the 2003 International Conference on Geometric Modeling and Graphics, London, UK, 16–18 July 2003; pp. 117–122.
140. Ahmad, A.; Gobithasan, R.; Ali, J.M. G2 Transition curve using Quartic Bezier Curve. In Proceedings of the Computer Graphics, Imaging and Visualisation (CGIV '07), Bangkok, Thailand, 14–17 August 2007; pp. 223–228.
141. Ahmad, A.; Ali, J.M. G3 Transition Curve Between Two Straight Lines. In Proceedings of the 5th International Conference on Computer Graphics, Imaging and Visualisation, (CGIV '08), Penang, Malaysia, 26–28 August 2008; pp. 154–159.
142. Srivastava, S.; Roychowdhury, J. Independent and Interdependent Latch Setup/Hold Time Characterization via Newton Raphson Solution and Euler Curve Tracking of State-Transition Equations. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2008**, *27*, 817–830. [CrossRef]
143. Quinlan, S.; Khatib, O. Elastic bands: Connecting path planning and control. In Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; Volume 2, pp. 802–807.
144. Quinlan, S. Real-Time Modification of Collision-Free Paths. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 1995.
145. Khatib, M. *Sensor-Based Motion Control for Mobile Robots*; LAAS-CNRS: Toulouse, France, 1996; Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.668&rep=rep1&type=pdf> (accessed on 19 September 2018).
146. Khatib, M.; Jaouni, H.; Chatila, R.; Laumond, J.P. Dynamic path modification for car-like nonholonomic mobile robots. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 25 April 1997; Volume 4, pp. 2920–2925.

147. Graf, B.; Manuel Hostalet, W.J.; Schaeffer, C. Flexible Path Planning for Nonholonomic Mobile Robots. In Proceedings of the Fourth European Workshop on Advanced Mobile Robots, Lund, Sweden, 19–21 September 2001.
148. Brock, O.; Khatib, O. Executing motion plans for robots with many degrees of freedom in dynamic environments. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 20–20 May 1998; Volume 1, pp. 1–6.
149. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [[CrossRef](#)]
150. Wray, K.H.; Ruiken, D.; Grupen, R.A.; Zilberstein, S. Log-space harmonic function path planning. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1511–1516.
151. Hong, R.; DeSouza, G.N. A real-time path planner for a smart wheelchair using harmonic potentials and a rubber band model. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3282–3287.
152. Garrido, S.; Moreno, L.; Blanco, D. Exploration and Mapping Using the VFM Motion Planner. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 2880–2892. [[CrossRef](#)]
153. Minguez, J.; Montano, L. Extending Collision Avoidance Methods to Consider the Vehicle Shape. Kinematics, and Dynamics of a Mobile Robot. *IEEE Trans. Robot.* **2009**, *25*, 367–381. [[CrossRef](#)]
154. Borenstein, J.; Koren, Y. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, 1179–1187. [[CrossRef](#)]
155. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [[CrossRef](#)]
156. Ulrich, I.; Borenstein, J. VFH+: Reliable obstacle avoidance for fast mobile robots. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), Leuven, Belgium, 20 May 1998; Volume 2, pp. 1572–1577.
157. Ulrich, I.; Borenstein, J. VFH/sup*/: Local obstacle avoidance with look-ahead verification. In Proceedings of the IEEE International Conference on Robotics and Automation, Millennium Conference (2000 ICRA) (Cat.No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 3, pp. 2505–2511.
158. Jie, D.; Xueming, M.; Kaixiang, P. IVFH*: Real-time dynamic obstacle avoidance for mobile robots. In Proceedings of the 2010 11th International Conference on Control Automation Robotics Vision, Singapore, 7–10 December 2010; pp. 844–847.
159. Babinec, A.; Duchoň, F.; Dekan, M.; Pászto, P.; Kelemen, M. VFH*TDT (VFH* with Time Dependent Tree). *Robot. Auton. Syst.* **2014**, *62*, 1098–1115. [[CrossRef](#)]
160. Molinos, E.; Llamazares, A.; Ocana, M.; Herranz, F. Dynamic obstacle avoidance based on curvature arcs. In Proceedings of the 2014 IEEE/SICE International Symposium on System Integration, Tokyo, Japan, 13–15 December 2014; pp. 186–191.
161. Brock, O.; Khatib, O. High-speed navigation using the global dynamic window approach. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 1, pp. 341–346.
162. Ogren, P.; Leonard, N.E. A convergent dynamic window approach to obstacle avoidance. *IEEE Trans. Robot.* **2005**, *21*, 188–195. [[CrossRef](#)]
163. Minguez, J.; Montano, L. Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Trans. Robot. Autom.* **2004**, *20*, 45–59. [[CrossRef](#)]
164. Minguez, J.; Montano, L. Nearness diagram navigation (ND): A new real time collision avoidance approach. In Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113), Takamatsu, Japan, 31 October–5 November 2000; Volume 3, pp. 2094–2100.
165. Minguez, J.; Montano, L.; Simeon, T.; Alami, R. Global nearness diagram navigation (GND). In Proceedings of the IEEE International Conference on Robotics and Automation (2001 ICRA) (Cat. No.01CH37164), Seoul, Korea, 21–26 May 2001; Volume 1, pp. 33–39.
166. Li, G.; Wu, G.; Wei, W. ND-DWA: A Reactive Method for Collision Avoidance in Troublesome Scenarios. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 2, pp. 9307–9311.

167. Durham, J.W.; Bullo, F. Smooth Nearness-Diagram Navigation. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 690–695.
168. Kamil, F.; Tang, S.; Khaksar, W.; Zulkifli, N.; Ahmad, S.A. A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments. *Adv. Robot. Autom.* **2015**, *4*, 134–142.
169. Gandhi, T.; Trivedi, M.M. Pedestrian Protection Systems: Issues, Survey, and Challenges. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 413–430. [[CrossRef](#)]
170. Zheng, G.; Chen, Y. A review on vision-based pedestrian detection. In Proceedings of the 2012 IEEE Global High Tech Congress on Electronics, Shenzhen, China, 18–20 November 2012; pp. 49–54.
171. Ko, B.C.; Kwak, J.; Nam, J. Online learning based multiple pedestrians tracking in thermal imagery for safe driving at night. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 78–79.
172. Zhu, C.; Peng, Y. A Boosted Multi-Task Model for Pedestrian Detection with Occlusion Handling. *IEEE Trans. Image Process.* **2015**, *24*, 5619–5629. [[CrossRef](#)] [[PubMed](#)]
173. Ghosh, S.; Amon, P.; Hutter, A.; Kaup, A. Detecting closely spaced and occluded pedestrians using specialized deep models for counting. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
174. Ravankar, A.A.; Hoshino, Y.; Emaru, T.; Kobayashi, Y. Map building from laser range sensor information using mixed data clustering and singular value decomposition in noisy environment. In Proceedings of the 2011 IEEE/SICE International Symposium on System Integration (SII), Kyoto, Japan, 20–22 December 2011; pp. 1232–1238.
175. Xu, W.; Pan, J.; Wei, J.; Dolan, J.M. Motion planning under uncertainty for on-road autonomous driving. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2507–2512.
176. Anderson, S.; MacTavish, K.; Barfoot, T.D. Relative Continuous-time SLAM. *Int. J. Robot. Res.* **2015**, *34*, 1453–1479. [[CrossRef](#)]
177. Anderson, S.; Barfoot, T.D. Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3). In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 157–164.
178. Dubé, R.; Sommer, H.; Gawel, A.; Bosse, M.; Siegwart, R. Non-uniform sampling strategies for continuous correction based trajectory estimation. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4792–4798.
179. Park, C.; Moghadam, P.; Kim, S.; Elfes, A.; Fookes, C.; Sridharan, S. Elastic LiDAR Fusion: Dense Map-Centric Continuous-Time SLAM. *arXiv* **2017**, arXiv:1711.01691.
180. Anderson, S.W. Batch Continuous-Time Trajectory Estimation. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2017.
181. Anderson, S.; Barfoot, T.D.; Tong, C.H.; Särkkä, S. Batch Nonlinear Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression. *arXiv* **2014**, arXiv:1412.0630.
182. Droeschel, D.; Schwarz, M.; Behnke, S. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robot. Auton. Syst.* **2017**, *88*, 104–115. [[CrossRef](#)]
183. Nüchter, A.; Bleier, M.; Schauer, J.; Janotta, P. Continuous-Time SLAM—Improving Google’s Cartographer 3D Mapping. In *Latest Developments in Reality-Based 3D Surveying and Modelling*; Chapter Continuous-Time SLAM—Improving Google’s Cartographer 3D Mapping; Remondino, F., Georgopoulos, A., Gonzalez-Aguilera, D., Agrafiotis, P., Eds.; MDPI: Basel, Switzerland, 2018; pp. 53–73.
184. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
185. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145. [[CrossRef](#)]
186. Robla-Gómez, S.; Becerra, V.M.; Llata, J.R.; González-Sarabia, E.; Torre-Ferrero, C.; Pérez-Oria, J. Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments. *IEEE Access* **2017**, *5*, 26754–26773. [[CrossRef](#)]

187. Lisetti, C.L.; Brown, S.M.; Alvarez, K.; Marpaung, A.H. A social informatics approach to human-robot interaction with a service social robot. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2004**, *34*, 195–209. [[CrossRef](#)]
188. Vanholme, B.; Glaser, S.; Mammar, S.; Gruyer, D. Manoeuvre-based trajectory planning for highly autonomous vehicles on real road with traffic. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 3281–3286.
189. Glaser, S.; Vanholme, B.; Mammar, S.; Gruyer, D.; Nouveliere, L. Maneuver-Based Trajectory Planning for Highly Autonomous Vehicles on Real Road With Traffic and Driver Interaction. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 589–606. [[CrossRef](#)]
190. Bhattacharya, P.; Gavrilova, M. Roadmap-Based Path Planning—Using the Voronoi Diagram for a Clearance-Based Shortest Path. *IEEE Robot. Autom. Mag.* **2008**, *15*, 58–66. [[CrossRef](#)]
191. Arcelli, C.; Di Baja, G.S. A Width-Independent Fast Thinning Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1985**, *PAMI-7*, 463–474. [[CrossRef](#)]
192. Zhang, T.Y.; Suen, C.Y. A Fast Parallel Algorithm for Thinning Digital Patterns. *Commun. ACM* **1984**, *27*, 236–239. [[CrossRef](#)]
193. Sakellariou, G.; Shanahan, M.; Kuipers, B. Skeletonisation as mobile robot navigation. In Proceedings of the Towards Autonomous Robotic Systems (TAROS-04), Essex, UK, 6–8 September 2004; pp. 149–155.
194. Bai, X.; Latecki, L.; Liu, W.Y. Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 449–462. [[CrossRef](#)] [[PubMed](#)]
195. Elbanhawi, M.; Simic, M.; Jazar, R. In the Passenger Seat: Investigating Ride Comfort Measures in Autonomous Cars. *IEEE Intell. Transp. Syst. Mag.* **2015**, *7*, 4–17. [[CrossRef](#)]
196. Morales, Y.; Abdur-Rahim, J.; Watanabe, A.; Even, J. Analysis of navigational habituation. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3056–3062.
197. Sawabe, T.; Okajima, T.; Kanbara, M.; Hagita, N. Evaluating passenger characteristics for ride comfort in autonomous wheelchairs. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 102–107.
198. Chang, C.; Lv, C.; Wang, H.; Wang, H.; Cao, D.; Velenis, E.; Wang, F. Multi-point turn decision making framework for human-like automated driving. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
199. Nagasaka, N.; Harada, M. Towards safe, smooth, and stable path planning for on-road autonomous driving under uncertainty. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 795–801.
200. Sawabe, T.; Kanbara, M.; Hagita, N. Diminished Reality for Acceleration—Motion Sickness Reduction with Vection for Autonomous Driving. In Proceedings of the 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Merida, Mexico, 19–23 September 2016; pp. 297–299.

