2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

From G^2 to G^3 Continuity: Continuous Curvature Rate Steering Functions for Sampling-Based Nonholonomic Motion Planning

Holger Banzhaf[1], Nijanthan Berinpanathan[2], Dennis Nienhüser[1], J. Marius Zöllner[3]

***Abstract*— Motion planning for car-like robots is one of the major challenges in automated driving. It requires to solve a two-point boundary value problem (BVP) in real time while taking into account the nonholonomic constraints of the vehicle and the obstacles in the non-convex environment. This paper introduces Hybrid Curvature Rate (HCR) and Continuous Curvature Rate (CCR) Steer: Two novel steering functions for car-like robots that compute a curvature rate continuous solution of the two-point BVP. Hard constraints on the maximum curvature, maximum curvature rate, and maximum curvature acceleration are satisfied resulting in directly driveable G^3 continuous paths. The presented steering functions are benchmarked in terms of computation time and path length against its G^1 and G^2 continuous counterparts, namely Dubins, Reeds-Shepp, Hybrid Curvature, and Continuous Curvature Steer. It is shown that curvature rate continuity can be achieved with only small computational overhead. The generic motion planner Bidirectional RRT* is finally used to present the effectiveness of HCR and CCR Steer in three challenging automated driving scenarios.**

I. Introduction

Planning feasible, collision-free motions is one of the major tasks of automated vehicles. Although the problem formulation of motion planning is well understood, finding (sub)optimal solutions in real time is still an actively researched topic [1]. The key challenge lies in solving a two-point boundary value problem (BVP) of a nonlinear system in a non-convex environment. Possible scenarios are maneuvering, parking, or inductive charging that require positioning the automated vehicle exactly at a goal configuration while avoiding obstacles in the environment. Within this context, many motion planners, e.g. Hybrid A* (goal sampling) [2] or RRT* [3], require a steering function that connects two vehicle configurations with a feasible path.

The two most prominent steering functions for car-like robots Dubins [4] and Reeds-Shepp (RS) [5] only take into account the kinematic constraints of the vehicle resulting in G^1 continuous paths with discontinuous curvature profiles. While increasing the smoothness of a path not only results in a better tracking performance of the vehicle controller [6], it also makes paths more comfortable to drive, and puts less mechanical stress on the vehicle. Therefore, the G^2 continuous steering functions Hybrid Curvature (HC) [7] and Continuous Curvature (CC) Steer [8] are proposed that compute a curvature continuous path by assuming infinite curvature acceleration. Since this assumption violates the actuator limits, G^3 continuity with bounds on the maximum curvature, maximum curvature rate, and maximum curvature acceleration is required to satisfy the dynamic constraints of the steering system.

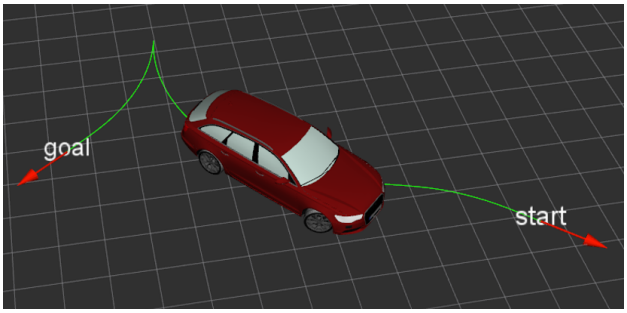

Fig. 1: Two vehicle configurations are connected with a G^3 continuous path that satisfies hard constraints on the maximum curvature, maximum curvature rate, and maximum curvature acceleration.

Within this regard, the main **contributions** of this paper are:

- Introduction and derivation of two novel curvature rate continuous steering functions called Hybrid Curvature Rate (HCR) and Continuous Curvature Rate (CCR) Steer. Hard constraints are enforced on the maximum curvature, maximum curvature rate, and maximum curvature acceleration. In contrast to CCR, HCR allows curvature discontinuties at direction switches.
- Benchmark of HCR and CCR Steer against the state of the art steering functions Dubins, RS, HC, and CC Steer with respect to computation time and path length. It is shown that curvature rate continuity is achieved with only small computational overhead.
- Integration of HCR and CCR Steer into the generic motion planner Bidirectional RRT* illustrating its effectiveness in three challenging automated driving scenarios.

The remainder of this paper is organized as follows. Section II describes the related work, and Section III derives HCR and CCR Turns as prerequisites for HCR and CCR Steer, which are introduced in Section IV. The experimental results are presented in Section V, and a conclusion and an outlook are given in Section VI.

II. Related Work

Three different approaches to design a steering function for car-like robots are described in this section. The first approach formulates an optimal control problem taking into account the constraints of the system [9], [10]. Numeric

[1]Holger Banzhaf and Dennis Nienhüser are with Robert Bosch GmbH, Corporate Research, Driver Assistance Systems and Automated Driving, Renningen, Germany.
[2]Nijanthan Berinpanathan is with ETH Zurich, Switzerland.
[3]J. Marius Zöllner is with FZI Research Center for Information Technology, Karlsruhe, Germany.

optimization is then used to compute an optimal solution. The entire complexity, such as the nonlinearity of the system, is therefore pushed to the optimizer resulting in problems such as high computation times for complex maneuvers or slow/no convergence.

The second approach constrains the possible solutions to a set of parametrized curves and optimizes the parameters according to a cost function [11], [12]. In order to guarantee fast convergence, the maneuver is commonly restricted to forwards driving only, and constraints on the maximum curvature or curvature rate are often relaxed or removed.

The third approach concatenates a set of geometric primitives, such as circular arcs and straight lines, in order to connect two configurations. The two most prominent steering functions in this field are Dubins [4] and Reeds-Shepp (RS) [5]. They compute the shortest, curvature-constrained path for a car that only moves forwards (Dubins) or forwards and backwards (RS). The computation of these two steering functions is not only computationally inexpensive, but they also guarantee to find a solution at all times. However, the major drawback is the G^1 continuity of the computed paths (curvature discontinuous). On the contrary, Hybrid Curvature (HC) [7] and Continuous Curvature (CC) Steer [8] compute curvature continuous G^2 paths by adding clothoids to the set of geometric primitives. Complex maneuvers can be computed for cars that go both forwards and backwards, however, infinite steering accelerations are assumed. As a consequence, significant deviations from the planned path may occur when being executed by a vehicle controller [13].

This effect can be reduced by G^3 continuous paths that enforce curvature and curvature rate continuity. Potential geometric curves in this context are cubic spirals [14], polynomial spirals as their generalization [15], or η^3 splines [16] to name a few. The key challenge lies in the concatenation of these curves to a G^3 continuous path for a car that goes both forwards and backwards. In contrast to [17] and [18], the presented approach in this paper does not only bound the maximum curvature and maximum curvature rate, but also the maximum curvature acceleration as required by the actuator limits. A similar idea has only recently been pursued in [13], where curvature rate continuity as well as the steering constraints are enforced for a vehicle that only moves forwards.

III. HCR and CCR Turns

In this section, HCR and CCR Turns are introduced as the basic components of HCR and CCR Steer. The overall goal is to derive a relation between turn length and turning angle, which is required for optimization in Section IV.

Both HCR and CCR Turns are based on the following vehicle model, which is given with respect to arc length s as

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \kappa' \\ \sigma' \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \kappa \\ \sigma \\ 0 \end{pmatrix} d + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rho, \tag{1a}$$

$$\kappa \in [-\kappa_{\max}, \kappa_{\max}], \tag{1b}$$

$$\sigma \in [-\sigma_{\max}, \sigma_{\max}], \tag{1c}$$

$$\rho \in [-\rho_{\max}, \rho_{\max}], \tag{1d}$$

where x, y describe the position of the rear axle's midpoint, θ denotes the heading angle, κ the curvature, σ the curvature rate, d the driving direction, and ρ the curvature acceleration. The configuration of the vehicle can be summarized to $\mathbf{q} = [x, y, \theta, \kappa, \sigma]^T$ and the input to $\mathbf{u} = [d, \rho]^T$. Derivatives with respect to arc length s are given by $(\bullet)'$ and the vehicle's kinematic and dynamic constraints by (1b-1d).

A. Hybrid Curvature Rate Turn

Figure 2 visualizes a HCR Turn including a HCR Circle C_t^d for a forwards movement ($d = 1$) to the left ($t = 1$). The turn begins in a start configuration $\mathbf{q}_s$ with $\kappa = \sigma = 0$, reaches the maximum curvature at the intermediate configuration $\mathbf{q}_i$, and terminates in a goal configuration $\mathbf{q}_g$ with $\kappa = \kappa_{\max}$ and $\sigma = 0$. The angular change between start and goal configuration is described by the deflection $\delta \in [0, 2\pi[$.

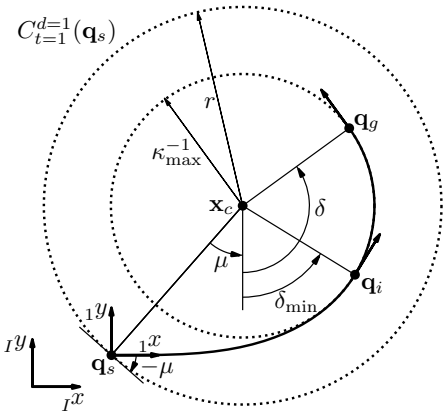

Fig. 2: Regular HCR Turn that starts with $\kappa = \sigma = 0$ in $\mathbf{q}_s$, passes the intermediate configuration $\mathbf{q}_i$, and ends with $\kappa = \kappa_{\max}$ and $\sigma = 0$ in $\mathbf{q}_g$.

The transition between the start configuration $\mathbf{q}_s$ and the intermediate configuration $\mathbf{q}_i$ is modeled with a set of cubic spirals in order to fulfill (1b-1d). In general, spirals are a class of curves that describe the orientation by a polynomial of degree n with respect to path length s according to

$$\theta(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0. \tag{2}$$

The curvature and the curvature rate of a spiral can be computed by

$$\kappa(s) = \frac{\mathrm{d}\theta(s)}{\mathrm{d}s}, \tag{3}$$

$$\sigma(s) = \frac{\mathrm{d}^2\theta(s)}{\mathrm{d}s^2}. \tag{4}$$

The Cartesian coordinates are obtained by

$$x(s) = \int_0^s \cos(\theta(\tau))\mathrm{d}\tau, \tag{5}$$

$$y(s) = \int_0^s \sin(\theta(\tau))\mathrm{d}\tau. \tag{6}$$

Clothoids, also known as Euler or Cornu spiral, are a special case of (2) ($n = 2$) that guarantee curvature continuity, but allow curvature rate discontinuities [7], [8]. In contrast to that, cubic spirals ($n = 3$) result in curvature and curvature rate continuous curves making them a suitable candidate for HCR Turns.

In order to allow the car to steer with its maximum curvature acceleration while satisfying the defined constraints, it is proposed to concatenate up to three cubic spirals to connect $\mathbf{q}_s$ with $\mathbf{q}_i$. Depending on the parameters $\kappa_{\max}$, $\sigma_{\max}$, $\rho_{\max}$, one of the two continuous curvature rate profiles illustrated in Figure 3 has to be selected.

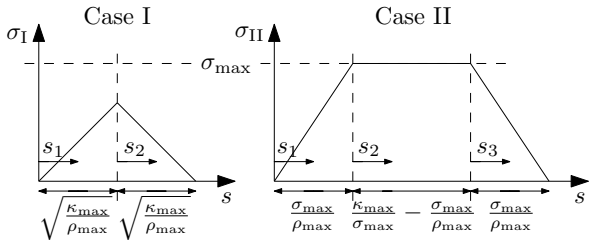

Fig. 3: Continuous curvature rate profiles that describe the concatenated cubic spirals between $\mathbf{q}_s$ and $\mathbf{q}_i$. If $\kappa_{\max} \leq \sigma_{\max}^2/\rho_{\max}$, the left profile has to be applied and the right one in all other cases.

Case I in Figure 3 consists of two cubic spirals and is applicable if $\kappa_{\max} \leq \sigma_{\max}^2/\rho_{\max}$. As it can be seen, this case applies if the maximum curvature can be reached without saturating the curvature rate. Otherwise, case II has to be applied, where three cubic spirals are concatenated. Taking into account that $\kappa = \sigma = 0$ at $\mathbf{q}_s$ and $\kappa = \kappa_{\max}$ and $\sigma = 0$ at $\mathbf{q}_i$, the curvature rate in case I can be computed as

$$\sigma_{\mathrm{I}}(s_1) = \rho_{\max} s_1, \qquad s_1 \in [0, \sqrt{\tfrac{\kappa_{\max}}{\rho_{\max}}}], \quad (7)$$

$$\sigma_{\mathrm{I}}(s_2) = -\rho_{\max} s_2 + \sqrt{\kappa_{\max}\rho_{\max}}, \quad s_2 \in [0, \sqrt{\tfrac{\kappa_{\max}}{\rho_{\max}}}], \quad (8)$$

and in case II as

$$\sigma_{\mathrm{II}}(s_1) = \rho_{\max} s_1, \qquad s_1 \in [0, \tfrac{\sigma_{\max}}{\rho_{\max}}], \quad (9)$$

$$\sigma_{\mathrm{II}}(s_2) = \sigma_{\max}, \qquad s_2 \in [0, \tfrac{\kappa_{\max}}{\sigma_{\max}} - \tfrac{\sigma_{\max}}{\rho_{\max}}], \quad (10)$$

$$\sigma_{\mathrm{II}}(s_3) = -\rho_{\max} s_3 + \sigma_{\max}, \quad s_3 \in [0, \tfrac{\sigma_{\max}}{\rho_{\max}}], \quad (11)$$

where s_1, s_2, and s_3 describe the arc length of the piecewise linear functions, see Figure 3.

The curvature profiles are obtained by integrating (7-11) according to

$$\kappa_{\mathrm{I}}(s_1) = \frac{\rho_{\max}}{2} s_1^2, \quad (12)$$

$$\kappa_{\mathrm{I}}(s_2) = -\frac{\rho_{\max}}{2} s_2^2 + \sqrt{\kappa_{\max}\rho_{\max}} s_2 + \frac{\kappa_{\max}}{2}, \quad (13)$$

and

$$\kappa_{\mathrm{II}}(s_1) = \frac{\rho_{\max}}{2} s_1^2, \quad (14)$$

$$\kappa_{\mathrm{II}}(s_2) = \sigma_{\max} s_2 + \frac{\sigma_{\max}^2}{2\rho_{\max}}, \quad (15)$$

$$\kappa_{\mathrm{II}}(s_3) = \frac{-\rho_{\max}}{2} s_3^2 + \sigma_{\max} s_3 + \kappa_{\max} - \frac{\sigma_{\max}^2}{2\rho_{\max}}. \quad (16)$$

Integrating (12-16) leads to $\theta_{\mathrm{I}}(s)$ and $\theta_{\mathrm{II}}(s)$ and allows to compute $\mathbf{q}_i$ in the local coordinate frame 1 (indicated by a left subscript, see Figure 2) as

$${}_1\mathbf{q}_i = \begin{pmatrix} x_i \\ y_i \\ \theta_i \\ \kappa_i \\ \sigma_i \end{pmatrix} = \begin{pmatrix} d\int_0^{l_{\min}} \cos(\theta(s))\mathrm{d}s \\ t\int_0^{l_{\min}} \sin(\theta(s))\mathrm{d}s \\ td\theta(l_{\min}) \\ t\kappa_{\max} \\ 0 \end{pmatrix}, \quad (17)$$

where $d \in \{-1, 1\}$ denotes the driving direction (backwards, forwards) and $t \in \{-1, 1\}$ the direction of a turn (right, left). The indices for case I and II are omitted for better readability. The length of the concatenated spirals $l_{\min}$ is given as

$$l_{\min} = \begin{cases} 2 \cdot \sqrt{\frac{\kappa_{\max}}{\rho_{\max}}}, & \text{if } \kappa_{\max} \leq \sigma_{\max}^2/\rho_{\max}, \\ \frac{\kappa_{\max}}{\sigma_{\max}} + \frac{\sigma_{\max}}{\rho_{\max}}, & \text{else.} \end{cases} \quad (18)$$

Since the integrals in (17) do not have a closed-form solution, they are evaluated using Gauss-Legendre quadrature. A coordinate transformation can then be applied to transform ${}_1\mathbf{q}_i$ from the local frame 1 to the inertial frame I given the start configuration ${}_I\mathbf{q}_s = (x_s, y_s, \theta_s, 0, 0)$.

The deflection $\delta_{\min}$ denotes the angular change between start and intermediate configuration and is given by

$$\delta_{\min} = \theta_i. \quad (19)$$

The center of the HCR Circle can be computed as

$${}_1\mathbf{x}_c = \begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x_i - \kappa_i^{-1}\sin(\theta_i) \\ y_i + \kappa_i^{-1}\cos(\theta_i) \end{pmatrix}. \quad (20)$$

Its radius is $r = \|{}_1\mathbf{x}_c\|_2$.

Rotating $\mathbf{q}_i$ on a circle with radius $1/\kappa_{\max}$ by $\delta - \delta_{\min}$ results in the goal configuration $\mathbf{q}_g$, which is given as

$${}_1\mathbf{q}_g = \begin{pmatrix} x_g \\ y_g \\ \theta_g \\ \kappa_g \\ \sigma_g \end{pmatrix} = \begin{pmatrix} x_c + \kappa_i^{-1}\sin(\theta_i + td(\delta - \delta_{\min})) \\ y_c - \kappa_i^{-1}\cos(\theta_i + td(\delta - \delta_{\min})) \\ \theta_i + td(\delta - \delta_{\min}) \\ \kappa_i \\ 0 \end{pmatrix}. \quad (21)$$

The tangent to the HCR Circle at the start configuration encloses the angle μ with the initial orientation. It can be calculated according to

$$\mu = td\arctan(x_c/y_c). \quad (22)$$

On the basis of the mathematical description in the previous paragraphs, the length of a HCR Turn can now be derived. In this respect, it is distinguished between a regular and an irregular HCR Turn. In contrast to a regular turn (see Figure 2), the irregular one allows a cusp (direction switch) at $\mathbf{q}_i$ resulting in a shorter path length for $\delta > \delta_{\min} + \pi$. Therefore, irregular turns should be used to minimize path length and regular ones to reduce the number of cusps.

1) Regular HCR Turn: The length of a regular HCR Turn, where $\mathbf{q}_s$ and $\mathbf{q}_g$ are connected without direction switches in between, is given as

$$l(\delta) = \begin{cases} l_{\min} + \kappa_{\max}^{-1}(2\pi + \delta - \delta_{\min}), & \text{if } \delta < \delta_{\min}, \\ l_{\min} + \kappa_{\max}^{-1}(\delta - \delta_{\min}), & \text{if } \delta \geq \delta_{\min}. \end{cases} \quad (23)$$

2) Irregular HCR Turn: Figure 4 visualizes an irregular HCR Turn, which allows a direction switch at $\mathbf{q}_i$. Its path

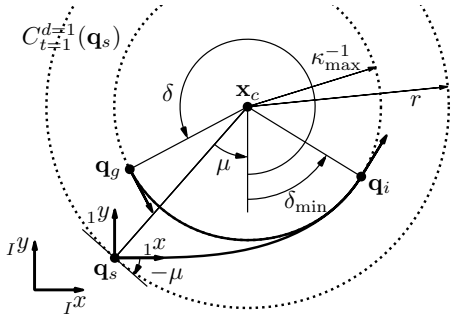

Fig. 4: Irregular HCR Turn that allows a direction switch at $\mathbf{q}_i$.

length can be derived as

$$l(\delta) = \begin{cases} l_{\min} + \kappa_{\max}^{-1}(-\delta + \delta_{\min}), & \text{if } \delta < \delta_{\min}, \\ l_{\min} + \kappa_{\max}^{-1}(2\pi - \delta + \delta_{\min}), & \text{if } \delta > \delta_{\min} + \pi, \\ l_{\min} + \kappa_{\max}^{-1}(\delta - \delta_{\min}), & \text{else.} \end{cases} \tag{24}$$

B. Continuous Curvature Rate Turn

CCR Turns are closely related to HCR Turns with the only difference that $\kappa = \sigma = 0$ at the goal configuration $\mathbf{q}_g$ as shown in Figure 5. This can be achieved by applying the

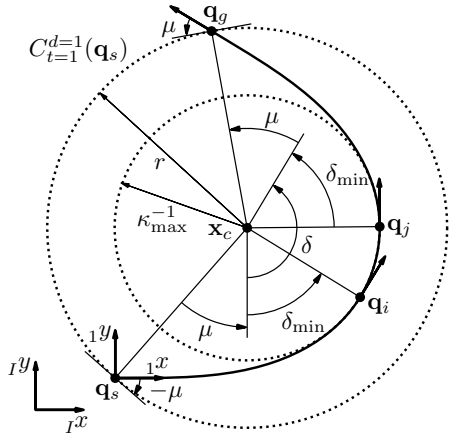

Fig. 5: Regular CCR Turn that starts with $\kappa = \sigma = 0$ in $\mathbf{q}_s$, passes the intermediate configurations $\mathbf{q}_i$ and $\mathbf{q}_j$, and ends with $\kappa = \sigma = 0$ in $\mathbf{q}_g$.

negated continuous curvature rate profile of Figure 3 once the vehicle reaches the second intermediate configuration $\mathbf{q}_j$.

The mathematical description of a CCR Circle C_t^d is the same as in the HCR case. Therefore, only differences in the computation of a CCR Turn are highlighted in the following.

While $\mathbf{q}_i$ is computed according to (17), $\mathbf{q}_j$ is obtained by rotating $\mathbf{q}_i$ on a circle with radius $1/\kappa_{\max}$ centered at $\mathbf{x}_c$ by $\delta - 2\delta_{\min}$. Thus, (21) can be applied by adjusting the rotation angle. Similarly, rotating the initial configuration $\mathbf{q}_s$ by $\delta + 2\mu$ around the center $\mathbf{x}_c$ of a circle with radius r results in the goal configuration $\mathbf{q}_g$. It is given as

$${}_1\mathbf{q}_g = \begin{pmatrix} x_g \\ y_g \\ \theta_g \\ \kappa_g \\ \sigma_g \end{pmatrix} = \begin{pmatrix} x_c + r\cos(-t\frac{\pi}{2} + td(\delta + \mu)) \\ y_c + r\sin(-t\frac{\pi}{2} + td(\delta + \mu)) \\ td\delta \\ 0 \\ 0 \end{pmatrix}. \tag{25}$$

1) Regular CCR Turn: A regular CCR Turn is visualized in Figure 5, where start and goal configuration are connected without direction switches in between. The path length of such a turn with respect to δ can therefore be computed as

$$l(\delta) = \begin{cases} 2r\sin(\mu), & \text{if } \delta = 0, \\ \text{see Section III-B.3}, & \text{if } 0 < \delta < 2\delta_{\min}, \\ 2l_{\min} + \kappa_{\max}^{-1}(\delta - 2\delta_{\min}), & \text{else,} \end{cases} \tag{26}$$

where the first case corresponds to a straight line.

2) Irregular CCR Turn: Irregular CCR Turns result in shorter path lengths than regular ones if $\delta > 2\delta_{\min} + \pi$. However, they require two additional direction switches as illustrated in Figure 6.

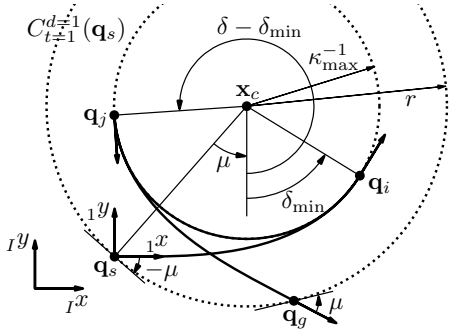

Fig. 6: Irregular CCR Turn that allows a direction switch at $\mathbf{q}_i$ and $\mathbf{q}_j$.

The path length of an irregular CCR Turn is given according to

$$l(\delta) = \begin{cases} 2r\sin(\mu), & \text{if } \delta = 0, \\ \text{see Section III-B.3}, & \text{if } 0 < \delta < 2\delta_{\min}, \\ 2l_{\min} + \kappa_{\max}^{-1}(2\pi - \delta + 2\delta_{\min}), & \text{if } \delta > 2\delta_{\min} + \pi, \\ 2l_{\min} + \kappa_{\max}^{-1}(\delta - 2\delta_{\min}), & \text{else.} \end{cases} \tag{27}$$

3) Elementary Paths: A unique extension of the previously described CCR Turn is the so-called elementary path introduced in [19] and applied to CC Turns in [8]. In contrast to CCR Turns, elementary paths connect the start and goal configuration without reaching the maximum curvature, see Figure 7.

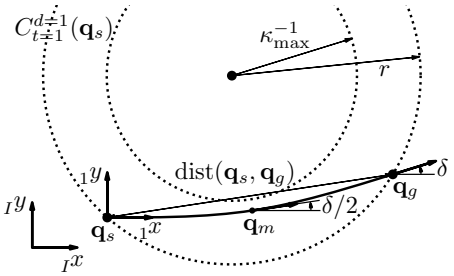

Fig. 7: Start and goal configuration are connected by an elementary path.

For deflections $\delta < 2\delta_{\min}$, elementary paths are favored over regular and irregular CCR Turns as they result in shorter path lengths. In contrast to [8], [19], curvature rate continuity is required here making new derivations of the elementary path necessary. Within this regard, two novel curvature rate continuous profiles are introduced in Figure 8. As it can be

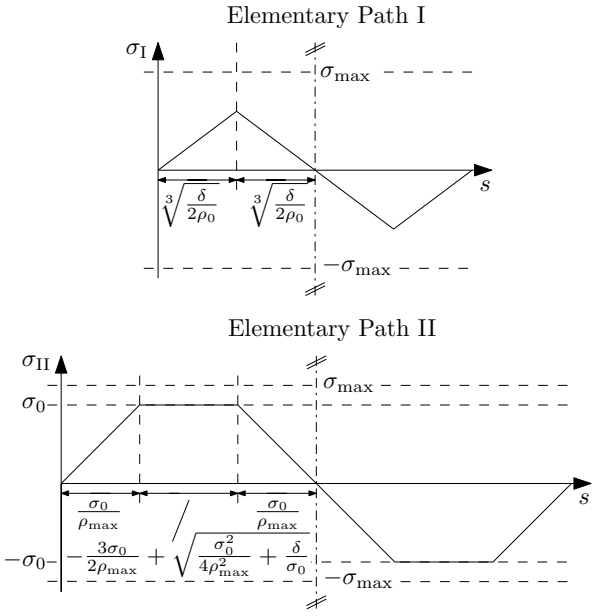

Fig. 8: Continuous curvature rate profiles of the first elementary path consisting of two cubic spirals and of the second elementary path consisting of three cubic spirals on either side of the symmetry axis.

seen, both elementary paths are symmetric, which allows to only consider the left side of the symmetry axis in the following.

Elementary path I consists of two cubic spirals on either side of the symmetry axis. The first one accelerates with $\rho_0 \leq \rho_{\max}$ and the second one decelerates with $-\rho_0$. The length of each spiral is derived from the fact that $\kappa = \sigma = 0$ at the start configuration $\mathbf{q}_s$ and $\theta = \delta/2$ at the middle configuration $\mathbf{q}_m$, see Figure 7. Note that $\delta \leq 2\sigma_{\max}^3/\rho_0^2$ has to be ensured in order to not violate the maximum curvature rate $\sigma_{\max}$.

In contrast to that, elementary path II consists of three cubic spirals on either side of the symmetry axis. The first one accelerates with $\rho_{\max}$ until it reaches $\sigma_0 \leq \sigma_{\max}$. The second one keeps the curvature rate σ_0 constant while the third cubic spiral decelerates to 0 with $-\rho_{\max}$. While the length of the first and third cubic spiral can be trivially derived, the length of the second one can be calculated by enforcing $\theta = \delta/2$ at $\mathbf{q}_m$.

Each elementary path has one unknown parameter ρ_0 and σ_0 respectively. It can be computed by projecting $\mathbf{q}_m$ onto the straight line connecting $\mathbf{q}_s$ and $\mathbf{q}_g$ according to

$$\begin{pmatrix}\cos(\delta/2)\\ \sin(\delta/2)\end{pmatrix} \cdot \begin{pmatrix}x_m\\ y_m\end{pmatrix} = \frac{\mathrm{dist}(\mathbf{q}_s, \mathbf{q}_g)}{2}, \tag{28}$$

where x_m, y_m denote the Cartesian coordinates of $\mathbf{q}_m$, and $\mathrm{dist}(\mathbf{q}_s, \mathbf{q}_g)$ evaluates the Euclidean distance between $\mathbf{q}_s$ and $\mathbf{q}_g$, see Figure 7. In order to solve (28), x_m and y_m are replaced by their corresponding integrals given in (5, 6) and evaluated along the left half of the elementary path. As previously mentioned, these integrals do not possess a closed-form solution, which requires a numerical root-finding algorithm to solve for ρ_0 and σ_0 respectively.

If $\rho_{\max} = \infty$, there exists a unique elementary path for all $\delta < 2\delta_{\min}$ as shown in [20]. However if $\rho_{\max} < \infty$, the existence of a solution to (28) can not be guaranteed anymore, neither can the maximum deflection for which a solution exists be derived analytically.

Therefore, the path length for $\delta < 2\delta_{\min}$ is given as

$$l(\delta) = \begin{cases} \frac{\sigma_0}{\rho_{\max}} + \sqrt{\frac{\sigma_0^2}{\rho_{\max}^2} + \frac{4\delta}{\sigma_0}}, & \text{if } \sigma_0 \neq \emptyset, \\ \sqrt[3]{\frac{32\delta}{\rho_0}}, & \text{elif } \rho_0 \neq \emptyset, \\ 2l_{\min} + \kappa_{\max}^{-1}(-\delta + 2\delta_{\min}), & \text{elif irreg. turn}, \\ 2l_{\min} + \kappa_{\max}^{-1}(2\pi + \delta - 2\delta_{\min}), & \text{else (reg. turn).} \end{cases} \tag{29}$$

IV. HCR and CCR Steer

This section briefly describes how to compute HCR and CCR Steer on the basis of the derived G^3 continuous turns. Note, however, that a similar procedure is applied in HC and CC Steer, and therefore, the reader is referred to [7], [8] for a detailed derivation and to [21] for an implementation.

Straight lines together with HCR and CCR Turns can be concatenated in order to create three different steering functions that satisfy (1). The first steering function is called CCR-Dubins, which only allows to drive either forwards or backwards. Going forwards and backwards while ensuring curvature rate continuity is the characteristic of the second steering function named CCR-Reeds-Shepp. In contrast to that, the third steering function HCR-Reeds-Shepp only enforces curvature rate continuity while the vehicle is going in one direction, but allows curvature discontinuities at direction switches.

The optimization criterion for all three steering functions is path length. Therefore, a set of families that concatenates different combinations of turns and straight lines with tangent conditions is evaluated, and the path with the shortest length is chosen. In contrast to Dubins and RS Steer, an infinite number of possible families exists, also compare [8]. It is therefore proposed for CCR-Dubins to evaluate the Dubins Families and for HCR-RS and CCR-RS to compute the Dubins, RS, and three additional families, which have proven to be beneficial in the past [7], [20]. All families are given in Table I.

TABLE I: HCR and CCR Families, where C describes a turn, S a straight line, and $|$ a direction switch.

Dubins Families	RS Families	Additional Families
CCC	$C\|C\|C$	$C\|SC$
CSC	$C\|CC$	$CS\|C$
	$CC\|C$	$C\|S\|C$
	CSC	
	$CC\|CC$	
	$C\|CC\|C$	
	$C\|CSC$	
	$CSC\|C$	
	$C\|CSC\|C$	

It has to be noted that HCR-RS comes in four different forms, namely HCR^{00}-RS, $\mathrm{HCR}^{0\pm}$-RS, $\mathrm{HCR}^{\pm 0}$-RS, and $\mathrm{HCR}^{\pm\pm}$-RS, where a superscript describes the initial and final curvature. For instance, $\mathrm{HCR}^{0\pm}$-RS starts with zero curvature and ends with either $\pm\kappa_{\max}$.

Only HCR$^{\pm\pm}$-RS can be guaranteed to be topologically admissible [22] without the introduction of additional, in reality impractical, topological paths. The proof is equal to the one shown in [7].

In order to evaluate the performance of CCR-Dubins, HCR-RS, and CCR-RS Steer, the following two subsections compare the computation times and path lengths with Dubins, CC-Dubins, RS, HC-RS, and CC-RS Steer implemented in [21]. The parameters are set to $\kappa_{\max} = 1\,\mathrm{m}^{-1}, \sigma_{\max} = 1\,\mathrm{m}^{-2}, \rho_{\max} = 1\,\mathrm{m}^{-3}$, and the results are shown for 10^5 random steering procedures. All computations are evaluated on a single core of an Intel Xeon E5@3.50 GHz, 10 MB cache.

A. Computation Times

The computation times of the different steering functions are shown in Table II, where the superscript denotes the initial and final curvature.

TABLE II: Comparison of the computation times

	computation time	
	mean [µs]	std [µs]
Dubins	1.2	±0.8
CC^{00}-Dubins	4.2	±1.9
CCR^{00}-Dubins	11.8	±12.9
RS	7.1	±1.4
HC$^{\pm\pm}$-RS	55.2	±8.9
HC^{00}-RS	53.8	±7.7
CC^{00}-RS	53.1	±8.2
HCR$^{\pm\pm}$-RS	59.5	±12.5
HCR^{00}-RS	99.4	±42.4
CCR^{00}-RS	139.1	±57.5

It can be observed that the curvature and curvature rate continuity come at the cost of higher computation times. This is due to the fact that numerically more expensive calculations have to be conducted compared to the analytic solutions of Dubins and RS Steer.

While all steering functions solve the two-point BVP on average under 139.1 µs, the continuous curvature rate solution is approximately one order of magnitude slower than its curvature discontinuous counterparts. Allowing the vehicle to drive forwards and backwards results in higher computation times because more families have to evaluated, see Table I.

The computation of CCR^{00}-RS is about three times slower than CC^{00}-RS while HCR$^{\pm\pm}$-RS and HC$^{\pm\pm}$-RS have almost equal computation times. It can therefore be concluded that solving the parameters of the elementary path by numerically evaluating (28) is the computationally most expensive part of the curvature rate continuous steering functions.

B. Path Length Comparison

Figure 9 illustrates a qualitative comparison of the steering functions Dubins, CC^{00}-Dubins, and CCR^{00}-Dubins. It can be seen that Dubins Steer results in the shortest path, however, assumes that $\sigma_{\max} = \rho_{\max} = \infty$. Since no physical actuator can fulfill these constraints when the vehicle is in motion, it is required to stop the car at every curvature discontinuity. In contrast to that, CC^{00}-Dubins Steer only requires $\rho_{\max} = \infty$ and therefore generates a slightly longer curvature continuous path. Arbitrary values for $\sigma_{\max}$ and $\rho_{\max}$ are allowed by CCR^{00}-Dubins Steer. Consequently, it results in the smoothest, but also longest path.

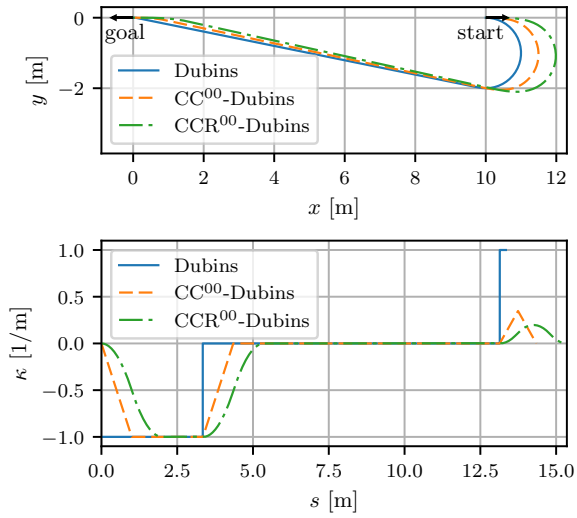

Fig. 9: Qualitative comparison of the driven path (top) and the resulting curvature profile (bottom) of Dubins, CC^{00}-Dubins, and CCR^{00}-Dubins Steer.

A quantitative comparison of the path lengths is given in Figure 10.

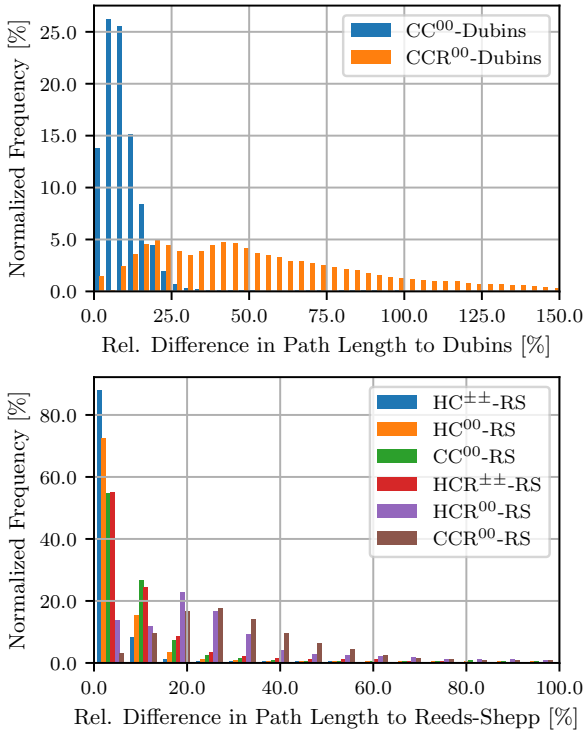

Fig. 10: Quantitative comparison of the path lengths with respect to the shortest path according to Dubins (top) and Reeds-Shepp (bottom).

The top image in Figure 10 compares the path lengths of Dubins Steer with the steering functions CC^{00}-Dubins and CCR^{00}-Dubins. It can be seen that smoothness comes at the cost of significantly longer paths for a car that only moves forwards. In contrast to that, allowing the car to drive both

forwards and backwards results in a better approximation of the shortest path given by RS Steer, see Figure 10 on the bottom. In case of $HC^{\pm\pm}$-RS, HC^{00}-RS, CC^{00}-RS, and $HCR^{\pm\pm}$-RS, more than $50\,\%$ of steering procedures deviate less than $7\,\%$ from the shortest path. The largest deviations from RS Steer are obtained by CCR^{00}-RS because it enforces curvature rate continuity along the entire path.

V. Experimental Results

The discussed G^1, G^2, and G^3 continuous steering functions are integrated into the motion planner Bidirectional RRT* (BiRRT*) [23], [24]. The performance is evaluated in three automated driving scenarios using Gazebo[1] and ROS[2]. As it can be seen in Figure 11, the scenarios include a dead end, a blocked driveway, and a high density parking lot [25]. A video of the experimental results can be found online[3].

The vehicle parameters are given in Table III, where the values of $\kappa_{\max}$, $\sigma_{\max}$, and $\rho_{\max}$ already include $10\,\%$ control reserve. The contour of the vehicle is approximated by a polygon with 20 vertices. It is inflated by a safety distance of $10\,\mathrm{cm}$.

TABLE III: Vehicle parameters

Parameter	Symbol	Value
Length	-	$4.926\,\mathrm{m}$
Width	-	$2.086\,\mathrm{m}$
Wheel Base	-	$2.912\,\mathrm{m}$
Max. Curvature	$\kappa_{\max}$	$0.1982\,\mathrm{m}^{-1}$
Max. Curvature Rate	$\sigma_{\max}$	$0.1868\,\mathrm{m}^{-2}$
Max. Curvature Acceleration	$\rho_{\max}$	$0.3905\,\mathrm{m}^{-3}$

To compensate for randomization effects of BiRRT*, every experiment is repeated 100 times. The sampling duration for an initial solution is set to $5\,\mathrm{s}$, otherwise, the run is classified as failed. If a solution is found, the planner will be allowed to optimize it for another $3\,\mathrm{s}$. The configurations are sampled uniformly and a goal sampling frequency of $5\,\%$ is applied. The parameter γ of BiRRT* is set to 6.

The selected cost function penalizes path length, cusps, and the curvature along the path, compare to [7]. An additional footprint cost evaluated on an inflated grid map with $7.5\,\mathrm{cm}$ resolution and $25\,\mathrm{cm}$ inflation radius is added to the total cost. Collision checks are performed every $10\,\mathrm{cm}$.

Table IV compares the performance of BiRRT* in the three given scenarios. Each scenario is solved with a G^1, G^2, and G^3 continuous steering function and evaluated according to the metrics: time to first solution (TTFS), number of curvature and curvature rate discontinuities, number of cusps, path length, and success rate.

As it can be seen in Table IV, BiRRT* in combination with the G^1 continuous steering functions solves the given problems with the least amount of time and shortest path length. However, an average of up to 13.2 curvature discontinuities makes the resulting paths uncomfortable to drive and puts a high mechanical stress on the steering system. In contrast to that, the G^2 continuous steering functions enforce curvature continuity, but allow up to 24.9 curvature rate discontinuities. The consequence might be a poor tracking performance of the vehicle controller leading to deviations and potential collisions in tight environments. The G^3 continuous steering functions outperform the G^1 continuous ones in terms of curvature continuity and the G^2 continuous ones in terms of curvature rate continuity. As a result, HCR and CCR Steer lead to the smoothest paths that can be easily tracked by the vehicle controller. Note, however, that the higher computation times and the longer path lengths of the G^3 continuous steering functions make finding solutions in very tight scenarios more difficult. This fact can be seen in Table IV by comparing the success rates in scenario II.

[1] http://gazebosim.org/

[2] https://ros.org/

[3] https://youtu.be/DLjeuGgDcTM

VI. Conclusion and Outlook

In this paper, the novel G^3 continuous steering functions Hybrid Curvature Rate (HCR) and Continuous Curvature Rate (CCR) Steer are introduced. Hard constraints on the maximum curvature, maximum curvature rate, and maximum curvature acceleration are enforced at all times leading to directly driveable smooth paths. The potential of HCR and CCR Steer in combination with Bidirectional RRT* is demonstrated in three challenging automated driving scenarios.

In the future, we aim to increase the computational efficiency of the presented steering functions by improving the computation of the elementary paths. Moreover, it is planned to speed up the motion planner by adapting the sampling to the environment.

References

[1] D. González *et al.*, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[2] D. Dolgov *et al.*, "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments," *The International Journal of Robotics Research*, 2010.

[3] S. Karaman and E. Frazzoli, "Incremental Sampling-based Algorithms for Optimal Motion Planning," *Robotics Science and Systems VI*, 2010.

[4] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, 1957.

[5] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, 1990.

[6] D. H. Shin and S. Singh, "Path Generation for Robot Vehicles Using Composite Clothoid Segments," Canergie Mellon University, Tech. Rep., 1990, Research Note CMU-RI-TR-90-312.

[7] H. Banzhaf *et al.*, "Hybrid Curvature Steer: A Novel Extend Function for Sampling-Based Nonholonomic Motion Planning in Tight Environments," in *IEEE International Conference on Intelligent Transportation Systems*, 2017.

[8] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to Continuous-Curvature Paths," *IEEE Transactions on Robotics*, 2004.

[9] J. Reuter, "Mobile Robots Trajectories with Continuously Differentiable Curvature: An Optimal Control Approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[10] C. Xie *et al.*, "Toward Asymptotically Optimal Motion Planning for Kinodynamic Systems using a Two-Point Boundary Value Problem Solver," in *IEEE International Conference on Robotics and Automation*, 2015.

[11] A. Kelly and B. Nagy, "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control," *The International Journal of Robotics Research*, 2003.

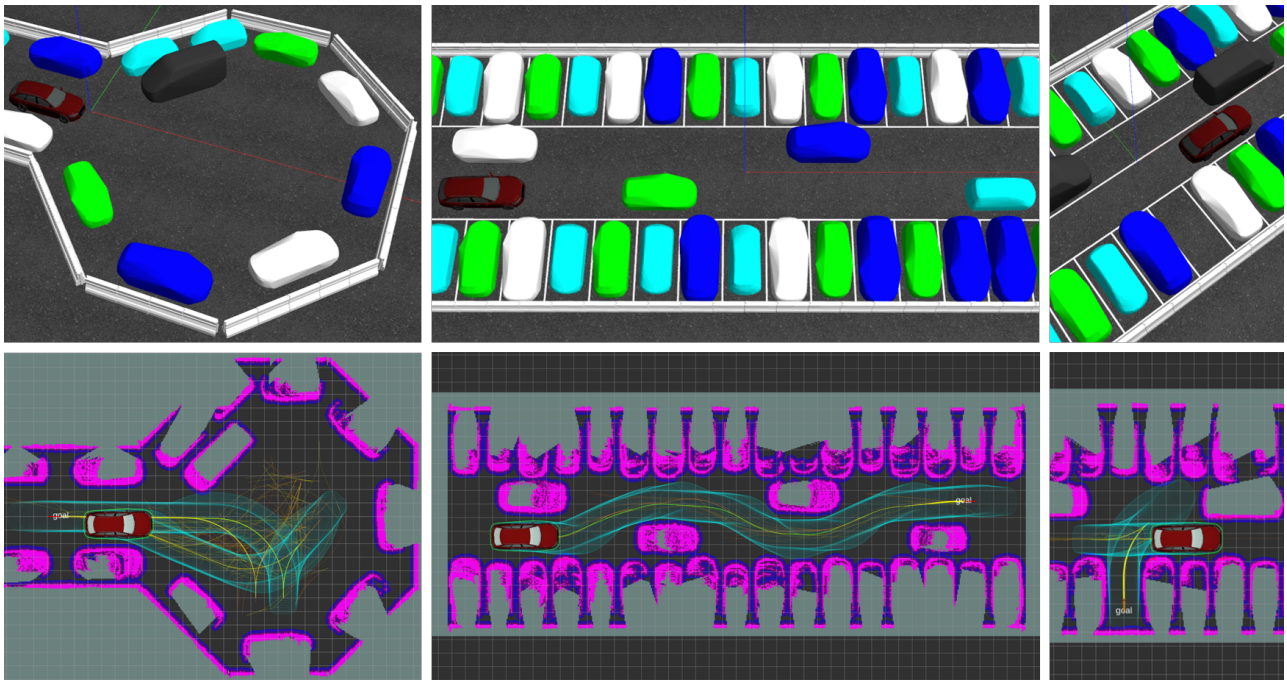

Fig. 11: The test scenarios for evaluating the motion planner are visualized in the top row, and example paths in the bottom row. Scenario I on the left shows a dead end, where the red ego vehicle needs to turn around while the parked black transporter is in its way. Scenario II in the middle requires the ego vehicle to drive through a blocked driveway. A parking maneuver has to be performed in scenario III on the right, where the complexity stems from the high density parking environment. The visualized paths in the bottom row are generated with CCR^{00}-RS on the left, CCR^{00}-Dubins in the middle, and HCR^{00}-RS on the right.

TABLE IV: Results of BiRRT* after 100 repetitions of the same experiment in the three described scenarios. The time to first solution t_{TTFS}, the number of curvature and curvature rate discontinuities, the number of cusps, and the path lengths are listed with mean and standard deviation per repetition.

		continuity	t_{TTFS} [s]	#curv. discon. [–]	#curv. rate discon. [–]	#cusps [–]	length [m]	success rate [%]
Scen. I	RS	G^1	0.01±0.00	6.5±0.9	0.0±0.0	2.0±0.0	33.0±2.3	100
	CC^{00}-RS	G^2	0.02±0.02	0.0±0.0	14.0±1.7	2.0±0.0	37.3±2.5	100
	CCR^{00}-RS	G^3	0.03±0.02	0.0±0.0	0.0±0.0	2.1±0.4	40.5±3.3	100
Scen. II	Dubins	G^1	0.29±0.22	13.2±2.0	0.0±0.0	0.0±0.0	34.9±0.1	100
	CC^{00}-Dubins	G^2	0.31±0.29	0.0±0.0	24.9±4.2	0.0±0.0	34.9±0.2	100
	CCR^{00}-Dubins	G^3	2.01±1.20	0.0±0.0	0.0±0.0	0.0±0.0	35.0±0.1	88
Scen. III	RS	G^1	0.02±0.02	5.9±1.3	0.0±0.0	2.2±0.5	11.1±0.8	100
	HC^{00}-RS	G^2 btw. cusps	0.08±0.10	1.7±0.9	8.8±1.8	2.9±0.9	13.9±1.9	100
	HCR^{00}-RS	G^3 btw. cusps	0.20±0.28	1.7±0.8	0.0±0.0	3.7±1.3	17.6±2.9	100

[12] M. McNaughton, *Parallel Algorithms for Real-time Motion Planning*. Carnegie Mellon University, 2011.

[13] R. Oliveira *et al.*, "Trajectory Generation Using Sharpness Continuous Dubins-Like Paths with Applications in Control of Heavy Duty Vehicles," in *European Control Conference*, 2018.

[14] Y. Kanayama and B. I. Hartman, "Smooth Local Path Planning for Autonomous Vehicles," in *IEEE International Conference on Robotics and Automation*, 1989.

[15] F. Dillen, "The Classification of Hypersurfaces of a Euclidean Space with Parallel Higher Order Fundamental Form," *Mathematische Zeitschrift*, 1990.

[16] A. Piazzi *et al.*, "η^3-Splines for the Smooth Path Generation of Wheeled Mobile Robots," *IEEE Transactions on Robotics*, 2007.

[17] T. Gawron and M. M. Michalek, "Planning G^3-continuous paths for state-constrained mobile robots with bounded curvature of motion," in *Trends in Advanced Intelligent Control, Optimization and Automation*, 2017.

[18] G. Parlangeli and G. Indiveri, "Dubins inspired 2D smooth paths with bounded curvature and curvature derivative," in *IFAC Symposium on Intelligent Autonomous Vehicles*, 2010.

[19] A. Scheuer and T. Fraichard, "Collision-Free and Continuous-Curvature Path Planning for Car-Like Robots," in *IEEE International Conference on Robotics and Automation*, 1997.

[20] A. Scheuer and T. Fraichard, "Continuous-Curvature Path Planning for Car-Like Vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1997.

[21] H. Banzhaf. (2017) Steering Functions for Car-Like Robots. (visited on 2017/12/04). [Online]. Available: https://github.com/hbanzhaf/steering_functions

[22] S. Sekhavat and J.-P. Laumond, "Topological Property for Collision-Free Nonholonomic Motion Planning: The Case of Sinusoidal Inputs for Chained Form Systems," *IEEE Transactions on Robotics and Automation*, 1998.

[23] M. Jordan and A. Perez, "Optimal Bidirectional Rapidly-Exploring Random Trees," Massachusetts Institute of Technology, Tech. Rep., 2013, TR 2013-021.

[24] S. Klemm *et al.*, "RRT*-Connect: Faster, Asymptotically Optimal Motion Planning," in *IEEE International Conference on Robotics and Biomimetics*, 2015.

[25] H. Banzhaf *et al.*, "High Density Valet Parking Using k-Deques in Driveways," in *IEEE Intelligent Vehicles Symposium*, 2017.