Trajectory Generation using Sharpness Continuous Dubins-like Paths with Applications in Control of Heavy-Duty Vehicles

Rui Oliveira[1], Pedro F. Lima[1], Marcello Cirillo[2], Jonas Mårtensson[1] and Bo Wahlberg[1]

Abstract— We present a trajectory generation framework for control of wheeled vehicles under steering actuator constraints. The motivation is smooth driving of autonomous heavy-duty vehicles, which are characterized by slow actuator dynamics. In order to deal with the slow dynamics, we take into account rate and, additionally, torque limitations of the steering actuator directly. Previous methods only take into account limitations in the path curvature, which deals indirectly with steering rate limitations. We propose the new concept of Sharpness Continuous curves, which uses cubic curvature paths together with circular arcs to steer the vehicle. The obtained paths are characterized by a smooth and continuously differentiable steering angle profile. The final trajectories computed with our method provide low-level controllers with reference signals which are easier to track, resulting in improved performance. The smoothness of the obtained steering profiles also results in increased passenger comfort. The method is characterized by fast computation times. We detail possible path planning applications of the method, and conduct simulations that show its advantages and real-time capabilities.

I. INTRODUCTION

A. Background and Motivation

Path planning deals with the generation of paths or trajectories (paths with an associated time law) for a vehicle. The generated paths/trajectories are used as a reference signal for the controllers implemented in the vehicle. Planning methods for autonomous vehicles have come a long way from the initial problem of finding collision free paths, being now focused on properties such as kinodynamic constraints, optimality, and uncertainty [1], [2]. Car-like vehicles must follow specific patterns of motion defined by their kinematic constraints. These constraints introduce an additional difficulty, as they limit the maneuverability of the vehicle, resulting in limited types of paths that are admissible, *i.e.*, can be feasibly followed. Heavy-duty vehicles introduce additional constraints due to their slow actuator dynamics. Thus planning methods for autonomous vehicles must be adapted, in order to deal with the additional control challenges imposed by heavy-duty vehicles.

Much research effort has been devoted to the field of $\mathbf{G}^3$ path planning. A $\mathbf{G}^3$ path is characterized by a continuously differentiable curvature profile. $\mathbf{G}^3$ paths are important as they avoid jerky motion and wheel slippage [3], simplifying

the tracking task and improving controller performance [4]. $\mathbf{G}^3$ path planning is analogous to the $\mathbf{G}^3$ interpolation problem with applications often related to Computer-Aided Design [5]. Some authors have focused on its applications to autonomous mobile robots [6].

Path planning for autonomous vehicles introduces additional demands. Besides the $\mathbf{G}^3$ property, another important property of a path is its length. Shorter paths are desirable as they result in more efficient driving.

Steering methods are a class of path planners that are able to efficiently compute a path between vehicle states in an environment without obstacles. Even though the majority of autonomous vehicle applications considers obstacles, steering methods still prove to be useful, as they are often used as components of more complex path planners which can take obstacles into account [7]–[10].

Dubins [11] and Reeds-Shepp [12] paths are steering methods that connect two arbitrary vehicle poses through a minimal length path. However, these paths have discontinuous curvatures, thus not being $\mathbf{G}^3$. [13] makes use of clothoidal paths to generate near length optimal curvature continuous paths. Clothoids are an obvious choice, since they have long been used in road design, as they allow for smooth driving [14]. However, the curvature derivative is discontinuous, resulting in paths that are not $\mathbf{G}^3$. [15] extends [11], so that $\mathbf{G}^3$ paths with near optimal length are planned. This last method deals with limitations of the path curvature and curvature derivative, which are not directly related to the steering actuator limitations of the vehicle.

B. Main Contributions

The contribution of this work comes from the generation of vehicle trajectories that:

- Directly take into account steering actuator magnitude, rate, and acceleration limitations, generating $\mathbf{G}^3$ paths;
- Ease the controller task and improve passenger comfort;
- Can connect arbitrary vehicle configurations;
- Have fast computation times.

We build upon the work of [13], replacing clothoids with cubic curvature paths. Additionally, we formulate the constraints so that we can limit the actual steering angle rate and steering angle acceleration of the vehicle, instead of the curvature derivative of the path, as done in previous approaches [13], [15]. This is because the maximum steering angle rate and acceleration are more intuitive constraints that can be directly obtained from the vehicle actuator limitations. The proposed method is also computationally fast and can be used online, as part of a more complex motion planner [16].

C. Outline

Section II introduces the vehicle model used and defines the problem we address. Section III presents Sharpness Continuous paths used to solve the stated problem. Section IV presents Cubic Curvature paths, a building block of Sharpness Continuous paths. Section V illustrates our simulation results, showing the performance of the method. Conclusions and future work are presented in Section VI.

II. PROBLEM STATEMENT

A. Vehicle Model

We start by defining the vehicle model as:

$$\dot{x} = v\cos\theta, \qquad \dot{y} = v\sin\theta, \qquad \dot{\theta} = v\kappa.$$

(x, y) represents the location of the vehicle rear wheel axle center, θ its orientation and v is the vehicle velocity. The curvature κ of a vehicle with wheelbase length L is related to its steering angle ϕ through

$$\kappa = \tan(\phi)/L. \tag{1}$$

A vehicle pose is defined by the three variables (x, y, θ). If an additional curvature is associated to a pose we obtain a configuration, defined as (x, y, θ, κ).

The steering angle of the vehicle is set by an actuator, which like any real system has physical limitations. The limitations with which we comply in this work are:

- Maximum steering angle amplitude $\phi_{\max}$,
- Maximum steering angle rate of change $\dot{\phi}_{\max}$,
- Maximum steering actuation acceleration $\ddot{\phi}_{\max}$.

These limitations effectively affect the vehicle motion capabilities and should be dealt with when generating paths.

B. Path Feasibility

Path feasibility depends on the capabilities of the vehicle that executes it and on the path itself. The limited steering angle amplitude $\phi_{\max}$ imposes a maximum allowed curvature on the path $\kappa_{\max}$. This limitation is addressed by generating paths which have a curvature profile $|\kappa| \leq \kappa_{\max}$ [11], [12]. Limited steering angle rate of change $\dot{\phi}_{\max}$ can be tackled by limiting the curvature derivative of the generated paths [13].

In this paper, we deal with the third limitation, related to the limited steering angle acceleration $\ddot{\phi}_{\max}$. Having a limited $\ddot{\phi}_{\max}$ results in $\dot{\phi}$ being a continuous function, which in turn indicates that ϕ is a continuously differentiable, $\mathbf{C^1}$ function. The paths generated by [13] have corresponding $\dot{\phi}$ profiles with discontinuities, that require an infinite $\ddot{\phi}_{\max}$. This is impossible to achieve by an actuator, and motivates the usage of paths with a $\mathbf{C^1}$ steering profile.

The steering profile is related to the curvature profile through (1). The sharpness α is defined as the change of curvature along the path length s:

$$\alpha = \partial\kappa/\partial s.$$

By ensuring sharpness continuity in a path, we guarantee that the curvature, and the steering profile of such a path is
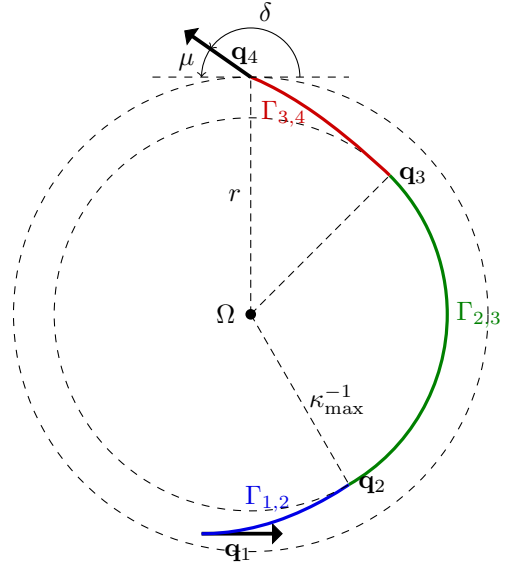

Fig. 1: Sharpness continuous turn general case.

$\mathbf{C^1}$, *i.e.*, the path is $\mathbf{G^3}$. A vehicle is thus able to follow the path using a bounded steering acceleration $\ddot{\phi}_{\max}$.

In the following section, we detail how to generate paths that respect all three limitations previously stated.

III. SHARPNESS CONTINUOUS PATHS

In this section we present the Sharpness Continuous (SC) paths. III-A introduces the principle behind SC paths. SC paths are composed of SC turns (detailed in III-B) connected over a line segment. The process of connecting SC turns over a line segment to form a continuous SC path is detailed in III-C. When generating an SC path there is a total of 16 possible combinations of different SC turns that can be used. III-D indicates how to choose the best combination.

A. Principle

[11], [12] use a combination of arc circle turns and/or line segments to connect two arbitrary poses. In [13], this idea is extended with Curvature Continuous (CC) turns, which replace arc circles by a combination of clothoid and arc circles. We extend this further, replacing the clothoid segments by cubic curvature paths, achieving sharpness continuity and respecting the limited steering acceleration $\ddot{\phi}_{\max}$.

B. Sharpness Continuous Turns

We propose Sharpness Continuous (SC) turns, which consist of three segments, an initial cubic curvature path $\Gamma_{1,2}$, a circular arc $\Gamma_{2,3}$, and a final cubic curvature path $\Gamma_{3,4}$. Figure 1 shows an example of an SC turn. The initial segment $\Gamma_{1,2}$ starts at a configuration $\mathbf{q}_1 = (x_1, y_1, \theta_1, \kappa_1)$ and ends with maximum curvature, $\pm\kappa_{\max}$, at a configuration $\mathbf{q}_2 = (x_2, y_2, \theta_2, \kappa_2)$. The second segment is a circular arc $\Gamma_{2,3}$ with radius $\kappa_{\max}^{-1}$ and arbitrary arc length, starting at $\mathbf{q}_2$ and ending at $\mathbf{q}_3 = (x_3, y_3, \theta_3, \kappa_3)$. The SC turn is completed with a path $\Gamma_{3,4}$, with starting curvature $\pm\kappa_{\max}$ and ending at a configuration $\mathbf{q}_4 = (x_4, y_4, \theta_4, \kappa_4)$.

We assume, without loss of generality, that the vehicle, and subsequently the path, starts at a configuration $\mathbf{q}_1 = (0, 0, 0, 0)$. From $\mathbf{q}_1$, it then follows the path $\Gamma_{1,2}$ taking it to a configuration $\mathbf{q}_2 = (x_2, y_2, \theta_2, \kappa_{\max})$. The path $\Gamma_{1,2}$ has initial and final curvatures 0 and $\kappa_{\max}$, respectively. The values x_2, y_2, and θ_2 are those that result from following the curvature profile of $\Gamma_{1,2}$ with a starting vehicle state $\mathbf{q}_1$.

Once the vehicle has a curvature $\kappa_{\max}$, it then follows a circular arc path $\Gamma_{2,3}$ with radius $\kappa_{\max}^{-1}$. The circular arc starts at (x_2, y_2) and has its center at a distance $\kappa_{\max}^{-1}$ perpendicular to the orientation θ_2 at point (x_2, y_2). Its center is given by

$$(x_\Omega, y_\Omega) = (x_2 - \kappa_{\max}^{-1} \sin \theta_2, y_2 + \kappa_{\max}^{-1} \cos \theta_2). \quad (2)$$

The last path segment $\Gamma_{3,4}$ departs from the circular arc and it brings the vehicle to a configuration $\mathbf{q}_4$. Configuration $\mathbf{q}_4$ depends on the point of departure from the circular arc, $\mathbf{q}_3$. However, it always lies in a circle Ω, which has the same center as the circular arc (x_Ω, y_Ω) in (2).

In order to find the radius of circle Ω, we first assume an auxiliary circular arc to be centered at $(x_{\Omega'}, y_{\Omega'}) = (0, \kappa_{\max}^{-1})$. We assume a departure configuration from the circle at $(0, 0, 0, \kappa_{\max})$. Then, by following the path given by a curvature profile with initial and final curvatures $\kappa_{\max}$ and κ_4, we will end at a configuration $\mathbf{q}_4 = (x_4, y_4, \theta_4, \kappa_4)$. $\mathbf{q}_4$ is a configuration located at an auxiliary Ω' circle (the auxiliary equivalent of the Ω circle), that has the same center as the circular arc. Thus we compute the radius of Ω', which is equal to the radius of Ω, as

$$r = \sqrt{(x_4 - x_{\Omega'})^2 + (y_4 - y_{\Omega'})^2} = \sqrt{x_4^2 + (y_4 - \kappa_{\max}^{-1})^2}.$$

An additional angle μ is defined as the difference between θ_4 and the tangential angle to Ω at configuration $\mathbf{q}_4$. It is computed using the previous auxiliary circular arc as

$$\mu = \arctan\left(\frac{y_4 - \kappa_{\max}^{-1}}{x_4}\right) + \frac{\pi}{2} - \theta_4. \quad (3)$$

Thus, given a certain initial configuration $\mathbf{q}_1$, the possible positions of the ending configuration $\mathbf{q}_4$, resulting from a combination of a cubic curvature path, a circular arc, and another cubic curvature path, *i.e.*, an SC turn, lie on a circle Ω. The possible θ_4 orientations of these configurations are given by the tangential angle at the circle plus μ.

C. Connecting Sharpness Continuous Turns

An SC path between start and goal configurations $\mathbf{q}_s$ and $\mathbf{q}_g$ can be found by connecting two SC turns. An SC path consists of three elements:

- an SC turn starting at the start configuration $\mathbf{q}_s$ and ending at a configuration $\mathbf{q}_a$ with null curvature,
- a line segment starting at $\mathbf{q}_a$ and ending at $\mathbf{q}_b$,
- an SC turn starting at configuration $\mathbf{q}_b$ with null curvature, and ending at the goal configuration $\mathbf{q}_g$.

Figure 2 shows an example of an SC path, with the three elements described above.

In order to connect two SC turns, we need to find the configurations $\mathbf{q}_a$ and $\mathbf{q}_b$ that belong to the starting and
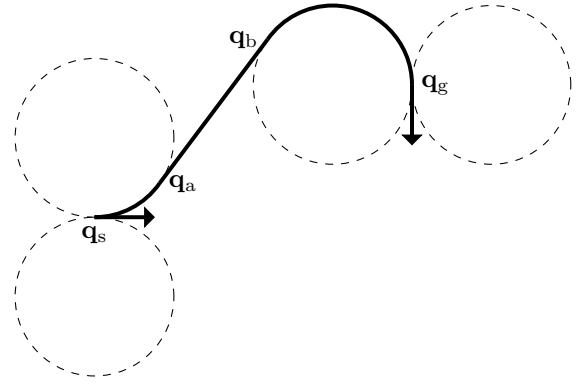

Fig. 2: Sharpness continuous path example. The path consists of an SC turn between $\mathbf{q}_s$ and $\mathbf{q}_a$, a line segment from $\mathbf{q}_a$ to $\mathbf{q}_b$, and an SC turn between $\mathbf{q}_b$ and $\mathbf{q}_g$. The dashed circles correspond to the SC turns that can span from $\mathbf{q}_a$ and $\mathbf{q}_b$.
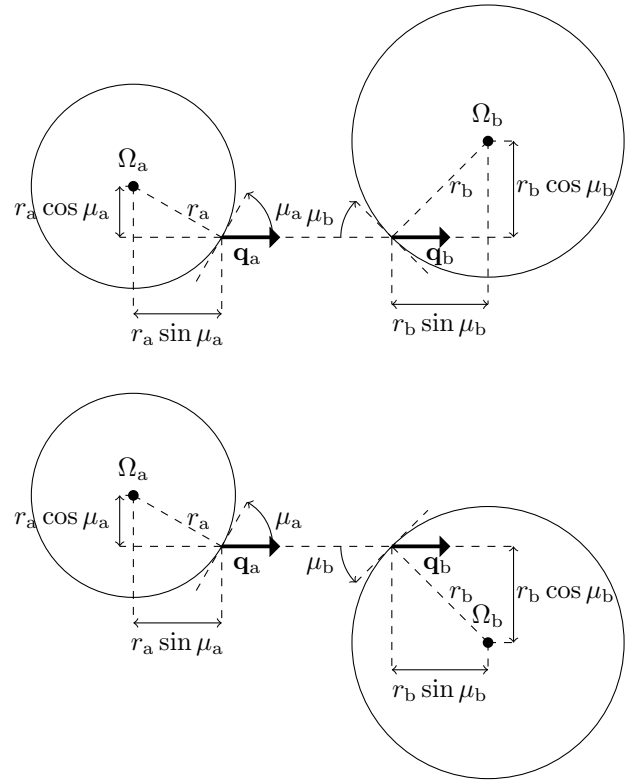

Fig. 3: Computing the external (top) and internal (bottom) tangents between two circles.

ending SC turn possible departure configurations, and that can be connected with a line segment. That is, $\mathbf{q}_a$ and $\mathbf{q}_b$ must have the same orientation, *i.e.*, $\theta_a = \theta_b$. Furthermore both must lie on a line segment with an inclination angle θ_a.

As seen before, the possible set of departure configurations of an SC turn are located in a circle, and its orientations differ from the circle tangent by μ. Thus, to connect two SC turns, we need a way to connect two circles Ω_s and Ω_f with arbitrary centers, radii, and μ values.

We first assume two auxiliary circles Ω_a and Ω_b, as depicted in Figure 3 (top). Ω_a and Ω_b have the same radii and μ values as the original circles Ω_s and Ω_f. Ω_a is centered

at $(0,0)$ and Ω_b is located so that $\mathbf{q}_a$ and $\mathbf{q}_b$ are collinear. We are interested in finding the center of $\Omega_b = (x_{\Omega_b}, y_{\Omega_b})$. From Figure 3 (top) it can be seen that

$$y_b = -r_a \cos \mu_a + r_b \cos \mu_b. \tag{4}$$

We assume that the distance $r(\Omega_a, \Omega_b)$ between the circle centers is the same as the distance between the original circles $r(\Omega_s, \Omega_f)$. We then have

$$x_{\Omega_b} = \sqrt{r(\Omega_s, \Omega_f)^2 - y_{\Omega_b}^2}. \tag{5}$$

We know that $\mathbf{q}_a = (r_a \sin \mu_a, -r_a \cos \mu_a, 0, 0)$ and that $\mathbf{q}_b = (x_{\Omega_b} - r_b \sin \mu_b, y_{\Omega_b} - r_b \cos \mu_b, 0, 0)$. To find these configurations in the original circles Ω_s and Ω_f, we need to first apply a rotation $\Delta_\theta = \arctan(y_{\Omega_f} - y_{\Omega_s}, x_{\Omega_f} - x_{\Omega_s})$ to $\mathbf{q}_a$ and $\mathbf{q}_b$. We then translate these configurations by $(\Delta_x, \Delta_y) = (x_{\Omega_s}, y_{\Omega_s})$. The resulting rotated and translated configurations correspond to the desired tangent configurations between the circles Ω_s and Ω_f.

The above procedure finds the departure configurations between two counter clockwise (left steering) SC turns, shown in Figure 3 (top). An analogous procedure can be used to find the possible departure configurations between any combination of clockwise (right steering) and counter clockwise turns, as shown in Figure 3 (bottom). This procedures are valid if the found tangent configurations $\mathbf{q}_a$ and $\mathbf{q}_b$ do not lie inside the circles Ω_b and Ω_a, respectively.

D. Finding the Shortest SC Path

In order to find the shortest SC path between two configurations $\mathbf{q}_s$ and $\mathbf{q}_g$, we need to compute all the possible SC turns that can be spanned from these configurations. The SC turns are then connected, in order to generate possible SC paths. The process is detailed below.

Each of the configurations $\mathbf{q}_s$ and $\mathbf{q}_g$ can span a total of four SC turns, depending if the vehicle is moving forwards or backwards, or if it is turning left or right. The possible SC turns that span from $\mathbf{q}_s$ and $\mathbf{q}_g$ are shown in Figure 2 as dashed circles (forward and backward SC turns are equivalent, so they lie on top of each other). Figure 1 shows an SC turn which assumes a vehicle moving forward and turning left. The method explained in section III-B can be readily used to obtain SC turns moving forward, independent of the direction they are turning. The procedure to obtain an SC turn moving backwards is analogous.

There are a total of 16 possible SC paths between the two sets of 4 SC turns spanned from $\mathbf{q}_s$ and $\mathbf{q}_g$. Each path is found by computing the SC path, resulting from connecting two SC turns, as detailed in III-C. Each SC path length is evaluated, and the shortest is selected as the solution.

IV. CUBIC CURVATURE PATHS

Cubic curvature paths are a building block of an SC turn. They are used as transition paths connecting configurations to and from an arc circle (paths $\Gamma_{1,2}$ and $\Gamma_{3,4}$ in III-B), and they guarantee the sharpness continuity of the whole path.

A. Introduction

Cubic curvature paths are defined as paths with a cubic curvature profile $\kappa(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0$, where s is the length along the path. A cubic curvature profile is the minimum degree polynomial that allows to define arbitrary initial and final curvatures, κ_i and κ_f, and sharpnesses α_i and α_f. The sharpness profile of these paths is given by:

$$\alpha(s) = \frac{\partial \kappa(s)}{\partial s} = 3a_3 s^2 + 2a_2 s + a_1. \tag{6}$$

In order to find the parameters of the cubic polynomial, we use the initial and final constraints:

$$\kappa(0) = \kappa_i, \quad \alpha(0) = \alpha_i, \quad \kappa(s_f) = \kappa_f, \quad \alpha(s_f) = \alpha_f, \tag{7}$$

where s_f is the path length, and is itself an unknown.

We want to ensure sharpness continuity, so we need to set the initial and final sharpness values, α_i and α_f, to zero. This allows us to stitch together cubic curvature paths with line and arc segments, which have null sharpness, while ensuring sharpness continuity. It should be noted that (7) does not take into account steering limitations, thus, infeasible paths can be generated. In the following we address this issue.

B. Ensuring Steering Rate and Acceleration Constraints

In order to have a feasible path, we need to ensure that a vehicle can follow it while complying with its steering constraints. If we make both κ_i and κ_f smaller in magnitude than κ_{max}, we ensure that the path always has a steering angle magnitude smaller than ϕ_{max}.

The steering angle profile corresponding to the cubic curvature path is first computed from the path curvature using (1). Assuming then that the vehicle is following the path at a given fixed velocity $\mathbf{v}$, the steering angle rate and acceleration profiles are computed. Both profiles have a peak magnitude rate $\dot{\phi}_{peak}$ and acceleration $\ddot{\phi}_{peak}$. In case $\dot{\phi}_{peak}$ is larger than the allowed maximum steering rate $\dot{\phi}_{max}$ the length s_f needs to be increased so that $\dot{\phi}_{peak} = \dot{\phi}_{max}$. This can be achieved by simply scaling the path length by $\dot{\phi}_{peak}/\dot{\phi}_{max}$. Similarly if $\ddot{\phi}_{peak} > \ddot{\phi}_{max}$, we scale the path length by a scaling factor of $\sqrt{(\ddot{\phi}_{peak}/\ddot{\phi}_{max})}$. To guarantee that the path respects both steering rate and acceleration limitations, we need to scale its length by the greater of the scaling factors. Once the new path length s_f is computed, the cubic curvature path is recomputed, by solving (7).

V. RESULTS

A. Convergence of SC Paths to Dubins Paths

As previously mentioned, Dubins paths are proven to be optimal in terms of length. SC paths are, however, longer than the Dubins path. This happens because SC turns have longer turning radii than a circular arc with radius κ_{max}^{-1}.

Figure 4 shows the Dubins path for a given start and goal configurations $\mathbf{q}_s$ and $\mathbf{q}_g$. Overlaid are SC paths with different maximum sharpness α_{max} for the same configurations $\mathbf{q}_s$ and $\mathbf{q}_g$. It is seen that the greater the maximum sharpness α_{max} of the SC paths is, *i.e.*, the greater the achievable steering rate and accelerations of the vehicle, the closer it
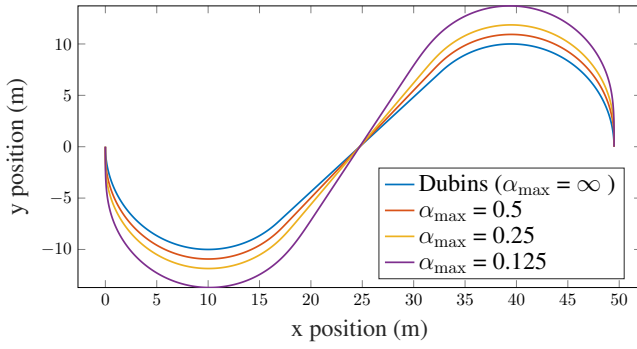
Fig. 4: SC paths with increasing sharpness α converge to the length optimal Dubins path.

approaches the Dubins path. This is somewhat intuitive, as increasing α_{max} results in increasing the rate of change of the curvature profile. If $\alpha_{max} \to \infty$, then the curvature changes would be immediate, and the SC path would be equivalent to the Dubins path, and as such, length optimal.

B. Notes on Computational Cost

As previously mentioned, given two configurations $\mathbf{q}_s$ and $\mathbf{q}_g$, the SC method computes all possible 16 SC turns and how they can be connected. The connection process, as detailed in section III-C, is computationally cheap. The bulk of processing comes from finding all 16 possible SC turns.

As seen in section III-B, an SC turn depends on the cubic curvature paths that are part of it. In order to evaluate these paths, one has to generate their curvature profiles from the given initial and final constraints. To comply with the steering constraints, a numerical evaluation of a steering profile must be done, in order to find the path length scaling factors, as detailed in section IV-B. When one has the desired curvature profile, the orientations θ can be obtained analytically. The x and y positions of the path are found by solving the vehicle model equations, using an Euler method, which has a high computational cost. This cost can be greatly reduced using precomputations, as detailed in the following.

C. Precomputation of Cubic Curvature Paths

As previously stated, the SC method computation speed is limited by the generation of the cubic curvature paths. Depending on the application, some assumptions can be made that greatly improve the computation speed, by allowing the precomputation of the cubic curvature paths to be used.

If one assumes that the start and end configurations $\mathbf{q}_s$ and $\mathbf{q}_g$ always have null curvatures, then one can compute, in an initialization procedure, all the possible cubic curvature paths starting and ending at curvatures $\kappa = 0, \pm\kappa_{max}$. Thus we skip the expensive generation of cubic curvature paths needed to find out the possible SC turns. In order to generate an SC turn, one just has to use the precomputed paths and apply rotations and translations on them.

The precomputation of paths can still be achieved, without limiting the start and goal configurations $\mathbf{q}_s$ and $\mathbf{q}_g$ to have null curvature. In fact, one can allow $\mathbf{q}_s$ and $\mathbf{q}_g$ to have curvature values belonging to finite discrete set.

D. Timing Evaluation

We test the steering method, measuring its computational speed for several problem instances. The method is implemented in `C++` and running on a Linux Mint distribution. The computer used is equipped with an Intel Core i7-6820HQ Processor running at 2.70 GHz, and with 16,0 GB of RAM.

We generate 1000 random pairs of start and goal configuration queries. Each query is repeated 100 times to get a better estimate of the average computational time. The start and goal configurations of each query are generated by sampling the x and y coordinates from a uniform random distribution between -50 and 50. The orientations sampled from the interval $[-\pi, \pi]$, and the curvatures from a discrete equispaced set of 11 curvatures $[-\kappa_{max}, \ldots, 0, \ldots, \kappa_{max}]$.

When making use of precomputations, we get an average time for finding a solution path of 70µs, while without precomputations we get an average time of 12ms. The precomputations greatly decrease the computational time. These results indicate that the steering method is extremely inexpensive when using precomputations. Even without precomputations, the method runs in few milliseconds, making it suitable for real-time applications.

E. Simulations

A simulation test is run in order to understand how the proposed paths affect the performance of a vehicle tracking them. A kinematic vehicle model coupled with a detailed steering actuator model are used to simulate a vehicle.

In the test, two paths consisting of a straight segment, a turn, and a straight segment are generated. The first path is a CC path [13], while the second is our proposed SC path. Both paths abide by the same maximum steering angle magnitude ϕ_{max} and steering angle rate $\dot{\phi}_{max}$ constraints. Additionally, the SC path respects also the steering angle acceleration $\ddot{\phi}_{max}$ constraint, unlike the CC paths.

The simulation assumes a steering actuator that is limited in terms of achievable steering angle magnitude, rate, and acceleration. The steering angle is controlled making use of a PID controller, which receives a steering angle reference, and actuates on the steering angle torque. The PID controller was tuned to achieve a step response with a relatively fast settling time and little overshoot. The steering angle reference is provided from a high-level path tracking controller. The high-level controller consists of a feedforward part and a feedback part. The feedforward part is obtained by finding the closest path point, and getting the corresponding steering angle reference at that point. The feedback part is a proportional controller regulating both lateral and heading errors. Such a controller is a simple implementation commonly used in path tracking applications.

Figure 5 shows the steering reference profiles of the paths to be tracked. The difference between them is in the shape of the increasing and decreasing sections of the steering angle. In the CC case, the steering angle change follows a linear profile while in the SC it follows a cubic profile.

Figure 6 shows the lateral and heading errors when the vehicle tracks both paths. The vehicle is initially placed
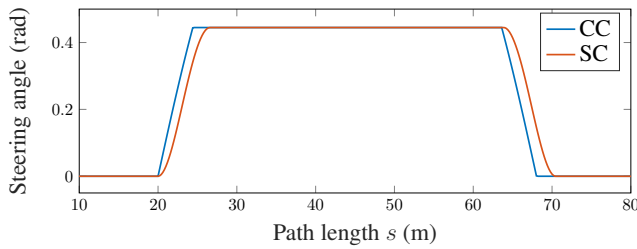
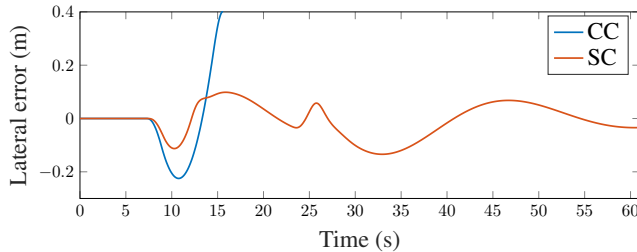Fig. 5: Steering reference profiles used in simulation.

Fig. 6: Lateral error when tracking a CC and an SC path. When tracking the CC path, the controller becomes unstable, resulting in an error that grows indefinitely, and out of scope of the graph. The SC path tracking is seen to be stable.

at the start of the path, and it follows the first straight segment perfectly. However, when the turning section starts, a deviation from the path begins to arise. The feedback part is then responsible for trying to regulate the errors to zero. Shortly after the turn begins, the CC case becomes unstable. On the other hand, the SC case is stable, and its error converges to zero. The error profiles show that the controller performance is worse when tracking CC paths.

The lateral acceleration and jerk (acceleration rate) experienced by the vehicle are related to passenger comfort. Figure 7 shows the lateral accelerations for a vehicle following the reference steering profiles without feedback actuation. It is seen that the CC path has large jerk values, which result from an aggressive steering actuation. The SC path steering profile achieves smoother lateral acceleration profiles.

VI. CONCLUSIONS

This paper presented the concept of SC paths. SC paths respect not only the maximum steering angle constraints,
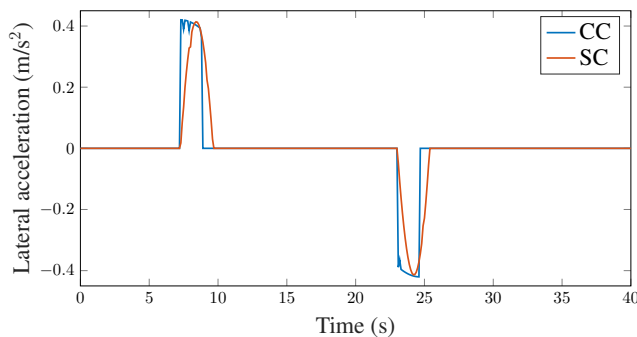

Fig. 7: Lateral acceleration when tracking a path without feedback, *i.e.*, using only feedforward references.

but also maximum steering rate and acceleration constraints. These properties ease the low-level controller task and introduce an higher degree of smoothness, improving the driving comfort and reducing actuator effort. This is of importance when dealing with heavy-duty vehicles, which are characterized by slow actuator dynamics.

As future work, one could extend this approach so that the SC paths handle more cases besides that of a combination of two SC turns connected by a line segment. This would allow to connect configurations that lie close together.

Controllers could also be designed so that they take advantage of the smooth properties of the path, without requiring the computational burden of more complex control approaches that are design to withstand lower quality paths.

This work can also be extended into time optimal trajectory planning. The velocity profile could be optimized, such that the vehicle performs the path in minimum time and obeys steering constraints. Moreover, different curvature profiles could be optimized with respect to time, and abiding by the steering constraints.

REFERENCES

[1] M. Buehler *et al.*, eds., The DARPA urban challenge: autonomous vehicles in city traffic. Vol. 56. springer, 2009.
[2] S. LaValle, Planning algorithms. Cambridge University Press press, 2006.
[3] J. Reuter, "Mobile robots trajectories with continuously differentiable curvature: an optimal control approach" Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (1998): 38-43.
[4] B. Nagy and A. Kelly, "Trajectory generation for car-like robots using cubic curvature polynomials" Field and Service Robots 11 (2001).
[5] J. McCrae and K. Singh, "Sketching piecewise clothoid curves" Computers & Graphics 33, no. 4 (2009): 452-461.
[6] A. Piazzi *et al.*, "η^3-Splines for the Smooth Path Generation of Wheeled Mobile Robots" IEEE Transactions on Robotics 23, no. 5 (2007): 1089-1095.
[7] Y. Kuwata *et al.*, "Real-time motion planning with applications to autonomous urban driving" IEEE Transactions on Control Systems Technology 17, no. 5 (2009): 1105-1118.
[8] S. Karaman *et al.*, "Anytime motion planning using the RRT*," in Robotics and Automation (ICRA), 2011 IEEE Int. Conf. on, pp. 1478-1483. IEEE, 2011.
[9] T. Shima *et al.*, "Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms" Computers & Operations Research 33, no. 11 (2006): 3252-3269.
[10] D. Dolgov *et al.*, "Path planning for autonomous vehicles in unknown semi-structured environments" The International Journal of Robotics Research 29, no. 5 (2010): 485-501.
[11] 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 79, no. 3 (1957): 497-516.
[12] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards" Pacific journal of mathematics 145, no. 2 (1990): 367-393.
[13] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths" IEEE Transactions on Robotics 20, no. 6 (2004): 1025-1035.
[14] P. F. Lima *et al.*, "Clothoid-based model predictive control for autonomous driving" In Control Conference (ECC), 2015 European, pp. 2983-2990. IEEE, 2015.
[15] G. Parlangeli and G. Indiveri, "Dubins inspired 2D smooth paths with bounded curvature and curvature derivative" IFAC Proceedings Volumes 43, no. 16 (2010): 252-257.
[16] M. Cirillo, "From videogames to autonomous trucks: A new algorithm for lattice-based motion planning" 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 148-153.