applied sciences

MDPI

Article

A $\mathbb{G}^3$-Continuous Extend Procedure for Path Planning of Mobile Robots with Limited Motion Curvature and State Constraints

Tomasz Gawron * and Maciej Marcin Michałek

Institute of Automation and Robotics, Poznań University of Technology (PUT), Piotrowo 3A, 60-965 Poznań, Poland; maciej.michalek@put.poznan.pl

* Correspondence: tomasz.gawron@put.poznan.pl

Received: 19 October 2018; Accepted: 30 October 2018; Published: 2 November 2018

Featured Application: The proposed method can be applied to planning of reference paths for autonomous wheeled robots and intelligent vehicles maneuvering in cluttered environments.

Abstract: Provably correct and computationally efficient path planning in the presence of various constraints is essential for autonomous driving and agile maneuvering of mobile robots. In this paper, we consider the planning of $\mathbb{G}^3$-continuous planar paths with continuous and limited curvature in a motion environment that is bounded and contains obstacles modeled by a set of (non-convex) polygons. In practice, the curvature constraints often arise from mechanical limitations for the robot, such as limited steering and articulation angles in wheeled robots, or aerodynamic constraints in unmanned aerial vehicles. To solve the planning problem under those stringent constraints, we improve upon known path primitives, such as Reeds–Shepp (RS) and CC-steer (curvature-continuous) paths. Given the initial and final robot configuration, we developed extend-procedure computing paths that can approximate RS paths with arbitrary precision, but guaranteeing $\mathbb{G}^3$-continuity. We show that satisfaction of all stated path constraints is guaranteed and, contrary to many other methods known from the literature, the method of checking for collisions between the planned path and obstacles is given by a closed-form analytic expression. Furthermore, we demonstrate that our approach is not conservative, i.e., it allows for precise maneuvers in tight environments under the assumption of a rectangular robot footprint. The presented extend procedure can be integrated into various motion-planning algorithms available in the literature. In particular, we utilized the Rapidly exploring Random Trees (RRT*) algorithm in conjunction with our extend procedure to demonstrate its feasibility in motion environments of nontrivial complexity and low computational cost in comparison to a $\mathbb{G}^3$-continuous extend procedure based on η^3-splines.

Keywords: path planning; mobile robots; curvature constraints; state constraints; extend procedure; $\mathbb{G}^3$-continuity; car-like kinematics

1. Introduction

In this paper, we focus on the development of a path primitive and the so-called extend procedure (i.e., a local planning algorithm generating a path connecting two robot configurations), which is crucial for many path-planning algorithms utilized in the navigation of mobile robots. Despite a large body of work concerning path planning for mobile robots and autonomous vehicles (e.g., References [1,2]), this problem remains a challenge, especially in the presence of various constraints arising in practical scenarios. On one hand, mechanical limitations of the robot, such as limited steering and articulation angles in wheeled robots, or aerodynamic constraints in unmanned aerial vehicles, result in path curvature limits. On the other hand, the presence of a bounded-motion environment with obstacles and forbidden areas leads to state constraints imposed on the robot, which reduces the set of feasible paths. It is also important to maintain a high degree of path continuity to achieve smooth control of the robot and increase the comfort of passengers or the safety of the payload. To account for all these constraints, we built upon our approach introduced in Reference [3], where we proposed the extend procedure generating $\mathbb{G}^3$-continuous planar paths (that is, paths with continuous curvature derivative with respect to curve arc length) taking into account a limited curvature of motion in cluttered environments. In contrast to Reference [3], we present the new extend procedure for the carlike kinematics taking into account vehicle-body dimensions in planning collision-free paths, admitting the nonconvex polygonal obstacles present in the operational space. As a consequence, the motion-planning strategy presented in the current paper inherits beneficial properties of the original approach presented in Reference [3], but extends its potential applications to more practical path-planning scenarios.

2. Prerequisites and Problem Statement

While our considerations are quite general, meaning that the planar paths planned with our approach can be applied to various systems and planning tasks, let us consider a rear-driven carlike-vehicle kinematics as an illustratory example used throughout this paper. This is shown in Figure 1. Using results from Reference [4], one can model this system by decomposition into unicycle vehicle-body kinematics $\dot{\bar{\boldsymbol{q}}} = \boldsymbol{G}(\theta)\boldsymbol{v}$ and steering dynamics $\dot{\beta} = u_1$ as follows:

$$\dot{\bar{\boldsymbol{q}}} = \begin{bmatrix} 1 & 0 \\ 0 & \cos\theta \\ 0 & \sin\theta \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \boldsymbol{G}(\bar{\boldsymbol{q}})\boldsymbol{v}, \tag{1}$$

$$\dot{\beta} = u_1, \quad \boldsymbol{v} = [v_1\ v_2] \triangleq \begin{bmatrix} \frac{u_2}{L}\tan\beta & u_2 \end{bmatrix}^\top \in \mathbb{R}^2, \tag{2}$$
$$\boldsymbol{u} = [u_1\ u_2]^\top \in \mathbb{R}^2,$$

where $\boldsymbol{q} = [\beta\ \theta\ x\ y]^\top = [\beta\ \bar{\boldsymbol{q}}^\top]^\top = [\beta\ \theta\ \tilde{\boldsymbol{q}}^\top]^\top \in \mathcal{Q} = [-\beta_m, \beta_m] \times \mathbb{R} \times \mathcal{P}$ denotes a configuration vector with $\beta_m < \pi/2$ being a steering angle limit, $\mathcal{P} \subseteq \mathbb{R}^2$ denotes a position space, $\boldsymbol{v}$ corresponds to vehicle-body kinematics-control input, with v_1 being vehicle-body angular velocity, and v_2 corresponding to longitudinal velocity of the guidance point $\tilde{\boldsymbol{q}} = [x\ y]^\top$ illustrated in Figure 1, while $\boldsymbol{u}$ is the control input of carlike kinematics comprising steering angle rate u_1 and longitudinal velocity $u_2 \equiv v_2$. As a result of limited steering angle $\beta \in [-\beta_m, \beta_m]$, the following constraint is present:

$$|\kappa| \leq \frac{1}{L}\tan\beta_m =: \kappa_B, \tag{3}$$

where

$$\kappa \triangleq \frac{v_1}{v_2} = \frac{1}{L}\tan\beta \tag{4}$$

is the motion curvature of the robot, whereas L denotes the distance between the rear and the front axle (see Figure 1).

Let us assume that control input $\boldsymbol{u}$ is continuous, which is often desirable in practical applications. According to Equation (2), this implies that $\dot{\beta}$ is continuous. Since $\dot{\beta}$ is related to curvature $\kappa(t)$ by Equation (4), one concludes that admissible trajectories of the carlike kinematics must have curvature $\kappa(t)$ of class C^1 (with a continuous time derivative), which satisfies Equation (3). As a consequence of such requirements, admissible paths for the carlike kinematics must be $\mathbb{G}^3$-continuous, that is, for a path $\tilde{\boldsymbol{q}}_d(s) = [x_d(s)\, y_d(s)]^\top$, its curvature $\kappa_d(s)$ must be at least of class C^1. In this paper we consider three problems of planning such $\mathbb{G}^3$-continuous paths. They shall be solved under the following assumptions:

A1. Planned positional path $\tilde{\boldsymbol{q}}_d(s)$ starts at $s = 0$ and finishes at $s = s_{fin}$, i.e., $s \in [0, s_{fin}]$, $s_{fin} \geq 0$, where s corresponds to the arc-length parameter.
A2. Initial reference steering angle $\beta_d(s = 0)$ and final reference steering angle $\beta_d(s = s_{fin})$ are fixed to 0, i.e., $\beta_d(s = 0) = \beta_d(s = s_{fin}) = 0$.
A3. Initial steering rate $\frac{d\beta_d}{ds}(s = 0)$ and final steering rate $\frac{d\beta_d}{ds}(s_{fin})$ are fixed to 0, i.e., $\frac{d\beta_d}{ds}(0) = \frac{d\beta_d}{ds}(s_{fin}) = 0$.
A4. $\mathcal{P}_f \subseteq \mathcal{P}$ is a free subset of position space bounded by a single (nonconvex) polygon and containing (nonconvex) polygonal obstacles.

Assumption A1 is a consequence of not knowing the resultant path length in advance. Assumptions A2 and A3 have been taken for simplicity of considerations since they help to fix the path structure. Assumption A2 can be lifted by simply changing the initial order of path segments, as explained in the description of the proposed extend procedure. Assumption A4 is used to efficiently check for satisfaction of state constraints by the reference path.

Problem 1. *Feasible path planning in free space. Given a collision-free initial and vehicle-body configurations $\bar{\boldsymbol{q}}_{dinit}$ and $\bar{\boldsymbol{q}}_{dfin}$ plan a $\mathbb{G}^3$-continuous path $\tilde{\boldsymbol{q}}_d(s)$ admissible for the car-like kinematics, such that $\bar{\boldsymbol{q}}_d(s = 0) = \bar{\boldsymbol{q}}_{dinit}$ and $\exists s_{fin} \;\; \bar{\boldsymbol{q}}(s_{fin}) = \bar{\boldsymbol{q}}_{dfin}$. Curvature $\kappa_d(s)$ of path $\tilde{\boldsymbol{q}}_d(s)$, is limited as follows*

$$\forall s \in [0, s_{fin}] \;\; |\kappa_d(s)| \leq \kappa_B. \tag{5}$$

Problem 2. *Feasible path planning. Solve Problem 1 by planning a collision-free path, that is, the additional constraint*

$$\forall s \in [0, s_{fin}] \;\; \tilde{\boldsymbol{q}}_d(s) \in \mathcal{P}_f$$

shall be satisfied.

Problem 3. *Feasible path planning with rectangular footprint. Solve Problem 1 by planning a collision-free path for a robot with rectangular footprint, that is, the reference path shall additionally satisfy*

$$\forall s \in [0, s_{fin}] \;\; \mathcal{V}(\bar{\boldsymbol{q}}_d(s)) \in \mathcal{P}_f, \tag{6}$$

where $\bar{\boldsymbol{q}}_d(s) = [\theta_d(s) \tilde{\boldsymbol{q}}_d(s)]^\top$ is the reference vehicle-body configuration along the path, and $\theta_d(s)$ is the reference robot orientation tangent to path $\tilde{\boldsymbol{q}}_d(s)$, while $\mathcal{V}(\bar{\boldsymbol{q}}_d(s)) \subsetneq \mathbb{R}^2$ is a position space subset occupied by rectangular footprint of the robot (see Figure 1). Rectangular footprint $\mathcal{V}(\bar{\boldsymbol{q}}_d(s))$ can be expressed in local coordinate frame $\{L\}$ of reference vehicle-body configuration $\bar{\boldsymbol{q}}_d(s)$ as follows:

$$\mathcal{V}^L \triangleq \{x^L, y^L : -c \leq \; x^L \leq a \wedge \left| y^L \right| \leq b/2\}.$$

Note that even though we only plan a position path $\tilde{\boldsymbol{q}}_d(s)$, orientation component $\theta_d(s)$ is known due to the differential flatness of vehicle-body kinematics. The foundations for solving Problem 1 are given in Sections 4.1–4.3, whereas the final solution is presented in Section 4.4. In Section 4.5, we extend our solution to Problem 1 with collision checking, such that it is capable of solving Problem 2 when coupled with a global motion-planning algorithm (e.g., sampling-based planner). Similarly, in Section 4.6 we extended the collision-checking method to rectangular robots. In Section 4.6 we discuss applications of the proposed extend procedure, and show how it can be applied to solve Problem 3. Before fully explaining our approach, let us briefly survey current path-planning primitives in the next section.

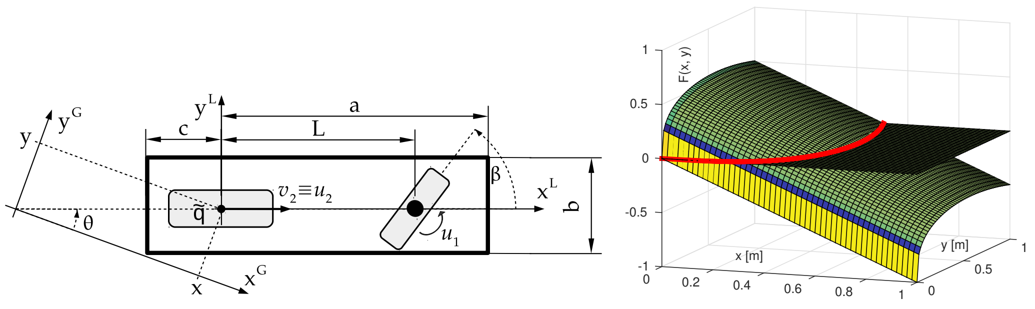

Figure 1. (**a**) A carlike robot with a rectangular footprint. (**b**) A level-curve representation of a transition path segment T, with path shown in red.

3. Related Work

Planning paths of limited curvature has been addressed with various methodologies. The most relevant properties of selected known approaches have been gathered in Table 1. A fundamental result from Reference [5] characterizes the shortest paths of bounded curvature as particular combinations of line segments and circle arcs, i.e., Reeds–Shepp (RS) paths. Note that this result does not consider obstacles, which have been partially taken into account for Dubins paths in Reference [6]. The drive toward paths with continuous curvature and a continuous-curvature arc-length derivative has led to the development of the CC-steer method [7] and its $\mathbb{G}^3$-continuous variant introduced in Reference [8]. Our work can also be viewed as a $\mathbb{G}^3$-continuous extension of the CC-steer method. However, contrary to the approach from Reference [8], our transition segments are not represented by curvature profiles. We propose transition segments with explicit representation of x as a function of y, which can easily be used with known path-following controllers (e.g., Reference [9]). We also devised a closed-form expression utilized to check for collisions between our $\mathbb{G}^3$-continuous path primitive and an obstacle, which leads to efficient and exact collision checking. There has also been some work on improving the computational cost of finding RS-like paths in Reference [10]; however, path-continuity and collision-checking issues for paths of high continuity were not explicitly addressed, to the best of our knowledge.

One can also find various methods of generating paths with a different structure to RS paths. For example, polynomial spline-based path primitives such as η^3-splines [11] and η^4-splines [12] have been developed. Their advantages are generality, conceptual simplicity, and high continuity. However, collision and curvature-constraint checking must usually be done numerically in the case of such primitives, which leads to significant computational cost. Furthermore, the parameters of such primitives can be hard to tune, even though this was partially addressed in Reference [13]. Another group of path primitives and extend procedures relies on B-splines [14–16] providing a limited curvature, curvature continuity (but not $\mathbb{G}^3$-continuity), and parameters that are easy to tune. Collision checking is done numerically in this case as well.

There are also various application-specific methods using the path primitives mentioned above, such as planning algorithms for environments with a roadlike structure similar to References [17,18], a method used to design curvature profiles of paths passing through waypoints [19], a spline-based approach exploiting sum-of-squares optimization to handle exact collision checking, or an elastic-band-like algorithm [20]. Those methods share the various benefits and drawbacks of different path primitives. However, thanks to the proposed path primitive, our approach combines $\mathbb{G}^3$-path continuity with analytically guaranteed curvature limits, as well as fast and exact analytic collision checking. To the best of our knowledge, such a combination of features is not exhibited by most known path primitives, as shown in Table 1.

Table 1. Comparison of the proposed $\mathbb{G}^3$-continuous path primitive with known path primitives.

Path Primitive	Continuity	Bounded κ	Collision Checking	Length Computation
Reeds–Shepp [5]	$\mathbb{G}^1$	yes	analytic	analytic
η^3-splines [11]	$\mathbb{G}^3$	no	numerical	numerical
η^4-splines [12]	$\mathbb{G}^4$	no	numerical	numerical
Clothoid-based	$\mathbb{G}^2$	yes	numerical	analytic
Fermat's spiral [21]	$\mathbb{G}^2$	yes	numerical	numerical
Low-order B-Splines [14,15]	$\mathbb{G}^2$	yes	numerical	numerical
Cubic curvature splines [8]	$\mathbb{G}^3$	yes	numerical	numerical
High-order B-splines	$\mathbb{G}^{3+}$	no	approx. analytic	numerical
HC-Steer [10]	$\mathbb{G}^1$	yes	analytic	numerical
$\mathbb{G}^3$-continuous path primitive	$\mathbb{G}^3$	**yes**	**analytic**	**numerical**

4. The $\mathbb{G}^3$-Continuous Extend Procedure

4.1. Main Concept

We introduce the so called $\mathbb{G}^3$-continuous path primitive, which consists of a transition segment (further explained in Section 4.2), a circle arc, a reversed transition segment, and a line segment. Similarly to the form of path encoding taken from Reference [5], we introduce a language for encoding paths with three words corresponding to path segments:

- $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$ denotes a transition segment connecting $\boldsymbol{w}_1$ with $\boldsymbol{w}_2$ (defined in Section 4.2),
- $C(\boldsymbol{w}_1, \boldsymbol{w}_2)$ is a circular arc of radius $1/\kappa_c$ connecting $\boldsymbol{w}_1$ with $\boldsymbol{w}_2$,
- $S(\boldsymbol{w}_1, \boldsymbol{w}_2)$ corresponds to a straight line connecting $\boldsymbol{w}_1$ with $\boldsymbol{w}_2$,

where $\boldsymbol{w}_k \triangleq [w_{k\theta}\ w_{kx}\ w_{ky}]^\top = [w_{k\theta}\ \boldsymbol{w}_k^\top]^\top$ for $k = 1, 2, 3, 4, 5$ correspond to the reference vehicle-body configurations at endpoints of path segments. Using this encoding, one can describe the proposed $\mathbb{G}^3$-continuous path primitive connecting $\boldsymbol{w}_1$ with $\boldsymbol{w}_5$ as

$$T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu_1)C(\boldsymbol{w}_2, \boldsymbol{w}_3)T(\boldsymbol{w}_4, \boldsymbol{w}_3, \mu_2)S(\boldsymbol{w}_4, \boldsymbol{w}_5).$$

The geometric interpretation of our $\mathbb{G}^3$-continuous path primitive is shown in Figure 2 (please, note the intentionally reversed order of the arguments in $T(\boldsymbol{w}_4, \boldsymbol{w}_3, \mu_2)$ in the above formula; it simply corresponds to a reversal of the segment's endpoints). Note that, by taking transition segments of zero length, one can obtain an RS path. Depending on a choice of parameters μ_1, μ_2, one can compromise between path length and smoothness resulting from longer transition segments. Since properties of RS paths are well known and problems such as collision checking or curvature limit checking are trivial in their case, we focused on transition segments in the sequel, and show how solutions to Problems 1–3 can be obtained with their help.

The extend procedure utilizing the $\mathbb{G}^3$-continuous path primitive consists of the following main stages:

1. Choose parameters μ_1, μ_2, and curvature $\kappa_c \neq 0$.
2. Find a sequence of $\mathbb{G}^3$-continuous path primitives connecting two prescribed vehicle-body configurations.
3. Check for collisions.
4. Return computed path or a collision signal.

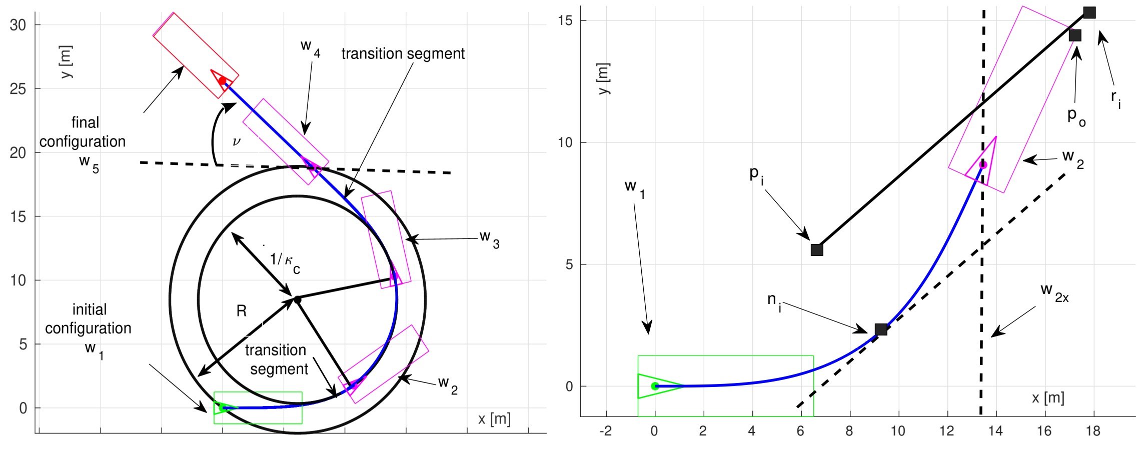

Figure 2. (**a**) Structure and parameters of the proposed $\mathbb{G}^3$-continuous path primitive. (**b**) Collision checking between transition segment and a line segment connecting $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$. Only points $\boldsymbol{p}_i$ and $\boldsymbol{n}_i$ have to be checked for collisions because point $\boldsymbol{r}_i$ lies outside the domain of interest.

4.2. Transition Segments and $\mathbb{G}^3$-Continuous Path Primitives

Since computing the endpoints of a line segment and a circle arc given its radius and length is straightforward, we explicitly only define relations concerning transition segments $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$. Let us denote variables expressed in a local coordinate frame fixed at point $\boldsymbol{w}_1$ by $(\cdot)^1$, that is, $\boldsymbol{w}_2^1$ corresponds to $\boldsymbol{w}_2$ expressed in coordinates of $\boldsymbol{w}_1$. A transition segment connecting vehicle-body configuration $\boldsymbol{w}_1$ with $\boldsymbol{w}_2$ is defined by the following curve, expressed in the coordinates of endpoint $\boldsymbol{w}_1$:

$$x^1 = f(y^1) = \begin{cases} \frac{-\operatorname{sgn}(w_{2x}^1)|y^1|}{2} \left[\left(\frac{y^1}{p}\right)^{\mu} - \left(\frac{y^1}{p}\right)^{-\mu} \right] & \text{for} \quad y^1 \neq 0, \\ 0 & \text{for} \quad y^1 = 0, \end{cases} \tag{7}$$

$$p \triangleq w_{2y}^1 \exp\left(\frac{\left| \operatorname{arsinh}\left(w_{2x}^1 / w_{2y}^1 \right) \right|}{\mu} \right), \quad w_{2y}^1 = \frac{K}{\kappa_c} y^*, \quad w_{2x}^1 = \frac{K}{\kappa_c} x^*, \tag{8}$$

with

$$K = \frac{y^* \left(-\mu + \mu^2 \frac{x^*}{r} \right)}{-(r)^2 \left(1 + \mu^2 - 2\mu \frac{x^*}{r} \right)^{3/2}}, \quad r \triangleq \sqrt{(x^*)^2 + (y^*)^2}, \quad x^* = f(y^*),$$

$$y^* = \left(\left(\frac{4}{3} p_5 - p_3 \right)^{1/3} - \frac{-6\mu^3 + \mu^2 + 2\mu + 3}{(6\mu + 3)(\mu + 1)^2} + p_2 \right)^{1/2\mu},$$

whereas

$$\begin{aligned}
p_1 &= -6\mu^4 - 5\mu^3 + 3\mu^2 + 5\mu + 3, \\
p_2 &= \frac{4\mu^2\left(18\mu^4 + 3\mu^3 - 17\mu^2 - 11\mu + 7\right)}{9p_1\left(\mu+1\right)^4\left(2\mu+1\right)^2}, \\
p_3 &= \frac{\left(p_1\right)^3}{27(2\mu+1)^3(\mu+1)^9} + \frac{(2\mu-1)(\mu-1)^3}{(4\mu+2)(\mu+1)^3} - p_4, \\
p_4 &= \frac{p_1(-6\mu^4 + 5\mu^3 + 3\mu^2 - 5\mu + 3)}{6(2\mu+1)^2(\mu+1)^6}, \\
p_5 &= \sqrt{\frac{\mu^6(\mu-1)^3(-36\mu^4 + 33\mu^2 - 29)}{(2\mu+1)^4(\mu+1)^9}},
\end{aligned}$$

where y^* is a rational function of $\mu \in (0.5, 1)$, $\kappa_c \in [-\kappa_B, \kappa_B] \setminus \{0\}$ denotes the curvature of adjacent circle arc C, while $\mu \in (0.5, 1)$ is a design parameter influencing the supremum value of curvature arc-length derivative $|d\kappa_d(s)/ds|$ during the transition segment and its length, as shown in Figures 3 and 4.

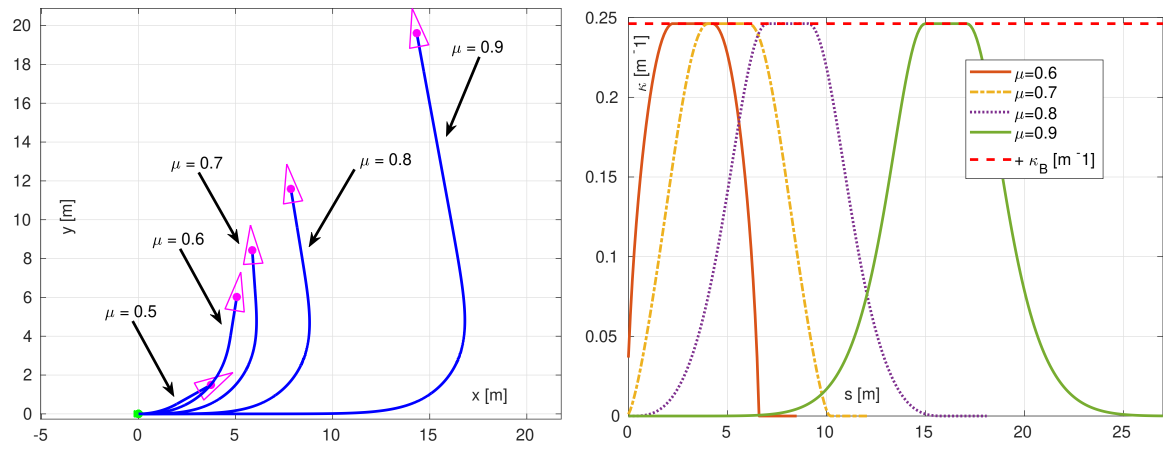

Figure 3. (**a**) $\mathbb{G}^3$-continuous path primitives for selected values of parameter μ. (**b**) Curvature profiles of selected $\mathbb{G}^3$-continuous path primitives.

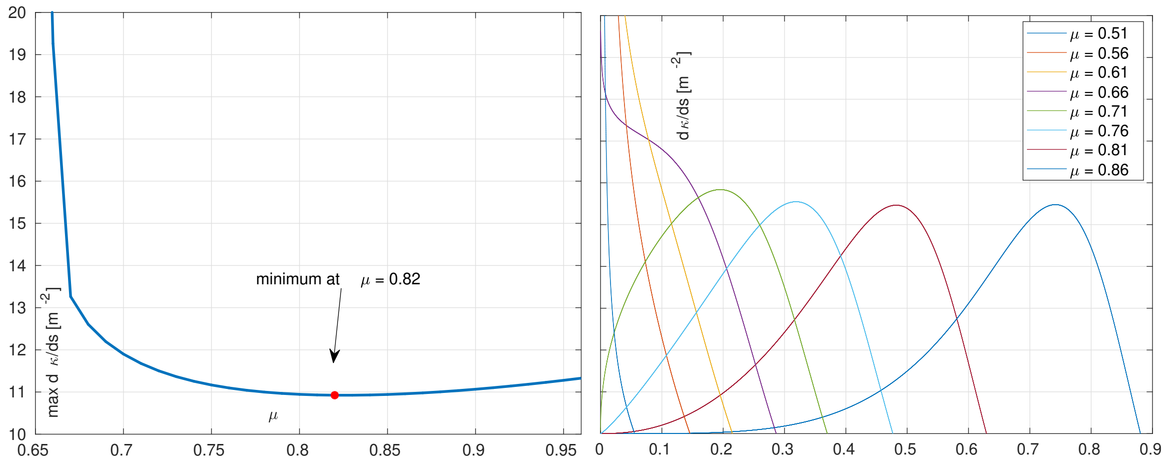

Figure 4. (**a**) Maximal arc-length derivative of transition segment curvature $\kappa_d(s)$ as a function of parameter μ. (**b**) Evolution of transition segment curvature arc-length derivative for selected values of μ.

Given a particular value of μ and transition segment endpoint $\boldsymbol{w}_1$, one can compute coordinates of the other endpoint $\boldsymbol{w}_2$ from Equation (8). The curve representing a transition segment is given by Curve (7) with particular value of parameter p resulting from Equation (8). Curve (7) was derived and analyzed in Reference [22]. It corresponds to an integral curve of the convergence vector field from the Vector Field Orientation (VFO) control law for the waypoint-following task. Its beneficial properties, proved in Reference [22], are instrumental in the construction and analysis of the $\mathbb{G}^3$-continuous path primitive. Given such a $\mathbb{G}^3$-continuous path primitive structure, we analyze its properties in the sequel.

Remark 1. *Transition segments are not explicitly parameterized by arc-length s, but they can immediately be used with path-following controllers utilizing level curves, such as the one from Reference [9].*

4.3. Path Continuity and Curvature Limit Satisfaction Analysis

Let us begin by showing that Curvature Limit (5) is satisfied. Since $\kappa_c \in [-\kappa_B, \kappa_B] \setminus \{0\}$, circle arcs and line segments satisfy the path curvature limit by construction. The curvature limit is also satisfied by both transition segments, since their curvature is limited by κ_c, as proven in Property 2 in Reference [22].

We now turn to path continuity. To ensure $\mathbb{G}^3$-continuity of the proposed primitive, one must guarantee that, for every transition segment T, the following relations hold:

$$\lim_{s \to s_1} \kappa_d(s) = 0 \quad \lim_{s \to s_1} \frac{d\kappa_d}{ds}(s) = 0, \quad \kappa_d(s_2) = \kappa_c, \quad \frac{d\kappa_d}{ds}(s_2) = 0,$$

where s_1 and s_2 are the values of s, such that $\bar{\boldsymbol{q}}_d(s_1) = \boldsymbol{w}_1$ and $\bar{\boldsymbol{q}}_d(s_2) = \boldsymbol{w}_2$. Note that a limit is sufficient to ensure $\mathbb{G}^3$-continuity in the case of point s_1, since at this point only a connection with line segments or boundary conditions of the path can occur, which guarantees continuity on the other side of the transition segment connection.

Condition $\kappa_d(s_2) = \kappa_c$ is immediately satisfied since point w^1_{2y} is defined in such a way, that it corresponds to the point of maximal curvature for the transition segment (assuming $\mu \in (0.5, 1)$, which is satisfied in our case) as shown in the proof of Property 2 in Reference [22]. Since a curvature maximum occurs at s_2, it also implies that $\frac{d\kappa_d}{ds}(s_2) = 0$ holds, because s_2 is a stationary point of $\kappa_d(s)$. In the proof of Property 2 from Reference [22], we have also shown that $\lim_{y^1 \to 0} \kappa_f(y^1) = 0$ for $\mu \in (0.5, 1)$, where κ_f denotes curvature of the transition segment as a function of y^1. We conclude that this implies $\lim_{s \to s_1} \kappa(s) = 0$.

Therefore, it remains to show that $\lim_{s \to s_1} \frac{d\kappa_d}{ds}(s) = 0$. Let us recall that, by the definition of curve's curvature, a transition segment curve T has a curvature that can be represented as

$$\kappa_f(y^1) = \frac{\frac{d^2 f^1}{d(y^1)^2}}{\left(1 + \left(\frac{df}{dy^1}\right)^2\right)^{3/2}}. \tag{9}$$

One can also compute the arc-length derivative of curvature $\frac{d\kappa_f}{ds}$ as follows:

$$\frac{d\kappa_f}{ds} = \frac{d\kappa_f(y^1)}{dy^1}\frac{dy^1}{ds} = \frac{d\kappa_f(y^1)}{dy^1} sin\theta_d(s) = \frac{d\kappa_f(y^1)}{dy^1} \frac{m \operatorname{sgn}(\mu)\, y^1}{\sqrt{f(y^1)^2 + (y^1)^2}}, \tag{10}$$

where $m = \text{const} > 0$, whereas the second equality results from the consideration of Equation (1) expressed in terms of parameter s (i.e., $t = s$), and the last equality results from the definition of $\theta_d(s)$ tangent to the transition segment T, which can be found from the definition of the VFO convergence vector field as shown in, e.g., Reference [22]. Then, differentiation of Equation (9) with respect to s and substitution of the result into the final form of Equation (10) results in a relation, which, after some tedious algebra, can be written as follows

$$\begin{aligned}\frac{d\kappa_f}{ds} &= \frac{8\,\mu\,\mathrm{sgn}\,(w^1_{2x})\,(1-5\mu+(y^1)^{2\mu}\,h_1+(y^1)^{4\mu}\,h2+(y^1)^{6\mu}\,h_3)}{((y^1)^{1-2\mu}((y^1)^{2\mu}+1)^{7/2}(1-\mu^2)^5)},\\ h_1 &= -6\mu^4+6\mu^3+3\mu^2-5\mu+3,\\ h_2 &= -6\mu^4-5\mu^3+3\mu^2+5\mu+3,\\ h_3 &= 2\mu^4+7\mu^3+9\mu^2+5\mu+1.\end{aligned}$$

This clearly implies that, for $\mu > 0.5$, limit $\lim_{y^1\to 0}\frac{d\kappa}{ds}(s) = 0$. As a consequence, $\lim_{s\to s_1}\frac{d\kappa}{ds}(s) = 0$, which concludes our analysis and proves that the proposed primitive is $\mathbb{G}^3$-continuous.

4.4. Computing Reeds–Shepp-Like Paths Using a $\mathbb{G}^3$-Continuous Path Primitive

In our extend procedure, we assume that parameters μ_1, μ_2, and κ_c are selected or randomly sampled by the global planning algorithm. For simplicity of considerations, we also assume $\mu_1 = \mu_2$, but the proposed procedure can be trivially adapted for the case of $\mu_1 \neq \mu_2$. Finding parameters of two $\mathbb{G}^3$-continuous primitives $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu_1)C(\boldsymbol{w}_2, \boldsymbol{w}_3)T(\boldsymbol{w}_4, \boldsymbol{w}_3, \mu_2)S(\boldsymbol{w}_4, \boldsymbol{w}_5)$ and $T(\boldsymbol{w}_5, \boldsymbol{w}_6, \mu_3)C(\boldsymbol{w}_6, \boldsymbol{w}_7)T(\boldsymbol{w}_8, \boldsymbol{w}_7, \mu_4)S(\boldsymbol{w}_8, \boldsymbol{w}_9)$ connecting prescribed vehicle-body configurations $\boldsymbol{w}_1$ and $\boldsymbol{w}_5$ is a nontrivial task. However, one can leverage extensions of the procedure devised by Reeds and Shepp, which were presented in Reference [7] and later Reference [8]. This path-construction procedure requires knowledge of auxiliary circles, on which endpoints $\boldsymbol{w}_4$ and $\boldsymbol{w}_8$ of the transition segments must lie. One must also know the difference between an orientation tangent to this circle and orientation tangent to the transition segment at endpoints $\boldsymbol{w}_4$ and $\boldsymbol{w}_8$. This difference is denoted by ν in Figure 2.

The reference path is computed during an extend procedure as follows:

1. The four possible transition segments T_1, T_2, T_3, T_4 starting at the prescribed initial vehicle body configuration are computed (Curve (7)). They correspond to forward motion with curvature κ_c, forward motion with curvature $-\kappa_c$, backward motion with curvature κ_c, and backward motion with curvature $-\kappa_c$. See Figure 5 for visual interpretation.
2. Step 1 is repeated for the four possible transition segments, T_5, T_6, T_7, T_8, ending at a prescribed final vehicle-body configuration.
3. For every transition segment $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$ computed up to this step, find center $\tilde{\boldsymbol{q}}_c = [x_c\, y_c]^\top$ of the auxiliary circle, on which the next transition segment must lie according to a simple geometric formula:
$$\tilde{\boldsymbol{q}}_c = [w_{2x}\, w_{2y}]^\top + 1/\kappa_c[-\sin w_{2\theta}\, \cos w_{2\theta}]^\top.$$
4. For every transition segment $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$ computed up to this step, find auxiliary circle radius R as follows:
$$R = \left\| \tilde{\boldsymbol{q}}_c - [w_{1x}\, w_{1y}]^\top \right\|.$$
5. For every transition segment $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$ without a fixed value of $\boldsymbol{w}_1$, find ν, which is straightforward given the knowledge of transition segment Curve (7) and the auxiliary circle.
6. For every circle segment $C(\boldsymbol{w}_1, \boldsymbol{w}_2)$, compute its remaining unknown endpoint using the algorithm from Reference [7].
7. If motion cost J is defined, compute the cost for all the paths and choose the optimal path. Otherwise, return a random path, or all found paths (depending on the utilized global planning algorithm).

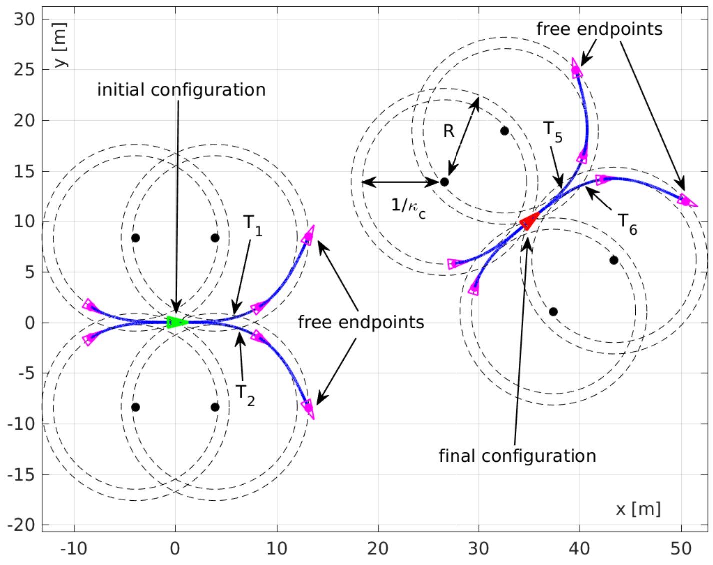

Figure 5. Visualization of the path computation procedure. Free endpoint positions must lie on auxiliary circles of radius R. Such circle arc lengths can be computed that allow for the connection of two free endpoints (i.e., endpoints of paths connected to the initial and final configuration, respectively) by a line segment without discontinuity in the reference orientation.

Note that the choice of μ should depend on the assumed motion costs. For example, if smooth paths are desired, then our computational studies summarized in Figure 4 show that choosing $\mu = 0.82$ leads to the paths with minimal $|d\kappa_d(s)/ds|$. On the other hand, for the shortest paths, choose μ close to 0.5. The proposed extend procedure solves Problem 1; however, to solve the other problems one must account for obstacles. This is solved in the next sections.

4.5. Satisfaction of State Constraints for Point Robots

Thanks to Assumption A4, set $\mathcal{P}_f$ can be described by all the obstacle edges and environment boundary edges (see Figure 2) gathered in the set of N line segments:

$$\mathcal{E} \triangleq \{(\boldsymbol{p}_i, \boldsymbol{r}_i)\}_{i=0}^{N}, \quad \boldsymbol{p}_i = [p_{ix}\, p_{iy}]^\top, \quad \boldsymbol{r}_i = [r_{ix}\, r_{iy}]^\top, \tag{11}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$ are endpoints of an i-th line segment. Condition (2) from Problem 2 is satisfied if no line segments from set $\mathcal{E}$ are crossed by the reference path. It is straightforward to analytically check this condition for circle arcs C and line segments S. We now show how to check this condition analytically for a transition segment $T(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$ with $w_{2x}^1 > 0$, since the proposed approach is easy to generalize for other cases.

After expressing edge $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$ in the coordinates of a transition segment endpoint $\boldsymbol{w}_1$, one concludes that a transition segment does not collide with line segment $(\boldsymbol{p}_i, \boldsymbol{r}_i)$ if

$$\forall k \in [0,1] \text{ and } d_x^1 \text{ such that } d_x^1 \in [0, w_{2x}^1] \quad \operatorname{sgn}\left(f(d_y^1) - d_x^1\right) = \text{const}, \tag{12}$$

where

$$\boldsymbol{d} = [d_x\, d_y]^\top = k\boldsymbol{p}_i + (1-k)\boldsymbol{r}_i.$$

This condition is illustrated in Figure 2. To explain, we consider curve f as a function and conclude that all points from the i-th line segment must lie either entirely above or below its graph. However, since the transition segment is bounded by its endpoints $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, we only consider such points from the i-th line segment that lie in subdomain $[0, w_{2x}^1]$ of curve f corresponding to the transition segment.

Condition (12) can be checked for the whole i-th line segment by simply checking it just for the maximal and minimal value of d_x^1 satisfying the left-hand-side conditions from Equation (12) and an additional critical point $\boldsymbol{n}_i \triangleq [n_{ix}\, n_{iy}]^\top$ computed according to Equation (42) from Reference [22]. At critical point $\boldsymbol{n}_i$, the orientation tangent to curve f is also tangential to the i-th line segment; thus, $\boldsymbol{n}_i$ is the point closest to or farthest from a line containing the i-th line segment. Note that, contrary to the approach from Reference [22], we check point $\boldsymbol{n}_i$ only if $\exists k \in [0, 1]$, such that $d_x = n_{ix}$.

4.6. Satisfaction of State Constraints for Robots with Rectangular Footprint

Following Reference [15], we only perform collision checking for the key points of a rectangular robot footprint. Namely, it was proven in Reference [15] that, apart from initial and final configuration, it is sufficient to check for the clearance of $0.5b$ m (see Figure 1) around the path and check for collisions of point $\boldsymbol{p}_o$ on the robot footprint (see Figure 2). The collision-checking procedure is performed as follows:

1. Using a simple algebra check if all footprint edges in the initial and final vehicle-body configurations are collision free.
2. Inflate the obstacles by $0.5b$ m. Perform the collision checking using our proposed fast method verifying condition (12).
3. Upon the instantaneous center of rotation compute the orientation θ_o tangent to instantaneous velocity of the point $\boldsymbol{p}_o$ (see Figure 2). Check for collisions of circle arcs and transition segments connected to a vehicle-body configuration $[\theta_o\, \boldsymbol{p}_o^\top]^\top$ with modified curvature $\kappa_a = \kappa_o(\kappa_c)$ instead of κ_c, where
$$\kappa_o(\kappa) = 1/\sqrt{(1/\kappa + b/2)^2 + L^2}, \tag{13}$$
where κ_o is the motion curvature of the point $\boldsymbol{p}_o$ which is furthest from the path, whereas κ is the curvature of robot motion.

As shown in Figure 6, the approach taken in Step 3 is conservative for the transition segments because by taking κ_a for collision checking, one assumes the motion curvature of point $\boldsymbol{p}_0$ to change linearly with respect to the curvature of the robot motion. However, it is also computationally efficient, and, as shown by our computational examples, its conservativeness does not hinder maneuverability of the robot in tight environments due to the ability to plan short transition segments.

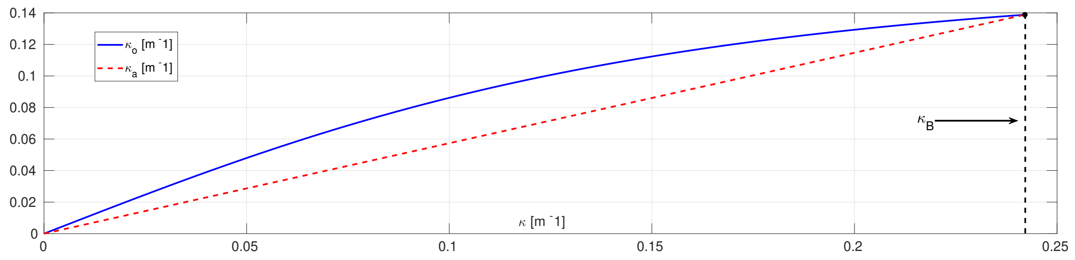

Figure 6. Curvature κ_o of motion for the outer vehicle point $\boldsymbol{p}_0$ (see Equation (13)), which is furthest from the path, expressed as a function of path curvature κ_d for $a = 5.9$ m and $b = 2.5$ m. Dashed line corresponds to a conservative inner approximation of this relation utilized during collision checking.

One can reason about the completeness of the presented extend procedure as follows. The proposed $\mathbb{G}^3$-continuous extend procedure leads to the solution of Problem 3 for its application to every complete global planning algorithm if there exists a collision-free RS path with ϵ-clearance for $\epsilon > 0$ solving this problem without the constraint of $\mathbb{G}^3$-continuity of the reference path. To illustrate why this is the case, let us consider that Equation (13) is not conservative for $\kappa(s) = 0$ and $\kappa(s) = \kappa_B$, that is, $\kappa_o = \kappa_a = 0$ and $\kappa_o = \kappa_a = \kappa_B$, respectively. Furthermore, due to Property 2 from Reference [22] and continuity of y^*, one concludes that $y^* \to 0$ as $\mu \to 0.5$. This means that, as $\mu \to 0.5$, the length of the transition segments tends to 0, and paths obtained from the concatenation of our $\mathbb{G}^3$-continuous path primitives approximate RS paths arbitrarily closely. Therefore, if there exists a feasible RS path that is no closer to obstacles than ϵ, one can always find such μ sufficiently close to 0.5, so that the maximal distance between the RS path and corresponding $\mathbb{G}^3$-continuous path is less than ϵ, meaning that the $\mathbb{G}^3$-continuous path is feasible.

5. Computational Results

To verify feasibility and effectiveness of planning with the proposed $\mathbb{G}^3$-continuous path primitive, it was investigated how it impacts computation times for the most crucial primitive operations in path planning, such as collision checking, checking of curvature-constraint satisfaction (κ checking), and extension procedure computation. Similarly, we tested the computational performance of our approach in path-planning scenarios S1–S3, shown in Figures 7–9. The proposed extend procedure was integrated with the RRT* motion-planning algorithm (see Reference [23] for details). All simulations were performed with the following parameters: $L = 5.7$ m, $b = 2.5$ m, $a = 5.9$ m, $\beta_m = 0.96$ rad. We used motion cost $J \triangleq \bar{l} + \int_0^{s_{fin}} \left(\frac{d\kappa}{ds}(s)\right)^2 ds$, where $\bar{l}$ corresponds to the total path length. Note that the planning procedure was finished when the obtained motion cost was within 5% of the value precomputed over the time of 600 s.

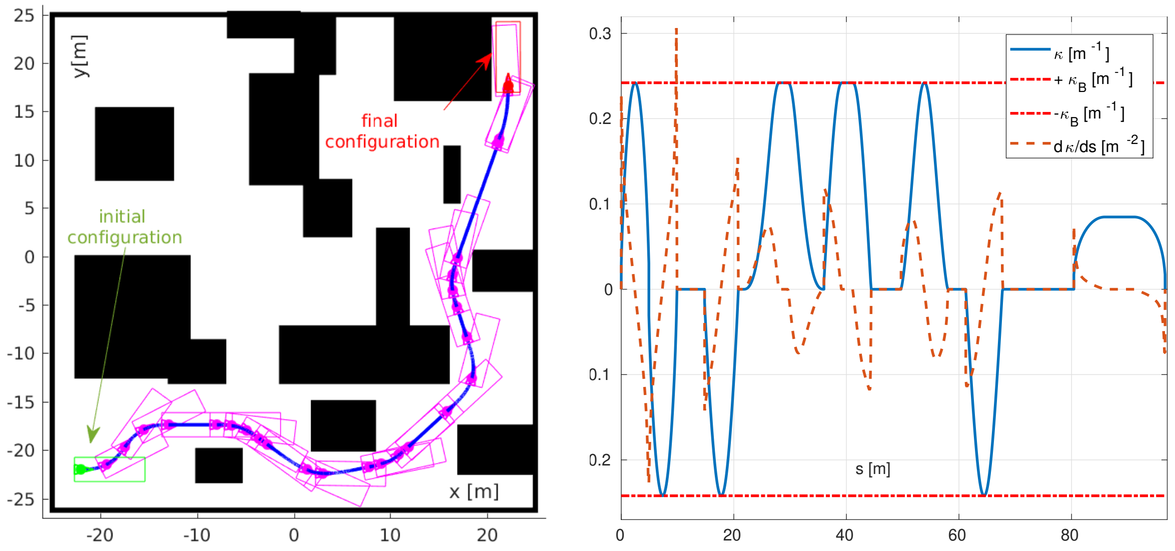

Figure 7. Planned path and curvature profile for forward parking Scenario S1. Only forward robot motion was allowed.

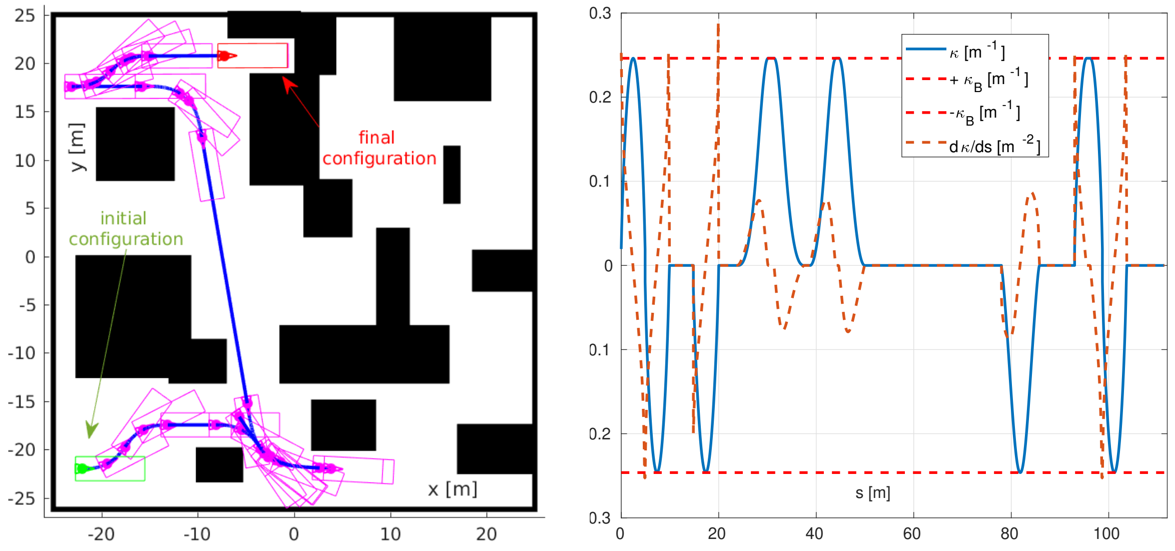

Figure 8. Planned path and curvature profile for parking Scenario S2. Both forward and backward robot motion were allowed.

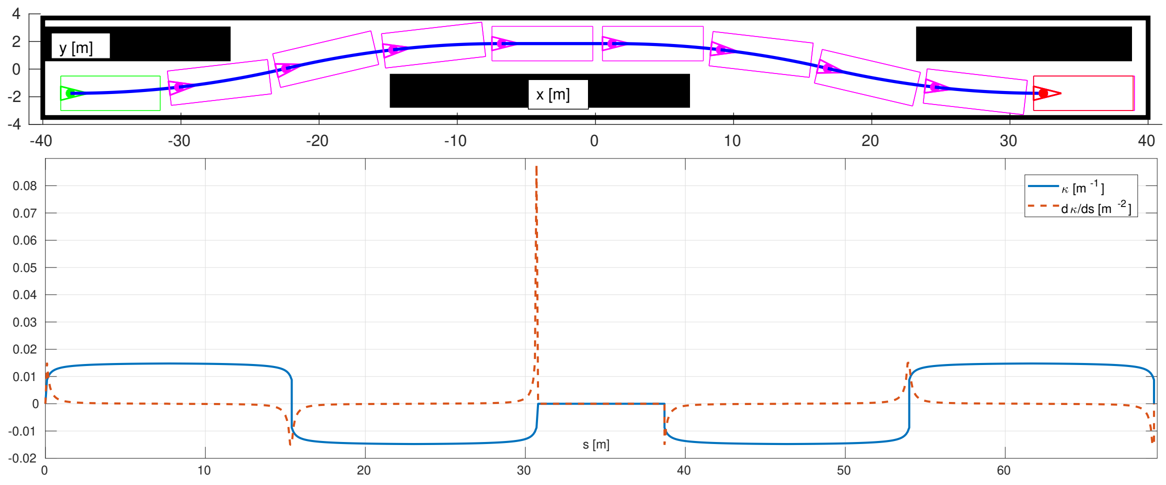

Figure 9. Planned path and curvature profile for Scenario S3 corresponding to a lane-change-like maneuver.

The computation times obtained in MATLAB are presented in Tables 2 and 3. One can observe that the $\mathbb{G}^3$ path continuity of our approach comes with the price of increased computational cost in comparison to the classic RS paths with curvature discontinuities. This was expected, since one has to specifically account for additional transition segments during collision checking, and also check a larger amount of paths due to availability of additional combinations of transition segments and circle arcs. However, we note that the computational cost for the more general η^3-splines, which are also $\mathbb{G}^3$-continuous, is significantly higher than in our approach. This observation holds even if one assumes that numerical checking of curvature constraints and collision checking can be performed in parallel. Unsurprisingly, since collision-checking procedures can account for over 90% of planning time, those computational performance tendencies propagate to computation times of a full path-planning procedure for example Scenario S2. Such results suggest that our approach can represent a viable alternative when the planning of $\mathbb{G}^3$-continuous paths is necessary due to task-specific constraints or the mechanical construction of the robot.

Table 2. Average computation times of primitive operations for 1000 random scenarios.

Path Primitive	Collision Checking (μs)	κ Checking (μs)	Extend Procedure (μs)
Reeds–Shepp	24	0	27
η^3-splines	1397	1264	63
$\mathbb{G}^3$-continuous path primitive	124	0	67

Table 3. Average computation times in MATLAB for 10 planning trials in Scenario S2.

Path Primitive	Planning Time of S2 (S)
Reeds-Shepp	27
η^3-splines	86
$\mathbb{G}^3$-continuous path primitive	39

Figures 7 and 8 illustrate the results of path planning with the proposed $\mathbb{G}^3$-continuous extend procedure for parking Scenarios S1 and S2, whereas in Figure 9 we show a challenging lane-change-like maneuver in Scenario S3. During Scenario S3, only forward robot motion was assumed admissible. Obstacles are shown in black, whereas the planned path is in blue. The magenta rectangles correspond to robot footprints at the endpoints of path segments comprising $\mathbb{G}^3$-continuous path primitives, whereas the green and red rectangles illustrate the initial and final robot configuration, respectively. It can be seen that planning is successfully performed in a severely constrained environment despite the conservative approximation utilized in our collision-checking algorithm. Curvature profiles and curvature arc-length derivative profiles are continuous; however, in some cases, e.g., in Scenario S3, relatively low μ values corresponding to relatively high curvature arc-length derivative values have been planned. This is due to the environment boundaries, which prohibited smoother curvature profiles, since those would lead to collision between the environment boundaries and the robot footprint, specifically point $\boldsymbol{p}_0$ of the footprint. One can also find that, in some cases, paths contain more reversals (changes in motion strategy), because additional space is needed for transition segments, which allow smooth evolution of path curvature. Such a tendency can be eliminated by putting a bigger emphasis on path length in the planning motion cost; however, this inevitably leads to low μ values and less smooth paths.

6. Conclusions

The $\mathbb{G}^3$-continuous path primitive proposed in this paper allows for an extension of the well-known RS paths, and constitutes an easy-to-implement component for various path planners available in the literature. The proposed method guarantees that planned paths are collision-free, satisfy curvature constraints, and preserve continuity of the curvature arc-length derivative. It is worth emphasizing the computational efficiency of the method due to the fact that distance between the robot with a rectangular footprint and obstacles can be effectively checked in a continuous domain using the derived analytical formulas. As opposed to other solutions (e.g., those using clothoid-based approaches), the introduced $\mathbb{G}^3$-continuous path primitives are represented by closed-form expressions, which can be conveniently utilized by path-following feedback controllers. Upon the results included in the paper, one may conclude that the proposed planning strategy provides all the mentioned beneficial properties under a reasonable computational cost when compared to other methods known from the literature.

Author Contributions: Conceptualization, T.G. and M.M.M.; methodology, T.G. and M.M.M.; software, T.G.; validation, T.G. and M.M.M.; writing—original draft preparation, T.G.; writing—review and editing, M.M.M.; visualization, T.G.; supervision, M.M.M.

Funding: This work was financially supported by the National Science Center, Poland via research grant No. 2016/21/B/ST7/02259.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
2. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [CrossRef]
3. Gawron, T.; Michałek, M.M. Planning $\mathbb{G}^3$-continuous paths for state-constrained mobile robots with bounded curvature of motion. In *Trends in Advanced Intelligent Control, Optimization and Automation*; Mitkowski, W., Kacprzyk, J., Oprzędkiewicz, K., Skruch, P., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 473–482.
4. Michałek, M.; Kozłowski, K. Feedback control framework for car-like robots using the unicycle controllers. *Robotica* **2012**, *30*, 517–535. [CrossRef]
5. Reeds, J.; Shepp, R. Optimal paths for a car that goes both forward and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [CrossRef]
6. Agarwal, P.K.; Biedl, T.; Lazard, S.; Robbins, S.; Suri, S.; Whitesides, S. Curvature-Constrained Shortest Paths in a Convex Polygon. *SIAM J. Comput.* **2002**, *31*, 1814–1851. [CrossRef]
7. Fraichard, T.; Scheuer, A. From Reeds and Shepp's to continuous-curvature paths. *IEEE Trans. Robot.* **2004**, *20*, 1025–1035. [CrossRef]
8. Oliveira, R.; Lima1, P.F.; Cirillo, M.; Martensson, J.; Wahlberg, B. Trajectory Generation using Sharpness Continuous Dubins-like Paths with Applications in Control of Heavy-Duty Vehicles. In Proceedings of the European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018.

9. Michałek, M.M.; Gawron, T. VFO Path following Control with Guarantees of Positionally Constrained Transients for Unicycle-Like Robots with Constrained Control Input. *J. Intell. Robot. Syst.* **2018**, *89*, 191–210. [CrossRef]
10. Banzhaf, H.; Palmieri, L.; Nienhüser, D.; Schamm, T.; Knoop, S.; Zöllner, J.M. Hybrid curvature steer: A novel extend function for sampling-based nonholonomic motion planning in tight environments. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems, Yokohama, Japan, 16–19 October 2017; pp. 1–8.
11. Lini, G.; Piazzi, A.; Consolini, L. Multi-optimization of η^3-splines for autonomous parking. In Proceedings of the Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, 12–15 December 2011; pp. 6367–6372.
12. Ghilardelli, F.; Lini, G.; Piazzi, A. Path Generation Using η^4-Splines for a Truck and Trailer Vehicle. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 187–203. [CrossRef]
13. Bianco, C.G.L.; Gerelli, O. Generation of Paths with Minimum Curvature Derivative with η^3-Splines. *IEEE Trans. Autom. Sci. Eng.* **2010**, *7*, 249–256. [CrossRef]
14. Elbanhawi, M.; Simic, M.; Jazar, R. Continuous Path Smoothing for Car-Like Robots Using B-Spline Curves. *J. Intell. Robot. Syst.* **2015**, *80*, 23–56. [CrossRef]
15. Yoon, S.; Lee, D.; Jung, J.; Shim, D.H. Spline-based RRT* Using Piecewise Continuous Collision-checking Algorithm for Car-like Vehicles. *J. Intell. Robot. Syst.* **2018**, *90*, 537–549. [CrossRef]
16. Yang, K.; Moon, S.; Yoo, S.; Kang, J.; Doh, N.; Kim, H.; Joo, S. Spline-Based RRT Path Planner for Non-Holonomic Robots. *J. Intell. Robot. Syst.* **2014**, *73*, 763–782. [CrossRef]
17. Bertolazzi, E.; Bevilacqua, P.; Biral, F.; Fontanelli, D.; Frego, M.; Palopoli, L. Efficient Re-planning for Robotic Cars. In Proceedings of the European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018.
18. Gim, S.; Adouane, L.; Lee, S.; Dérutin, J. Clothoids Composition Method for Smooth Path Generation of Car-Like Vehicle Navigation. *J. Intell. Robot. Syst.* **2017**, *88*, 129–146. [CrossRef]
19. 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 17th Mediterranean Conference Control and Automation, Makedonia Palace, Thessaloniki, Greece, 24–26 June 2009; pp. 73–78.
20. Ahmadzadeh, A.; Jadbabaie, A.; Pappas, G.J.; Kumar, V. Elastic multi-particle systems for bounded-curvature path planning. In Proceedings of the 17th American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 5035–5040.
21. Candeloro, M.; Lekkas, A.M.; Sørensen, A.J.; Fossen, T.I. Continuous Curvature Path Planning using Voronoi diagrams and Fermat's spirals. *IFAC Proc. Vol.* **2013**, *46*, 132–137. [CrossRef]
22. Gawron, T.; Michałek, M.M. The VFO-Driven Motion Planning and Feedback Control in Polygonal Worlds for a Unicycle with Bounded Curvature of Motion. *J. Intell. Robot. Sys.* **2018**, *89*, 265–297. [CrossRef]
23. Karaman, S.; Frazzoli, E. Sampling-based optimal motion planning for non-holonomic dynamical systems. In Proceedings of the IEEE International Conference Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5041–5047.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).