# Hybrid Curvature Steer: A Novel Extend Function for Sampling-Based Nonholonomic Motion Planning in Tight Environments

Holger Banzhaf[1], Luigi Palmieri[2], Dennis Nienhüser[1], Thomas Schamm[1], Steffen Knoop[1], J. Marius Zöllner[3]

***Abstract*— Finding optimal paths for self-driving cars in cluttered environments is one of the major challenges in autonomous driving. The complexity stems from the nonlinearity of the system and the non-convexity of the configuration space. This paper introduces a novel extend function called Hybrid Curvature (HC) Steer for sampling-based nonholonomic motion planning in tight environments. HC Steer solves the two-point boundary value problem by computing continuous curvature paths while the vehicle is going in one direction, but allows curvature discontinuities at switches in the driving direction. The resulting paths approximate Reeds-Shepp's paths and are directly drivable by an autonomous vehicle. BiRRT*, an optimal sampling-based motion planner, is used to evaluate and compare HC Steer's performance to state of the art steering functions, namely Reeds-Shepp (RS) and Continuous Curvature (CC) Steer. Extensive experiments in challenging environments show HC Steer's advantage of computing smoother paths than RS Steer in equally tight environments and finding solutions with less direction switches, higher success rates, and shorter planning time than CC Steer.**

## I. Introduction

Advances in autonomous driving will soon show the first commercially available Automated Valet Parking (AVP) systems, where the vehicle is left in a drop-off zone and executes the driving and parking task autonomously [1]. Within this context, a future concept is High Density Parking (HDP) [2]. It leverages the potential of AVP by increasing the number of parked vehicles on an existing parking lot by (1) packing the vehicles denser since humans can already exit the car in the drop-off zone, and (2) changing the parking layout, e.g. allowing vehicles to also park on one side of the driveway [3], see Figure 1. As a consequence, free space in the parking area will be reduced thus making the motion planning problem even more challenging.

Despite the numerous research projects since the DARPA Urban Challenge (DUC) in 2007, motion planning for nonholonomic systems in cluttered environments is still an actively researched topic [4]. The key challenge lies in the design of a generic motion planner, which takes into account the nonlinearity of the system and the non-convexity of the configuration space. Preferably, a motion planner for such a problem should fulfill the following four criteria [4], [5]: (1) Given an arbitrary environment, it should find a collision-free solution if one exists (completeness). (2) The solution should minimize an objective function (optimality). (3) The computed path should be easily drivable with the given actuator limits, and (4) the computational cost should be rather small.

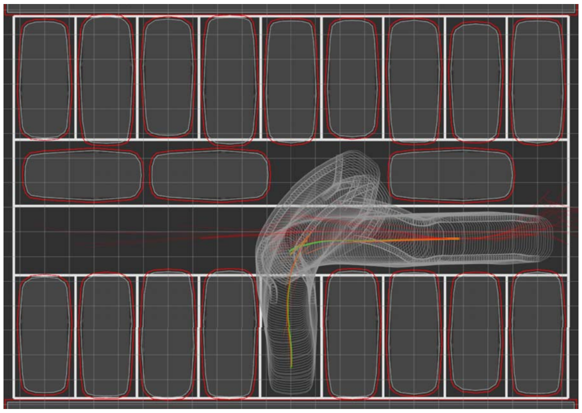

Fig. 1: Maneuvering in a tight parking environment with BiRRT* and Hybrid Curvature Steer. The solution path is visualized in green with four direction switches. The trees rooted at the start and the goal configuration are colored in red and orange.

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

- Introduction of a novel steering function called Hybrid Curvature (HC) Steer. It generates curvature continuous paths while the vehicle is going in one direction, but allows curvature discontinuities at direction switches.
- Comparison of HC Steer with state of the art steering functions, namely Reeds-Shepp (RS) [6] and Continuous Curvature (CC) Steer [7], in terms of path length, computation time, and topological admissibility. HC Steer's capability of approximating RS Steer and outperforming CC Steer in terms of path length while having comparable computation times to CC Steer is shown.
- Evaluation and comparison of RS, HC, and CC Steer's performance in the optimal motion planner BiRRT* [8] on two HDP scenarios. HC Steer's advantage of computing smoother paths than RS Steer in equally tight environments and finding solutions with less direction switches, higher success rates, and shorter planning time than CC Steer is illustrated.

The remainder of this paper is **organized** as follows.

[1]Holger Banzhaf, Dennis Nienhüser, Thomas Schamm, and Steffen Knoop are with Robert Bosch GmbH, Corporate Research, Driver Assistance Systems and Automated Driving, 71272 Renningen, Germany.

[2]Luigi Palmieri is with Robert Bosch GmbH, Corporate Research, Future Robotics Systems, 71272 Renningen, Germany. His work has been partly funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732737 (ILIAD).

[3]J. Marius Zöllner is with FZI Research Center for Information Technology, 76131 Karlsruhe, Germany.

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

Section II gives a brief overview of related work. BiRRT* is described in Section III, and HC Steer is introduced in Section IV. The collision checker and the cost function for planning with BiRRT* are detailed in Section V. The experimental results are analyzed in Section VI, and a conclusion and outlook is given in Section VII.

## II. Related Work

Recent surveys on motion planning for self-driving cars can be found in [9] and [10]. Based on [9], the different approaches can be categorized into four groups: graph search, random/deterministic sampling, interpolating curves, and numerical optimization.

The core idea of graph search-based planners is to discretize the state and action space and search the resulting graph for the optimal solution [11], [12], [13]. With appropriate heuristics, graph search-based planners compute fast solutions. However, they suffer from completeness and optimality only with respect to the discretization, and exponentially growing computations with smaller discretization. In addition, the discretization may result in unnatural paths requiring an additional smoothing step [12].

Sampling-based planners draw their samples either from a discretized set of actions [14], [15] or randomly [16], [17], [18]. While discrete sampling shares the advantages and drawbacks of graph-based planners, random sampling does not rely on discretization resulting in probabilistic completeness. Additionally, randomized planners like RRT* [19] also guarantee asymptotic optimality. Compared to graph search, these advantages come at a cost of higher computation times.

Interpolating curve planners such as [20] concatenate a set of curves, e.g. lines, circles, and clothoids, to plan a feasible path. In general, they are fast, but inflexible because scenario-specific rules for the concatenation of the curves have to be derived. Besides, completeness can not be guaranteed.

Optimization-based planners [21], [22], [23] formulate the path planning problem as an optimal control problem and solve it with numerical optimization. A discretization of the state space can be avoided. In order to guarantee a fast convergence to the optimal solution, the optimization problem needs to be convexified. Approximating general environments as convex spaces is either challenging or suffers from inaccuracies due to convex approximations.

The following chapter briefly describes Bidirectional RRT* (BiRRT*) [8] as the general motion planner (sampling-based, randomized) used in this paper.

## III. BiRRT*

Rapidly-exploring Random Trees (RRT) [24] have proven to quickly solve high dimensional single-query motion planning problems in various robotic domains, such as autonomous driving [25]. A probabilistically complete and asymptotically optimal variant of RRT, called RRT*, was introduced in [19]. The basic idea of RRT* is to incrementally build a tree from a start to a goal configuration in three steps: (1) Sample a random configuration in the obstacle-free configuration space, (2) connect it with a collision-free, minimum-cost extension to the tree, and (3) rewire the tree locally in order to converge asymptotically to an optimal solution. An in-depth description of the algorithm for nonholonomic systems can be found in [26]. BiRRT* [8] is a two-tree version of RRT*, which builds a tree from the start to the goal and vice versa. It has shown an improved performance compared to RRT* in complex environments.

Applying RRT* to autonomous driving requires a fast, (sub)optimal solution to the two-point boundary value problem (BVP) of a car model, which connects a newly sampled configuration to the tree. This is often referred to as extend/steering procedure. Known steering functions for forward driving are either optimal controllers for linearized vehicle models [27], shooting methods [16], or geometric approaches with a Dubins' car [28], [17], [26]. Steering functions for driving with direction switches are Reeds-Shepp (RS) [6], [18] and Continuous Curvature (CC) Steer [7].

While RS Steer solves the BVP optimally with respect to path length, CC Steer does not strictly optimize any objective function anymore. The discrete nature of RS Steer exposes the vehicle to significant stress and makes the resulting paths uncomfortable to drive. In contrast to that, CC Steer adds comfort to the computed paths, but decreases the maneuverability of the car significantly, e.g. requires more direction switches for the same maneuver. A novel steering function that overcomes these issues is described in the next section.

## IV. Hybrid Curvature Steer

This section introduces Hybrid Curvature (HC) Steer, a novel steering function that is inspired by human driving in tight environments. HC Steer computes continuous curvature paths while the vehicle is going in one direction, but allows curvature discontinuities at changes in the driving direction, in the following referred to as cusps. This approach combines the advantages of RS and CC Steer, namely approximating RS Steer in terms of path length, ensuring curvature continuity between cusps, and increasing the maneuverability compared to CC Steer.

In the following subsections, HC Steer is described in detail. The car model for HC Steer is described in Subsection IV-A. HC Turns as the basic components of HC Paths are introduced in Subsection IV-B. A description of the computation of HC Paths, a comparison with RS and CC Steer, and an analysis of the topological admissibility are given in Subsections IV-C, IV-D, and IV-E, respectively.

### A. Car Model for HC Steer

For path planning at small velocities, the dynamics of a vehicle can be described by a kinematic bicycle model [29]. It is given with respect to arc length $s$ by

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

where the position of the midpoint of the rear axle is described by $(x, y)$, the orientation of the car by $\theta$, and the curvature of the path at position $(x, y)$ by $\kappa$. The driving direction $d$ and the change of curvature $\sigma$, also referred to as sharpness, are the inputs of the system. The derivatives with respect to $s$ are given by $(\bullet)'$. The configuration of the vehicle can be abbreviated by $\mathbf{q} = [x, y, \theta, \kappa]^T$ and the input by $\mathbf{u} = [d, \sigma]^T$.

The curvature $\kappa$ and the sharpness $\sigma$ at a given velocity $v$ are constrained by the physical limits of the car $\kappa_{\text{max}}$ and $\sigma_{\text{max}}$. Note that $\sigma_{\text{max}}$ is inversely proportional to $v$. Therefore, HC Steer requires

$$|\sigma| = +\infty, \text{ if cusp,} \tag{2}$$

$$|\sigma| \leq \sigma_{\text{max}}, \text{ else,} \tag{3}$$

where $\sigma_{\text{max}}$ is given for the maximum velocity along the path.

### B. HC Turns

HC Turns are the fundamental component of HC Paths. A HC Turn starts with a configuration of zero curvature and results in a configuration with maximum curvature. Therefore, it consists of a clothoid and a circular arc and is entirely defined by the direction of the movement $d \in \{-1, 1\}$ (backwards, forwards), the direction of the turn $t \in \{-1, 1\}$ (right, left), the start configuration $\mathbf{q}_s$, and the change of orientation between the start and the goal configuration $\delta \in [0, 2\pi[$, also denoted as deflection.

Similar to CC Turns, there are two different HC Turns, a regular and an irregular one, which are described in the following.

*1) Regular HC Turn:* Given a start configuration $\mathbf{q}_s$ and the direction of the movement, a HC Circle $C_t^d(\mathbf{q}_s)$ can be computed that contains the position of the start configuration and whose tangent encloses the angle $\mu$ with $\mathbf{q}_s$. Figure 2 visualizes such a HC Circle including a HC Turn for a forwards movement ($d = 1$) to the left ($t = 1$).

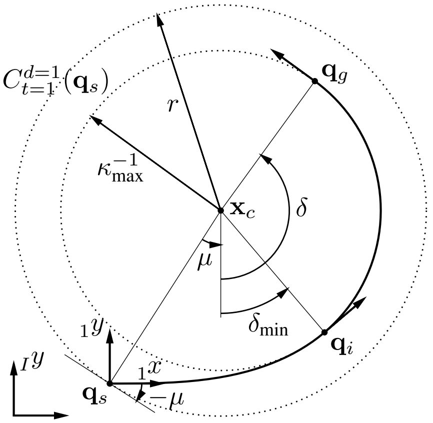

Fig. 2: A regular HC Turn, where the vehicle starts in $\mathbf{q}_s$ and moves on a clothoid and a circle to $\mathbf{q}_g$.

The following paragraphs detail the mathematical description of a HC Circle and a HC Turn.

The start configuration of the vehicle in an inertial frame $I$ is given as ${}_I\mathbf{q}_s = (x_s, y_s, \theta_s, 0)$. The intermediate configuration ${}_1\mathbf{q}_i$ in the local frame 1 at the end of the clothoid can be computed as

$${}_1\mathbf{q}_i = \begin{pmatrix} x_i \\ y_i \\ \theta_i \\ \kappa_i \end{pmatrix} = \begin{pmatrix} d\sqrt{\pi/\sigma_{\text{max}}} C_f(\sqrt{\kappa_{\text{max}}^2/(\pi\sigma_{\text{max}})}) \\ t\sqrt{\pi/\sigma_{\text{max}}} S_f(\sqrt{\kappa_{\text{max}}^2/(\pi\sigma_{\text{max}})}) \\ td\kappa_{\text{max}}^2/(2\sigma_{\text{max}}) \\ t\kappa_{\text{max}} \end{pmatrix}, \tag{4}$$

where the Fresnel Integrals are defined as $C_f(t) = \int_0^t \cos(\frac{\pi}{2}u^2)du$ and $S_f(t) = \int_0^t \sin(\frac{\pi}{2}u^2)du$.

The transformation between the local frame 1 and the inertial frame $I$ is given by

$${}_I\mathbf{q}_\star = {}_I\mathbf{q}_s + A_{I1} \cdot {}_1\mathbf{q}_\star, \tag{5}$$

where $\star$ stands for an arbitrary configuration and $A_{I1}$ is the transformation matrix from the local frame to the inertial frame.

The deflection $\delta_{\text{min}}$ describes the change of orientation between start and intermediate configuration and can be calculated as

$$\delta_{\text{min}} = \kappa_{\text{max}}^2/(2\sigma_{\text{max}}). \tag{6}$$

The center of the HC Circle $\mathbf{x}_c$ is given by

$${}_1\mathbf{x}_c = \begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} x_i - \kappa_i^{-1}\sin(\theta_i) \\ y_i + \kappa_i^{-1}\cos(\theta_i) \end{pmatrix}, \tag{7}$$

and its radius by $r = \|{}_1\mathbf{x}_c\|_2$.

The goal state $\mathbf{q}_g$ of the HC Turn is obtained by rotating $\mathbf{q}_i$ on a circle with radius $\kappa_{max}^{-1}$ and center $\mathbf{x}_c$ by $\delta - \delta_{\text{min}}$ according to

$${}_1\mathbf{q}_g = \begin{pmatrix} x_g \\ y_g \\ \theta_g \\ \kappa_g \end{pmatrix} = \begin{pmatrix} x_c + \kappa_i^{-1}\sin(\theta_i + td(\delta - \delta_{\text{min}})) \\ y_c - \kappa_i^{-1}\cos(\theta_i + td(\delta - \delta_{\text{min}})) \\ \theta_i + td(\delta - \delta_{\text{min}}) \\ \kappa_i \end{pmatrix}. \tag{8}$$

The angle between the orientation at the start configuration and the tangent to the HC Circle at that position is denoted by $\mu$ and computed as

$$\mu = \arctan(x_c/y_c). \tag{9}$$

The length of the path on a regular HC Turn is defined as

$$l(\delta) = \begin{cases} l_{\text{min}} + \kappa_{\text{max}}^{-1}(2\pi + \delta - \delta_{\text{min}}), & \text{if } \delta < \delta_{\text{min}} \\ l_{\text{min}} + \kappa_{\text{max}}^{-1}(\delta - \delta_{\text{min}}), & \text{if } \delta \geq \delta_{\text{min}}, \end{cases} \tag{10}$$

where $l_{\text{min}} = \kappa_{\text{max}}\sigma_{\text{max}}^{-1}$ describes the length of the clothoid.

*2) Irregular HC Turn:* This turn results in a shorter path than the regular one for $\delta < \delta_{\text{min}}$ and $\delta > \delta_{\text{min}} + \pi$ when a switch in direction is allowed at the intermediate configuration $\mathbf{q}_i$. Figure 3 visualizes an irregular HC Turn for a forwards movement ($d = 1$) to the left ($t = 1$).

The length of the path on an irregular HC Turn can be computed as

$$l(\delta) = \begin{cases} l_{\text{min}} + \kappa_{\text{max}}^{-1}(-\delta + \delta_{\text{min}}) & \text{if } \delta < \delta_{\text{min}}, \\ l_{\text{min}} + \kappa_{\text{max}}^{-1}(2\pi - \delta + \delta_{\text{min}}) & \text{if } \delta > \delta_{\text{min}} + \pi, \\ l_{\text{min}} + \kappa_{\text{max}}^{-1}(\delta - \delta_{\text{min}}) & \text{else.} \end{cases} \tag{11}$$

Note however that the driving direction at the start and end positions are different due to the cusp at the intermediate configuration.

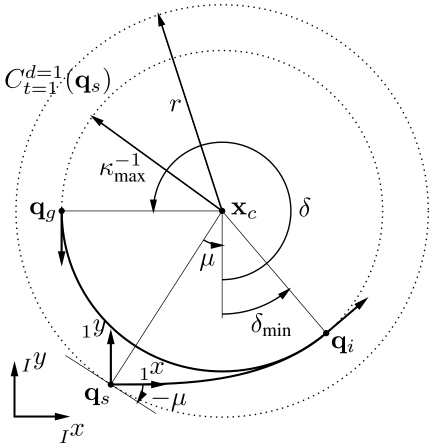

Fig. 3: An irregular HC Turn, where the vehicle starts in $\mathbf{q}_s$, moves on a clothoid to the intermediate configuration $\mathbf{q}_i$, switches direction, and reaches $\mathbf{q}_g$ on a circle.

*3) Comparison of RS, HC, CC Turns:* In order to better understand the differences between RS, HC, and CC Steer, the respective turns are compared here. Figure 4 illustrates a RS, HC, and CC Turn for a given deflection $\delta$ and Figure 5 compares the turn lengths for $\delta \in [0, 2\pi[$.

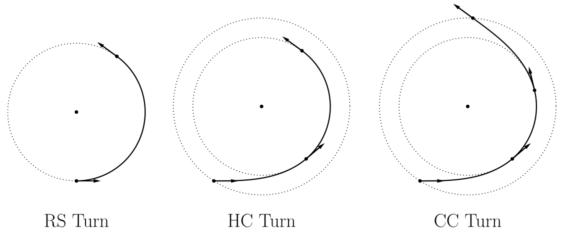

Fig. 4: Comparison of the different turns for $\delta = 0.8\pi$.

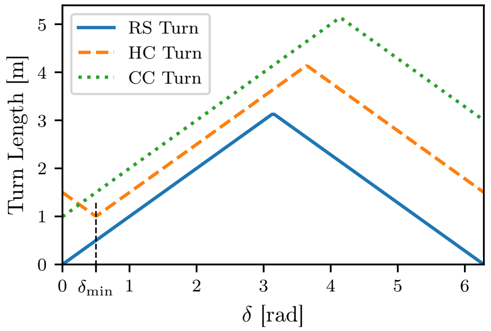

Fig. 5: Comparison of the turn lengths along a RS, HC and CC Turn with respect to the deflection $\delta$ for $\kappa_{\max} = 1\mathrm{m}^{-1}$ and $\sigma_{\max} = 1\mathrm{rad\,m}^{-1}$.

It can be observed in Figure 5 that the turn lengths are lower bounded by the RS Turn for all $\delta$. It increases linearly until it reaches its peak at $\delta = \pi$ and decreases for larger deflections again due to a change of the driving direction. The symmetric nature of CC Turns allows to use elementary paths [30] for small deflections resulting in shorter turn lengths than a HC Turn. However for $\delta < \delta_{\min}$, there exists a deflection for which a HC Turn leads to a shorter turn than a CC Turn. For $\delta \geq \delta_{\min}$, a HC Turn always results in a smaller turn length than a CC Turn because each deflection can be reached with only one clothoid and an arc instead of two clothoids and an arc. Note that the nature of the irregular HC and CC Turns avoids that the turn lengths keep growing monotonously for large deflections.

### *C. HC Paths*

According to Reeds and Shepp, the shortest path for a car can be computed by evaluating 9 path families using RS Turns and straight lines [6]. For paths, which include clothoids, there exists an infinite number of possibilities to connect two configurations [7]. Therefore, it is proposed for HC Steer to select a path with minimal length out of 13 HC Families, see Table I. They consist of the RS Families and four additional ones based on experimental results and experience [30].

TABLE I: HC Families: $C$ denotes a turn, $S$ a straight line, and $|$ a cusp.

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

HC Paths provide the flexibility to start and end with zero curvature, in the following denoted as HC$^{00}$, or with maximum curvature $\pm\kappa_{\max}$ denoted as HC$^{\pm\pm}$. Compared to HC$^{00}$, HC$^{\pm\pm}$ Steer leaves the initial and final clothoid away resulting in shorter paths, see Subsection IV-D. The next paragraph outlines the general procedure for computing the HC Families and explicitly details the computation of HC$^{00}$ Steer for the family $C|C|C$.

For every HC Family and two given configurations $\mathbf{q}_s$ and $\mathbf{q}_g$, a path can be computed in four steps: (1) Fit 4 start and 4 goal HC Circles (forwards/backwards, left/right) to the given configurations. (2) Out of the $4 \cdot 4$ possibilities to connect $\mathbf{q}_s$ and $\mathbf{q}_g$, remove the combinations that can not be realized by the corresponding family. For instance in case of $C|C|C$, the HC Circle $C_{t=1}^{d=1}(\mathbf{q}_s)$ always requires the HC Circle $C_{t=1}^{d=-1}(\mathbf{q}_g)$ at the goal configuration, see Figure 6. (3) For every resulting combination, use RS, HC, CC Turns,

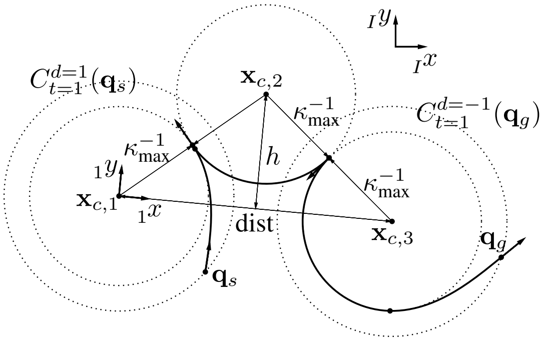

Fig. 6: Illustration of a HC$^{00}$ Path for a $C|C|C$ family and the two HC Circles $C_{t=1}^{d=1}(\mathbf{q}_s)$ and $C_{t=1}^{d=-1}(\mathbf{q}_g)$.

and tangency conditions to connect start and goal HC Circle by enforcing curvature continuity between cusps. In case of $C|C|C$, this is shown in Figure 6, where a RS Turn is used to connect the start and goal HC Circle. The center of the RS Turn $\mathbf{x}_{c,2}$ in the local frame can then be computed as

$$h = \sqrt{(4\kappa_{\max}^{-2} - \frac{1}{4}\text{dist}^2(\mathbf{x}_{c,1}, \mathbf{x}_{c,3}))}, \qquad (12)$$

$$_1\mathbf{x}_{c,2} = \begin{pmatrix} \frac{1}{2}\text{dist}(\mathbf{x}_{c,1}, \mathbf{x}_{c,3}) \\ h \end{pmatrix}, \qquad (13)$$

where the symbols correspond to Figure 6. (4) Finally select the start and goal HC Circle that results in the shortest path for the corresponding family.

### D. Comparison of RS, HC, CC Steer

This subsection evaluates and compares HC Steer with RS and CC Steer in terms of path length and computation time.

Figure 7 illustrates the relative difference in path length of HC$^{\pm\pm}$, HC$^{00}$, and CC Steer compared to RS Steer for $10^5$ random steering procedures.

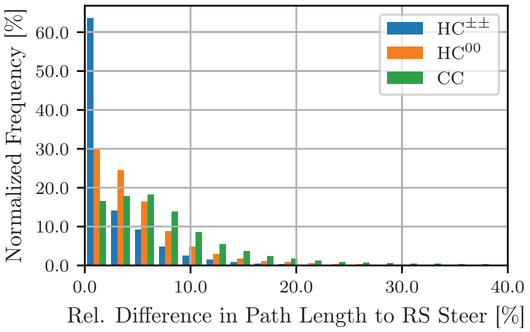

Fig. 7: Relative difference in path length of HC$^{\pm\pm}$, HC$^{00}$, and CC Steer compared to RS Steer for $10^5$ randomly sampled start and goal configurations ($\kappa_{\max} = 1\,\text{m}^{-1}$, $\sigma_{\max} = 1\,\text{rad}\,\text{m}^{-1}$).

While CC Steer generates the largest deviations from RS Steer, HC$^{\pm\pm}$ Steer generates paths that deviate in more than 60 % less than 2.5 % from the length of RS Paths. This shows HC$^{\pm\pm}$ Steer's capability of approximating RS Steer while maintaining curvature continuity between cusps. The performance of HC$^{00}$ Steer lies between HC$^{\pm\pm}$ and CC Steer.

The computation times of the different steering functions are evaluated on a single core of an Intel Xeon E5@3.50 GHz, 10 MB cache, and listed in Table II. It can be observed that the curvature continuity comes at a cost of up to 13 times longer computations. RS Paths can be computed on average in 6.9 µs. HC$^{00}$ and HC$^{\pm\pm}$ Steer perform almost equally and find solutions on average in less than 89.5 µs. The computation of CC Steer is slightly faster and takes on average 79.0 µs. Standard deviations are below 15.0 µs for all analyzed steering functions.

TABLE II: Comparison of the computation time of RS, HC$^{00}$, HC$^{\pm\pm}$, and CC Steer for $10^5$ random steering procedures.

| | computation time | |
|---|---|---|
| | mean [µs] | std [µs] |
| RS | 6.9 | ±1.3 |
| HC$^{00}$ | 89.5 | ±10.5 |
| HC$^{\pm\pm}$ | 89.1 | ±15.0 |
| CC | 79.0 | ±11.5 |

As shown in Section VI, the longer computations have a minor effect on the overall runtime when integrated into BiRRT* since most of the computation time is spent in the collision checker.

### E. Topological Admissibility

RS, HC, and CC Steer always find a connecting path for two random configurations making all three steering functions complete. In order to guarantee that these methods also result in a collision free, (probabilistically) complete path when integrated into a general motion planner like BiRRT*, they have to be topologically admissible [31], [7]:

$$\begin{aligned} &\forall \varepsilon > 0, \exists \eta > 0, \forall (\mathbf{q}_s, \mathbf{q}_g) \in \mathcal{C}^2, \\ &\mathbf{q}_g \in \mathcal{B}(\mathbf{q}_s, \eta) \Rightarrow \text{steer}(\mathbf{q}_s, \mathbf{q}_g) \subset \mathcal{B}(\mathbf{q}_s, \varepsilon), \end{aligned} \qquad (14)$$

where $\mathcal{B}(\mathbf{q}, \bullet)$ describes a ball of a given size centered at configuration $\mathbf{q}$, and $\text{steer}(\mathbf{q}_s, \mathbf{q}_g)$ denotes a steering procedure.

Equation (14) states that a steering function is topologically admissible if the computed path stays in an $\varepsilon$-neighborhood when start and goal configurations are located in an $\eta$-neighborhood. By nature, RS Steer fulfills this condition since its paths only consist of straight lines and circles without a minimal required path length. In contrast to that, CC Steer is only topologically admissible when additional so-called topological paths are introduced [7]. Similar to CC Steer, HC$^{00}$ Steer also requires additional topological paths for completeness in a general motion planner. The reason is that the clothoids at the start and goal configuration result in path lengths that are always lower bounded by $2l_{\min}$. Experiments have shown however that these topological paths are only a theoretical construct and not practical in reality. This is because the nature of the clothoids in the topological paths allows only small $\eta$-neighborhoods for a moderate $\varepsilon$ limiting the maneuverability of the vehicle significantly.

In contrast to CC and HC$^{00}$ Steer, HC$^{\pm\pm}$ Steer is topologically admissible. This is due to the fact that the families $C|C|C$ and $C|S|C$ in HC$^{\pm\pm}$ Steer only consist of circles and straight lines (HC$^{\pm\pm}$ starts and ends with maximum curvature). The absence of clothoids allows to generate paths, whose length is not lower bounded anymore. Therefore, HC$^{\pm\pm}$ Steer results in completeness when integrated into a general motion planner.

## V. Planning with BiRRT*

In order to compute a collision-free, asymptotically optimal path, BiRRT* requires a collision checker and a cost function for minimization. This section introduces the implemented collision checker in Subsection V-A and the chosen cost function in Subsection V-B.

### A. Collision Checking

Designing a fast collision checker is essential since BiRRT* spends most of its computation time checking tree extensions for collisions with the environment. Currently, it is assumed that the environment is static and that obstacle $i$, labeled as $O_i$, is given as a convex polygon. For non-convex shapes, this can be achieved by a convex decomposition [32]. Note that in the following, a calligraphic letter is used whenever a set of points occupied by an object is described.

At every discrete configuration $\mathbf{q}$ along the path, the body of the ego vehicle $A_1$ and all actuated tires $j$, labeled as $T_j$, are checked in two consecutive steps for collision with the entire environment. This setup is illustrated in Figure 8, where the body's circumscribed circle is denoted as $A_2$ and $\mathcal{A}_1, \mathcal{A}_2, \mathcal{O}_i, \mathcal{T}_j \subset \mathbb{R}^2$.

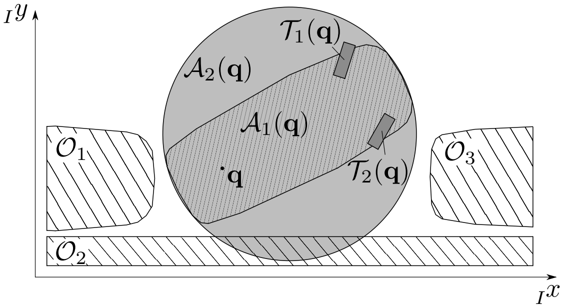
${}_I y$
$\mathcal{T}_1(\mathbf{q})$
$\mathcal{A}_2(\mathbf{q})$
$\mathcal{A}_1(\mathbf{q})$
$\mathcal{O}_1$
$\mathbf{q}$
$\mathcal{T}_2(\mathbf{q})$
$\mathcal{O}_3$
$\mathcal{O}_2$
${}_I x$

Fig. 8: Illustration of the ego vehicle's body $A_1$, the car body's circumscribed circle $A_2$, its actuated tires $T_j$ at configuration $\mathbf{q}$, and the obstacles $O_i$ in the environment.

In the first step of the collision check, obstacle $O_i$ is labeled as a collision hypothesis $H_i$ if

$$\mathcal{A}_2(\mathbf{q}) \cap \mathcal{O}_i \neq \emptyset, \tag{15}$$

where $\mathcal{A}_2(\mathbf{q})$ describes the car body's circumscribed circle at configuration $\mathbf{q}$. In the second step, all collision hypotheses are checked against the body of the car $A_1$ and its actuated tires $T_j$ at configuration $\mathbf{q}$. Collision hypothesis $H_i$ is collision-free if

$$\mathcal{A}_1(\mathbf{q}) \cap \mathcal{H}_i = \emptyset \;\wedge\; \mathcal{T}_j(\mathbf{q}) \cap \mathcal{H}_i = \emptyset,\ \forall j. \tag{16}$$

We use the Gilbert-Johnson-Keerthi (GJK) algorithm [33] to perform the second step of the collision check. The GJK algorithm takes on average 700 ns for a binary collision check of two polygons each consisting of 23 vertices on a single core of an Intel Xeon E5@3.50 GHz, 10 MB cache. It is also capable of computing the minimal distance between two polygons if they are not colliding, which takes on average additional 300 ns.

### B. Cost Function

The cost function $J$ is evaluated along the path, which is given by $N$ segments, see Figure 9. It consists of four terms, computes a positive cost for every non-trivial path [34], and is given as

$$J = \mathbf{w}_J^T \begin{pmatrix} J_{\text{length}} \\ J_{\text{cusp}} \\ J_{\text{curv}} \\ J_{\text{obs}} \end{pmatrix}, \tag{17}$$

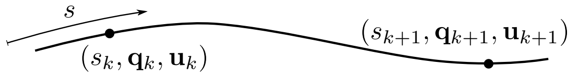
$s$
$(s_k, \mathbf{q}_k, \mathbf{u}_k)$
$(s_{k+1}, \mathbf{q}_{k+1}, \mathbf{u}_{k+1})$

Fig. 9: Illustration of the $k^{\text{th}}$ path segment connecting configuration $\mathbf{q}_k$ and $\mathbf{q}_{k+1}$. The arc length along the path is described by $s$ and $\mathbf{u}_k, \mathbf{u}_{k+1}$ denote the inputs at distance $s_k$ and $s_{k+1}$, respectively.

where $\mathbf{w}_J$ allows to weight each cost term, $J_{\text{length}}$ penalizes the length of the path, $J_{\text{cusp}}$ punishes cusps in the path, $J_{\text{curv}}$ makes uncomfortable paths in terms of curvature more expensive, and $J_{\text{obs}}$ puts a cost on paths with little clearance to static obstacles.

The cost terms are computed as

$$J_{\text{length}} = \int_{s_0}^{s_N} ds, \tag{18}$$

$$J_{\text{cusp}} = \sum_{k=0}^{N-1} \mathbb{1}_{d_{k+1} \cdot d_k < 0}, \tag{19}$$

$$J_{\text{curv}} = (\kappa_{\max}(s_N - s_0))^{-1} \int_{s_0}^{s_N} |\kappa(s)| ds, \tag{20}$$

$$J_{\text{obs}} = 1 - \min_{s \in [s_0, s_N]} (d_{\text{obs}}(s), d_{\text{safety,1}}) \Big/ d_{\text{safety,1}}, \tag{21}$$

where the driving direction at distance $s_k$ is denoted as $d_k$, the minimal clearance along the path as $d_{\text{obs}}$, and the soft safety distance as $d_{\text{safety,1}}$. Equation (18) computes the length of the path while equation (19) counts the cusps in the path. Equation (20) integrates and normalizes the curvature along the path, and equation (21) compares the minimal clearance of the path with the soft safety distance, also see [15].

## VI. Experiments

The experimental results analyze and compare the performance of BiRRT* with RS, HC$^{\pm\pm}$, and CC[1] Steer on two different HDP scenarios.[2] Figure 10 visualizes the two scenarios.

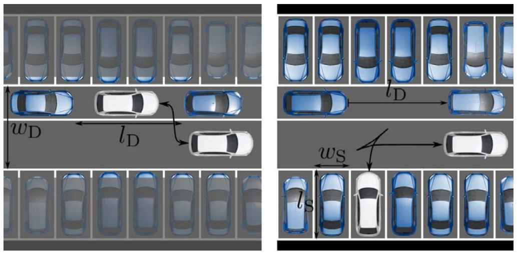

Fig. 10: Scenario I (left) and scenario II (right), where the gray shaded area marks the region that is excluded from the sampling region of BiRRT*.

The setup used in the experiments is described in Subsection VI-A, and Subsection VI-B discusses the results.

### A. Setup

The width of the driveway is given by $w_{\text{D}} = 5.5\,\text{m}$, the width and length of a parking spot by $w_{\text{S}} = 2.5\,\text{m}$ and $l_{\text{S}} =$

[1] CC Steer is used without topological paths as explained in Section IV-E.

[2] This link `https://youtu.be/RlZZ4jnEhTM` provides a video of the results, and the source code of the steering functions is available at `https://github.com/hbanzhaf/steering_functions`.

5 m according to the national standards in Germany [35]. In order to analyze how the proposed motion planner performs in each scenario, the experiments incrementally increase the length $l_\mathrm{D}$ in the driveway.

As it can be seen in Figure 1, the vehicles in the environment, which consist of commercially available mid-size and full-size cars, are given by their convex hull. Each is inflated by a hard safety distance of $d_{\text{safety},2} = 10\,\mathrm{cm}$. The hard safety distance shrinks the available space for maneuvering since $l_\mathrm{D}$ describes the actual distance between the car bodies. An additional 20 cm is added as a soft safety distance $d_{\text{safety},1}$, see Subsection V-B.

The ego vehicle's parameters are listed in Table III, where $\kappa_\mathrm{max}$ and $\sigma_\mathrm{max}$ already include 10 % control reserve. The max. sharpness is given at a longitudinal velocity of $1\,\mathrm{m\,s^{-1}}$.

TABLE III: Vehicle parameters

| Parameter | Symbol | Value |
|---|---|---|
| Length | - | 4.926 m |
| Width | - | 2.086 m |
| Wheel Base | $L$ | 2.912 m |
| Max. Curvature | $\kappa_\mathrm{max}$ | $1/4.994\,\mathrm{m}^{-1}$ |
| Max. Sharpness | $\sigma_\mathrm{max}$ | $0.315\,\mathrm{rad\,m}^{-1}$ |

The planner BiRRT* is executed for 6 s by uniformly sampling configurations in the operating region, which can be seen in Figure 10 ($21\,\mathrm{m} \times 5.5\,\mathrm{m}$ in scenario I, $21\,\mathrm{m} \times 18\,\mathrm{m}$ in scenario II, and $[0, 2\pi[$ for the heading angle in both scenarios). A goal sampling frequency of 5 % is applied, collision checks are performed every 10 cm, and the constant $\gamma$ [26] is set to 6.0. To mitigate randomization effects, every experiment is repeated 100 times with the same setup. The extend procedures in BiRRT* are selected as RS, CC, and HC$^{\pm\pm}$ Steer due to its topological admissibility.

The proposed motion planner is implemented as ROS node in C++ based on [36] and executed on a single core of an Intel Xeon E5@3.50 GHz, 10 MB cache.

### B. Results

The results with regard to $l_\mathrm{D}$, namely the time to first solution $t_\mathrm{TTFS}$, the number of curvature discontinuities, the number of cusps, the path length, and the success rate of finding a solution after 100 repetitions of the same experiment, are given in Table IV. Note that BiRRT* grows a tree from start to goal and vice versa and therefore performs equally for maneuvering in and out of the parking spot.

Based on Table IV, Figure 11 illustrates the time to first solution $t_\mathrm{TTFS}$ and the success rate relating to $l_\mathrm{D}$ in scenario I.

BiRRT* with RS and HC$^{\pm\pm}$ Steer finds solutions for $l_\mathrm{D} \geq 5.6\,\mathrm{m}$ in scenario I while CC Steer requires an additional 80 cm. In scenario II, solutions are generated with RS and HC$^{\pm\pm}$ Steer for $l_\mathrm{D} \geq 4.0\,\mathrm{m}$, and $l_\mathrm{D} \geq 4.6\,\mathrm{m}$ for CC Steer.

Overall the average time to first solution, the number of curvature discontinuities, and the number of cusps decrease when $l_\mathrm{D}$ is increased while the success rate raises, also see Figure 11. RS Steer mostly results in slightly faster solutions with less cusps and higher success rates than HC$^{\pm\pm}$ Steer. However, Table IV shows that RS Steer's paths suffer from more than twice as many curvature discontinuities than HC$^{\pm\pm}$ Steer's paths because HC$^{\pm\pm}$ Steer enforces $\kappa$ to be continuous between cusps, see Figure 12. Consequently, the resulting paths of HC$^{\pm\pm}$ Steer are more comfortable to drive and easier to be tracked by a controller than the curvature discontinuous paths of RS Steer.

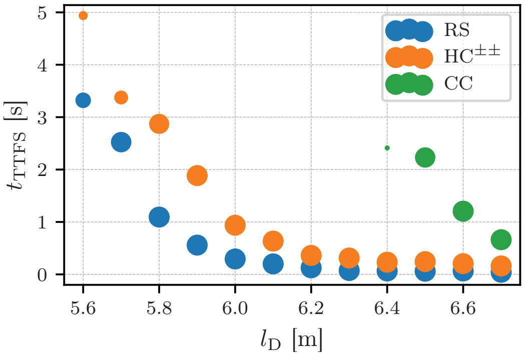

Fig. 11: Comparison of BiRRT*'s performance with RS, HC$^{\pm\pm}$, and CC Steer by measuring the time to first solution $t_\mathrm{TTFS}$ with respect to $l_\mathrm{D}$ in scenario I. The size of the visualized markers indicates the success rate in finding a path within 6 s.

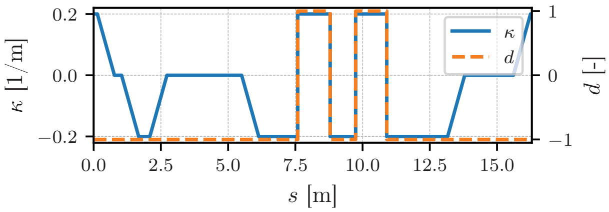

Fig. 12: Curvature $\kappa$ along a solution path in scenario II when maneuvering into the parking spot ($l_\mathrm{D} = 4.8\,\mathrm{m}$) with BiRRT* and HC$^{\pm\pm}$ Steer. The driving direction at a given arc length $s$ is given by $d$.

Compared to CC Steer, HC$^{\pm\pm}$ Steer computes solutions faster, leaving BiRRT* more time for optimizing the initially generated path. Additionally, it outperforms CC Steer in terms of number of cusps and success rate.

## VII. Conclusion and Outlook

In this paper, a novel extend function called Hybrid Curvature (HC) Steer for sampling-based nonholonomic motion planning in tight environments is introduced. HC Steer approximates Reeds-Shepp's paths in terms of path length, but enforces curvature continuity between cusps resulting in directly drivable paths for nonholonomic systems. Experiments in tight environments with the optimal motion planner BiRRT* show HC Steer's advantage of computing smoother paths than RS Steer in equally challenging environments. Compared to CC Steer, HC Steer with BiRRT* finds solutions with less direction switches, shorter planning time, and higher success rates. Hence HC Steer clearly outperforms both RS and CC Steer from a practical point of view.

In the future, we aim to combine the proposed HC$^{\pm\pm}$ steering function with HC$^{0\pm}$ and HC$^{\pm 0}$ Steer, which start or end with either zero or maximal curvature. Such a combination would allow to generate paths with preferably zero

TABLE IV: Results of BiRRT* in Scenario I/II after 6 s of sampling time and 100 repetitions of the same experiment. The time to first solution $t_{\text{TTFS}}$, the number of curvature discontinuities, the number of cusps, the path lengths, and success rates are listed with respect to $l_{\text{D}}$. Mean path lengths are rounded.

| | | $t_{\text{TTFS}}$ (mean ± std) [s] | | | #curv. discon. (mean ± std) [–] | | | #cusps (mean ± std) [–] | | | length (mean ± std) [m] | | | success rate [%] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $l_{\text{D}}$ [m] | RS | HC$^{\pm\pm}$ | CC | RS | HC$^{\pm\pm}$ | CC | RS | HC$^{\pm\pm}$ | CC | RS | HC$^{\pm\pm}$ | CC | RS | HC$^{\pm\pm}$ | CC |
| Scenario I | 5.4 | - | - | - | - | - | - | - | - | - | - | - | - | 0 | 0 | 0 |
| | 5.6 | 3.3±1.4 | 4.9±0.9 | - | 12.5±2.7 | 6.8±1.8 | - | 6.4±1.5 | 9.1±2.1 | - | 14±1.7 | 17±3.8 | - | 50 | 13 | 0 |
| | 5.8 | 1.1±1.0 | 2.9±1.6 | - | 10.2±3.1 | 4.6±1.5 | - | 4.6±1.4 | 6.7±2.5 | - | 14±3.5 | 17±3.9 | - | 100 | 90 | 0 |
| | 6.0 | 0.3±0.2 | 0.9±0.7 | - | 8.3±2.3 | 3.2±1.1 | - | 3.4±1.0 | 4.7±1.8 | - | 13±1.5 | 15±2.3 | - | 100 | 100 | 0 |
| | 6.2 | 0.1±0.1 | 0.4±0.3 | - | 7.7±2.4 | 2.3±1.2 | - | 2.3±1.1 | 3.5±1.7 | - | 14±2.1 | 15±2.8 | - | 100 | 100 | 0 |
| | 6.4 | 0.1±0.1 | 0.2±0.2 | 2.4±0.7 | 6.7±2.1 | 1.8±0.9 | 0±0 | 1.7±0.9 | 2.6±1.3 | 4.5±0.5 | 13±1.3 | 14±2.3 | 16±0.1 | 100 | 100 | 2 |
| | 6.6 | 0.1±0.1 | 0.2±0.2 | 1.2±0.8 | 6.5±1.9 | 1.5±0.9 | 0±0 | 1.5±0.8 | 2.2±1.5 | 4.7±1.7 | 13±1.5 | 14±3.3 | 18±3.1 | 100 | 100 | 100 |
| Scenario II | 3.8 | - | - | - | - | - | - | - | - | - | - | - | - | 0 | 0 | 0 |
| | 4.0 | 2.7±1.4 | 5.1±0.7 | - | 13.4±3.6 | 4.0±0.0 | - | 6.2±1.8 | 5.0±0.0 | - | 18±3.3 | 21±0.9 | - | 12 | 2 | 0 |
| | 4.2 | 2.5±1.5 | 3.6±1.7 | - | 12.9±2.6 | 4.4±0.7 | - | 5.7±1.4 | 6.5±2.1 | - | 18±3.2 | 23±2.0 | - | 50 | 13 | 0 |
| | 4.4 | 1.9±1.3 | 2.6±1.6 | - | 11.8±2.4 | 3.2±1.0 | - | 4.6±1.4 | 5.5±1.8 | - | 18±4.9 | 21±4.4 | - | 85 | 51 | 0 |
| | 4.6 | 1.3±1.3 | 2.2±1.5 | 3.1±1.7 | 10.4±2.3 | 2.9±1.3 | 0±0 | 3.6±1.5 | 4.9±2.0 | 6.5±1.7 | 17±2.8 | 22±3.1 | 20±3.0 | 95 | 75 | 4 |
| | 4.8 | 0.6±0.6 | 1.7±1.2 | 3.4±2.0 | 9.6±2.1 | 2.5±0.9 | 0±0 | 3.1±1.2 | 4.1±1.3 | 5.1±1.5 | 16±1.9 | 21±4.0 | 19±2.5 | 100 | 97 | 21 |
| | 5.0 | 0.5±0.5 | 1.0±1.0 | 3.2±1.6 | 9.3±1.8 | 2.4±0.8 | 0±0 | 3.0±1.1 | 4.0±1.6 | 5.2±2.0 | 16±2.1 | 21±3.4 | 20±5.4 | 100 | 98 | 64 |
| | 5.2 | 0.3±0.2 | 0.8±0.7 | 1.6±1.4 | 8.5±1.9 | 2.3±0.8 | 0±0 | 2.5±0.9 | 3.7±1.2 | 4.1±1.6 | 16±1.5 | 20±3.2 | 18±2.9 | 100 | 100 | 92 |
| | 5.4 | 0.2±0.2 | 0.6±0.5 | 1.0±0.9 | 8.5±1.6 | 2.1±0.7 | 0±0 | 2.3±0.7 | 3.2±1.0 | 3.2±1.3 | 15±1.1 | 19±3.2 | 17±2.4 | 100 | 100 | 100 |

curvature at the beginning and at the end while still providing the desirable characteristics of HC$^{\pm\pm}$ Steer in between.

## References

[1] (2017) Bosch and Daimler demonstrate driverless parking in real-life conditions. Robert Bosch GmbH. (visited on 2017/07/26). [Online]. Available: http://http://www.bosch-presse.de/pressportal/de/en/news/

[2] H. Banzhaf *et al.*, "The Future of Parking: A Survey on Automated Valet Parking with an Outlook on High Density Parking," in *IEEE Intelligent Vehicles Symposium*, 2017.

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

[4] G. Schildbach and F. Borrelli, "A Dynamic Programming Approach for Nonholonomic Vehicle Maneuvering in Tight Environments," in *IEEE Intelligent Vehicles Symposium*, 2016.

[5] D. Kim *et al.*, "Practical motion planning for car-parking control in narrow environment," *IET Control Theory & Applications*, 2010.

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

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

[8] M. Jordan and A. Perez, "Optimal Bidirectional Rapidly-Exploring Random Trees," *CSAIL, MIT, Cambridge, MA, TR 2013-021*, 2013.

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

[10] B. Paden *et al.*, "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, 2016.

[11] M. Likhachev and D. Ferguson, "Planning Long Dynamically-Feasible Maneuvers for Autonomous Vehicles," *The International Journal of Robotics Research*, 2009.

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

[13] C. Siedentop *et al.*, "Path-Planning for Autonomous Parking with Dubins Curves," in *Uni-DAS Workshop Fahrerassistenzsysteme*, 2015.

[14] J. Ziegler, M. Werling, and J. Schröder, "Navigating car-like Robots in unstructured Environments using an Obstacle sensitive Cost Function," in *IEEE Intelligent Vehicles Symposium*, 2008.

[15] U. Schwesinger *et al.*, "Vision-Only Fully Automated Driving in Dynamic Mixed-Traffic Scenarios," *it-Information Technology*, 2015.

[16] J. hwan Jeon *et al.*, "Anytime Computation of Time-Optimal Off-Road Vehicle Maneuvers using the RRT*," in *IEEE Conference on Decision and Control and European Control Conference*, 2011.

[17] J. hwan Jeon *et al.*, "Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving," in *IEEE American Control Conference*, 2013.

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

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

[20] H. Vorobieva *et al.*, "Automatic Parallel Parking with Geometric Continuous-Curvature Path Planning," in *IEEE Intelligent Vehicles Symposium*, 2014.

[21] J. Schulman *et al.*, "Motion Planning with Sequential Convex Optimization and Convex Collision Checking," *The International Journal of Robotics Research*, 2014.

[22] P. Zips *et al.*, "Optimisation based path planning for car parking in narrow environments," *Robotics and Autonomous Systems*, 2016.

[23] J. Ziegler *et al.*, "Trajectory Planning for Bertha - a Local, Continuous Method," in *IEEE Intelligent Vehicles Symposium*, 2014.

[24] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Tech. Rep., 1998.

[25] Y. Kuwata *et al.*, "Real-time Motion Planning with Applications to Autonomous Urban Driving," *IEEE Transactions on Control Systems Technology*, 2009.

[26] S. Karaman and E. Frazzoli, "Optimal Kinodynamic Motion Planning using Incremental Sampling-based Methods," in *IEEE Conference on Decision and Control*, 2010.

[27] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically Optimal Motion Planning for Robots with Linear Dynamics," in *IEEE International Conference on Robotics and Automation*, 2013.

[28] 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.

[29] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science, 2011.

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

[31] 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.

[32] C. Ericson, *Real-Time Collision Detection*. CRC Press, 2004.

[33] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space," *IEEE Journal on Robotics and Automation*, 1988.

[34] S. Karaman and E. Frazzoli, "Sampling-based Optimal Motion Planning for Non-holonomic Dynamical Systems," in *IEEE International Conference on Robotics and Automation*, 2013.

[35] "Verordnung des Ministeriums für Verkehr und Infrastruktur über Garagen und Stellplätze," §4 Stellplätze und Fahrgassen, Frauenparkplätze, Ministerium für Verkehr und Infrastruktur, 1997.

[36] S. Karaman. (2017) SMP Library. (visited on 2017/03/24). [Online]. Available: https://svn.csail.mit.edu/smp