2021 IEEE Conference on Control Technology and Applications (CCTA).
August 8-11, 2021. San Diego, California

2021 IEEE Conference on Control Technology and Applications (CCTA).
August 8-11, 2021. San

2021 IEEE Conference on Control Technology and Applications (CCTA) | 978-1-6654-3643-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCTA48906.2021.9658923

G^2 smooth, curvature constrained, local motion planning for automated vehicles

Sanghoon Oh, Linjun Zhang, H. Eric Tseng, Lu Xu, and Gábor Orosz

***Abstract*— A new local motion planning method for automated vehicles in structured road scenarios is considered. Utilizing three consecutive clothoids, it is shown that real time generation of feasible trajectories for automated vehicles is possible. Separate path generation and consecutive velocity planning is considered. The path planning is reduced to a small set of nonlinear algebraic equations, while the velocity planning comprised of forward, backward and joint iteration processes. By varying some continuous parameters, multiple feasible plans can be generated and one of those can be chosen based on multiple criteria. A left turn at an intersection is used as an illustrative example.**

I. Introduction

Motion planning is one of the most challenging problems for problems automated vehicles [1], [2]. In the hierarchical architecture of the decision making, motion planning, and control, the middle layer plays the crucial role of providing a safe and efficient reference trajectory to be followed. The scope of this paper is limited to local motion planning after global waypoints are defined and the planner outputs a continuous motion that the vehicle could follow locally. We separate the motion planning problem to path planning and velocity planning that are executed consecutively.

Since it is known that optimal path planning subject to geometric and kinematic constraints is PSPACE-hard [3], there have been different approximate approaches for solving the problem using variational methods [4], graph-search methods [5], [6], and incremental search methods [7], [8], which lead to different challenges including local optimality, requiring motion primitives, or heavy computational burden. These global planning methods, are well suited for complex scenarios where obstacles are distributed over unstructured environment [9] and no specification of the local waypoints are required. Considering the structured environment of urban roads, typically a set of waypoints can be defined along the route [2]. Thus, interpolating curve planning, that can generate local smooth paths between given waypoints, looks appealing.

Interpolating curve planning methods may use different kinds of base curves, and clothoids are among the most intriguing ones as they can make smooth transitions between straight lines and curved arcs [2], [10], [11]. In a recent work [12], G^2 Hermite interpolation was proposed to generate paths that consists of three consecutive clothoids, are second order smooth, and satisfy the boundary conditions for arbitrary initial and final positions, angles, and curvatures. In this paper, an extension of this G^2 Hermite interpolation method is adapted for local motion planning of an automated vehicle when the target waypoint and configuration are given. There are three significant advantages of the proposed framework. First, the generated paths are twice continuously differentiable preventing undesirable jumps of the desired states. Second, by varying two free parameters continuously we can generate a large set of feasible paths for a given pair of waypoints. Third, the algorithm is computationally efficient allowing real time implementation. The proposed method is applicable for a large set of scenarios including intersections, lane changes, and U-turns.

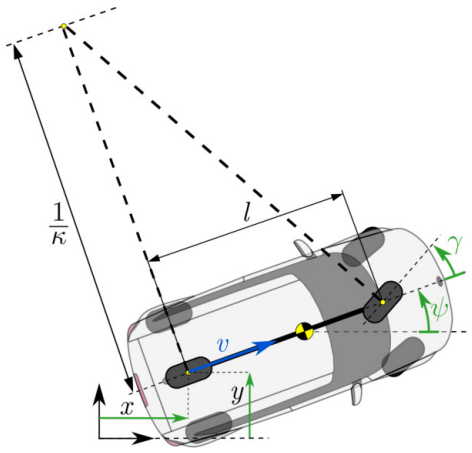

Fig. 1. Single track model of an automobile used in the paper.

The rest of the paper is organized as follows. We formulate the problem of motion planning in Section II. In Section III the lateral planning problem is described followed by the longitudinal planning in Section IV. In Section V a representative urban road scenario is used as an illustrative example. We conclude the results in Section VI where we also lay out some future research directions.

II. Problem Statement

Below we formulate the motion planning problem in a mathematical form. It assumed that the initial and the final waypoints are given by a higher level planner. First we formulate the mathematical model used for planning followed by the boundary value problem one need to solve to generate a planned trajectory.

S. Oh and G. Orosz are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA {osh,orosz}@umich.edu

L. Zhang, H. E. Tseng, and L. Xu are with Ford Motor Company, Dearborn, MI 48124, USA {lzhan203,htseng,lxu72}@ford.com

G. Orosz is also with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109, USA

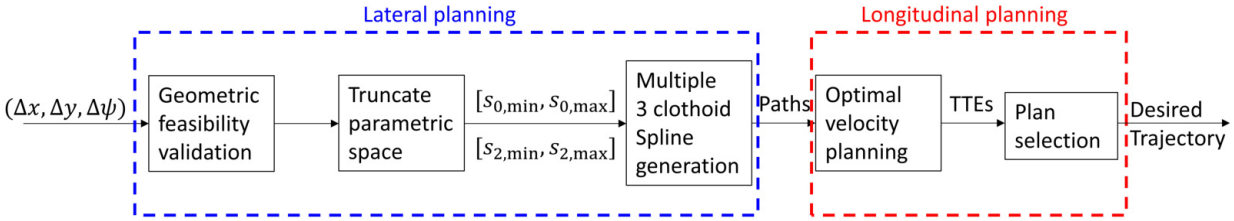

Fig. 2. Flow chart of the motion planner: lateral planning and longitudinal planning is done sequentially. The input of the motion planner is the relative position and heading of two consecutive waypoints. The output is the path in the configuration space and the longitudinal velocity along the path.

A. Single track vehicle model

In this paper, we use a so-called single track or bicycle model of an automobile depicted in Fig. 1 where the front wheels are unifies as one wheel and the rear wheels are also unified as one wheel. To simplify the model we assume no slip conditions at the wheels. The vehicle is described by three configuration coordinates: x and y denote the position of the center of the rear axle and ψ denotes the yaw angle; and the longitudinal velocity v. These constitute the state $X = [x, y, \psi, v]^\top$ such that $X \in \mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}_{\geq 0}$, that is, we assume the speed to be non-negative. The steering angle γ and the longitudinal acceleration a constitute the input, that is, $u = [\gamma, a]^\top$. The differential equations are formulated as

$$\begin{aligned} \dot{x} &= v\cos\psi, \\ \dot{y} &= v\sin\psi, \\ \dot{\psi} &= \frac{v}{l}\tan\gamma, \\ \dot{v} &= a, \end{aligned} \tag{1}$$

where the dot represents the derivative with respect to time t and l denotes the wheel base. The input constraints are expressed by

$$\begin{aligned} |\gamma| &\leq \gamma_{\max}, \\ |\dot{\gamma}| &\leq \Omega_{\max}, \\ a_{\min} \leq a &\leq a_{\max}. \end{aligned} \tag{2}$$

Here we use the values $\gamma_{\max} = \pi/6$, $\Omega_{\max} = \pi/6$ 1/s, $a_{\min} = -8$ m/s^2, and $a_{\max} = 5$ m/s^2. Note that acceleration constraint correspond to the performance limits of a passenger vehicle. By setting these for lower values one may take into account driver comfort. We also remark that additional velocity constraints will be defined further below by limiting the lateral acceleration.

Since the wheels assumed to have no side slip, the curvature for the rear wheel center point can be expressed by

$$\kappa = \frac{\tan\gamma}{l}. \tag{3}$$

This shows that setting the steering angle in (1) is equivalent to setting the curvature of the path. Additionally, the steering input constraint in (2) can be translated to curvature constraint.

B. Motion planning problem

The proposed planner generates a continuous trajectory in the 4-dimensional state space. This corresponds to a path in the 3-dimensional configuration space $\hat{X} = [x, y, \psi]^\top \in \mathbb{R}^2 \times \mathbb{S}$ and the velocity $v \in \mathbb{R}_{\geq 0}$ given along this path, while satisfying the constraints of the vehicle. We assume that the vehicle starts at a global waypoint, and that the next global waypoint to be reached is given. For simplicity the initial position is placed to the origin and the initial yaw angle is set to zero. This is equivalent to planners generating a plan in the frame attached to the vehicle and leads to the boundary conditions

$$\begin{aligned} x(0) &= 0, \quad x(T) = \Delta x, \\ y(0) &= 0, \quad y(T) = \Delta y, \\ \psi(0) &= 0, \quad \psi(T) = \Delta\psi, \\ v(0) &= v_{\mathrm{i}}, \quad v(T) = v_{\mathrm{f}}, \\ \gamma(0) &= 0, \quad \gamma(T) = 0, \end{aligned} \tag{4}$$

where the final time T is not fixed. Moreover, the boundary conditions for the velocity depend on the context of the road scenario, that is, v_{i} and v_{f} are not necessarily fixed. Finally, notice that the initial and final steering angles are set to zero for the sake of simplicity. However, the methods presented below are also applicable for nonzero steering angles. Thus, the motion planning problem can be formulated as follows.

Problem 1. Given the initial and final positions and headings one should find a trajectory that satisfies the differential equations (1), the constraints (2), and the boundary conditions (4).

This problem is divided into two parts: lateral planning and longitudinal planning, as illustrated by the flowchart in Fig. 2. The lateral planner generates a set of kinematically feasible continuous paths in the configuration space while taking into account the steering angle constraint in (2). During this process the geometric feasibility of the next waypoint is verified and the parameter space of is truncated to the domain where feasible paths can be generated. The longitudinal planning generates a velocity profile for each path while taking into account the constraints for the steering rate and longitudinal acceleration in (2) as well as some additional velocity constraints. A plan among the set of designed trajectories can be selected based on criteria set by the user, for example, based on the time to reach (TTR) as indicated in Fig. 2.

III. Lateral Planing

In order to separate the lateral planning and longitudinal planing, the path is parameterized by the distance s traveled by the rear axle center point that is often called arclength parameterization. The total traveled distance is denoted by s_f and this is not fixed a priori. The derivatives of the variables x, y, ψ with respect the arclength s can be obtained from the first three equations of (1) using $\dot{s} = v$ and the chain rule:

$$\begin{aligned} x'(s) &= \cos\psi(s), \\ y'(s) &= \sin\psi(s), \\ \psi'(s) &= \kappa(s), \end{aligned} \tag{5}$$

while (4) yields the boundary conditions

$$\begin{aligned} x(0) &= 0, \quad x(s_\mathrm{f}) = \Delta x, \\ y(0) &= 0, \quad y(s_\mathrm{f}) = \Delta y, \\ \psi(0) &= 0, \quad \psi(s_\mathrm{f}) = \Delta\psi, \\ \kappa(0) &= 0, \quad \kappa(s_f) = 0. \end{aligned} \tag{6}$$

Finally, using (3), the first constraint in (2) can be rewritten for the curvature:

$$|\kappa(s)| \leq \frac{\tan\gamma_{\max}}{l}. \tag{7}$$

Thus, the lateral planning (or path planning problem) can be formally cast as follows.

Problem 2. Given the differential equation (5), the boundary condition (6), and the constraint (7), the lateral planning problem searches for a continuous function $\kappa(s)$.

This problem may be solved in many different ways. One one hand, one may try to assume a specific form of $\kappa(s)$, e.g., polynomial. However, such methods often ignore the boundary condition or continuity, which is important for generating smooth path. On the other hand, one may generate a mesh in the archlength s and try to generate numerical solution, e.g., by using numerical collocation. This, however, leads to a large set of algebraic equations, whose solution demand large computational efforts. Here, inspired by the method in [12], we take a different approach assuming $\kappa(s) = as + b$ (clothoid) on three consecutive segments with lengths s_0, s_1, and s_2 as illustrated in Fig. 3 where $s_\mathrm{f} = s_0 + s_1 + s_2$. This will lead to a small number of algebraic equations that can be solved very efficiently, resulting in an extremely fast method to generate paths that are second order smooth and satisfy the boundary conditions. Moreover, by varying the lengths of the different segment allows us to generate a large set of feasible paths.

A. Three-clothoid path generation

Here we formulate the three-clothoid problem mathematically. As shown below the design results in two free parameters, which are chosen to be the lengths of the first and thirds segments s_0 and s_2. Without the curvature constraint (7), the existence of the three-clothoid curve was proven in [12] (for arbitrary $s_0, s_2 > 0$). With (7) in place, we need to prove the geometric feasibility of the next waypoint by showing that there exist at least one three-clothoid path.

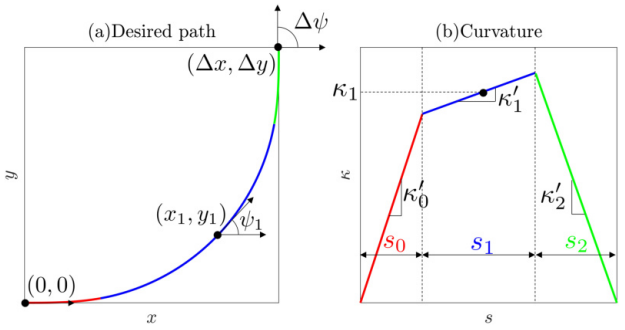

Fig. 3. Planning a path with three clothoids in a left turn example with all relevant variables variables are depicted. The curvature of the spline is continuous while meeting all of the boundary conditions.

Then, we need to find the domain in the (s_0, s_2) parameter space with three-clothoid solutions. Varying the values within this restricted domain, that we refer to as the truncation of the parameter space, allow us to compute a large set of different solutions.

Let us start with defining the Fresnel integrals:

$$\begin{aligned} A(a,b,c) &:= \int_0^1 \cos\Big(\frac{a}{2}t^2 + bt + c\Big)\mathrm{d}t, \\ B(a,b,c) &:= \int_0^1 \sin\Big(\frac{a}{2}t^2 + bt + c\Big)\mathrm{d}t. \end{aligned} \tag{8}$$

Using these one can set up the continuity conditions and the boundary conditions for three clothoids yielding 8 nonlinear algebraic equations with 10 variables:

$$\begin{aligned} x_1 - \frac{s_1}{2}A\Big(\frac{\kappa_1' s_1^2}{4}, -\frac{\kappa_1 s_1}{2}, \psi_1\Big) &= s_0 A\big(\kappa_0' s_0^2, 0, 0\big), \\ x_1 + \frac{s_1}{2}A\Big(\frac{\kappa_1' s_1^2}{4}, \frac{\kappa_1 s_1}{2}, \psi_1\Big) &= \Delta x - s_2 A(\kappa_2' s_2^2, 0, \Delta\psi), \\ y_1 - \frac{s_1}{2}B\Big(\frac{\kappa_1' s_1^2}{4}, -\frac{\kappa_1 s_1}{2}, \psi_1\Big) &= s_0 B\big(\kappa_0' s_0^2, 0, 0\big), \\ y_1 + \frac{s_1}{2}B\Big(\frac{\kappa_1' s_1^2}{4}, \frac{\kappa_1 s_1}{2}, \psi_1\Big) &= \Delta y - s_2 B\big(\kappa_2' s_2^2, 0, \Delta\psi\big), \\ \frac{\kappa_1' s_1^2}{8} - \frac{\kappa_1 s_1}{2} + \psi_1 &= \frac{\kappa_0' s_0^2}{2}, \\ \frac{\kappa_1' s_1^2}{8} + \frac{\kappa_1 s_1}{2} + \psi_1 &= \frac{\kappa_2' s_2^2}{2} + \Delta\psi, \\ -\frac{\kappa_1' s_1}{2} + \kappa_1 &= \kappa_0' s_0, \\ \frac{\kappa_1' s_1}{2} + \kappa_1 &= -\kappa_2' s_2. \end{aligned} \tag{9}$$

After choosing the lengths s_0 and s_2 of the first and the third segments, these equations can be solved for the 8 unknowns $(s_1, x_1, y_1, \psi_1, \kappa_1, \kappa_1', \kappa_0', \kappa_2')$. As indicated in Fig. 3, x_1, y_1 and ψ_1 denote the location and orientation of the middle point of the second arc, $\kappa_1 = \kappa\big(s_0 + \frac{s_1}{2}\big)$ denote the curvature at this middle point, while κ_0', κ_1' and κ_2' denote the constant sharpness of the first, second and third arcs, respectively. Using algebraic manipulations, (9) can be reduced to two nonlinear equations for the two unknowns s_1 and κ_1'. These can be solved using, for example, Newton's method very fast; see [12] for details.

B. Geometric feasibility validation and parameter selection

Given the next waypoint the lateral planner needs to validate whether the planning problem is feasible, that is, whether there exist at least one three-clothoid curve that satisfies the boundary conditions and the curvature constraint. If the given waypoint is determined to be not feasible, the global planner should search for a substitute waypoint. Let us define the parameter vector $S = [s_0, s_2]^\top$ such that $S \in \mathbb{R}^2_{\geq 0}$ and search for the parameters $S^{\text{nec}} = [s_0^{\text{nec}}, s_2^{\text{nec}}]^\top$ that result in the path with the smallest maximum curvature. The existence of such point presents a necessary condition for the feasibility of the waypoint.

In order to find the point S^{nec} in the parameter space, we differentiate three different shape classes based on the configuration sketched in Fig. 4. As defined above, Δx and Δy indicate the location of the next waypoint while $\Delta\psi$ denote the orientation at the next waypoint. The shape classes are differentiated using the triangle formed by the green dashed line linking the waypoints and the lines given by the directions of the initial and final orientations. Namely, we differentiate the "symmetric" case $a_1 = a_2$ and the "asymmetric" cases $a_1 > a_2$ and $a_1 < a_2$. The proposition below specifies the location of the point S^{nec} in the parameter space.

Proposition 1. If for given $(\Delta x, \Delta y, \Delta\psi)$ there exists parameter combination S^{nec}, that generates the smallest maximum curvature along the three-clothoid curve. This point is located at the origin for $a_1 = a_2$, at the s_0 axis for $a_1 > a_2$, and the the s_2 axis for $a_1 < a_2$.

This proposition is proven numerically using the contour plots in Fig. 5. The contours indicate the domains in the parameter space where there exist three-clothoid solutions given the constraint (7). For a given value of the maximum steering angle $\gamma_{\max}$, the contour can be determined by assuming that either $|\kappa(s_0)| = \tan\gamma_{\max}/l$ or $|\kappa(s_2)| = \tan\gamma_{\max}/l$; cf. (7). As $\gamma_{\max}$ is decreased, the contours shrink to the point S^{nec} indicated by a red cross.

For the symmetric case $a_1 = a_2$ depicted in panel (a), S^{nec} is located at the origin. This corresponds to a special case where the first and last sections are of zero length $s_0 = s_2 = 0$ and the middle section becomes an arc of zero sharpness $\kappa_1' = 0$. For the asymmetric case $a_1 > a_2$ depicted in panel (b), the length of the third segment is zero $s_2 = 0$ while the sharpness of the second segment is zero $\kappa_1' = 0$. This simplifies (9) to four nonlinear algebraic equations:

$$\begin{aligned} s_0 A\big(\kappa_0' s_0^2, 0, 0\big) + c_0 A\big(0, -\kappa_1 s_1, \Delta\psi\big) &= \Delta x, \\ s_0 B\big(\kappa_0' s_0^2, 0, 0\big) + c_0 B\big(0, -\kappa_1 s_1, \Delta\psi\big) &= \Delta y, \\ c_0\kappa_0' s_0 - \kappa_1 s_1 &= 0, \\ \kappa_1 - \kappa_0' s_0 &= 0, \end{aligned} \tag{10}$$

where

$$c_0 = \frac{\Delta\psi}{\kappa_0' s_0} - \frac{s_0}{2}. \tag{11}$$

This system can be solved for the four unknowns $(s_0, s_1, \kappa_1, \kappa_0')$. By substituting the third equation in (10) into the first two equations and noticing that last equation

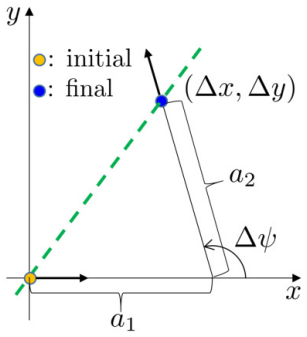

Fig. 4. Sketch of the configuration.

decouples, we obtain the two nonlinear algebraic equations

$$\begin{aligned} s_0 A\big(\kappa_0' s_0^2, 0, 0\big) + c_0 A\big(0, -c_0\kappa_0' s_0, \Delta\psi\big) &= \Delta x, \\ s_0 B\big(\kappa_0' s_0^2, 0, 0\big) + c_0 B\big(0, -c_0\kappa_0' s_0, \Delta\psi\big) &= \Delta y. \end{aligned} \tag{12}$$

These can be solved for the two unknowns (s_0, κ_0') giving the location of S^{nec}. Finally, the other asymmetric case $a_1 < a_2$ depicted in panel (c), can be handled by simply switching the roles of s_0 and s_2.

Once the waypoint is shown to be geometrically feasible, it is guaranteed that at least one feasible path can be generated with the three-clothoid method. However, as illustrated in Fig. 5, there exist a domain in parameter space where multiple three-clothoid paths exist. Indeed, the size of this domain depends on $\gamma_{\max}$ as indicated by color. Generating multiple paths is beneficial since one can choose the "best" plan among them based on multiple criteria. For efficient multiple path generation, one shall truncate the parameter space and restrict the choice of s_0 and s_2 inside the contour given by $\gamma_{\max}$. Within this domain one may select different pairs (s_0, s_2) and generate the corresponding paths by solving (9) (or the two equivalent nonlinear equations for s_1 and κ_1'). Generating a path takes about 6ms in Matlab, and this is expected to be much faster when using lower level languages. Moreover, to generate multiple paths one can use parallel computing. These make the proposed method viable for on board implementation running in real time.

IV. Longitudinal Planing

For each path generated in the lateral planning layer, a longitudinal velocity profile shall be generated. The last equation of (1) combined with $\dot{s} = v$ yields the second order system

$$\begin{aligned} \dot{s} &= v, \\ \dot{v} &= a, \end{aligned} \tag{13}$$

which is augmented by the boundary conditions

$$\begin{aligned} s(0) &= 0, \quad s(T) = s_{\text{f}}, \\ v(0) &= v_{\text{i}}, \quad v(T) = v_{\text{f}}, \end{aligned} \tag{14}$$

cf. the fourth row in (4). Note that s_{f} is obtained from the lateral planning, T is not fixed, while v_{i} and v_{f} may or may not be fixed based on the particular scenario. Also, we use the last two equations of (2) and with an additional constraint

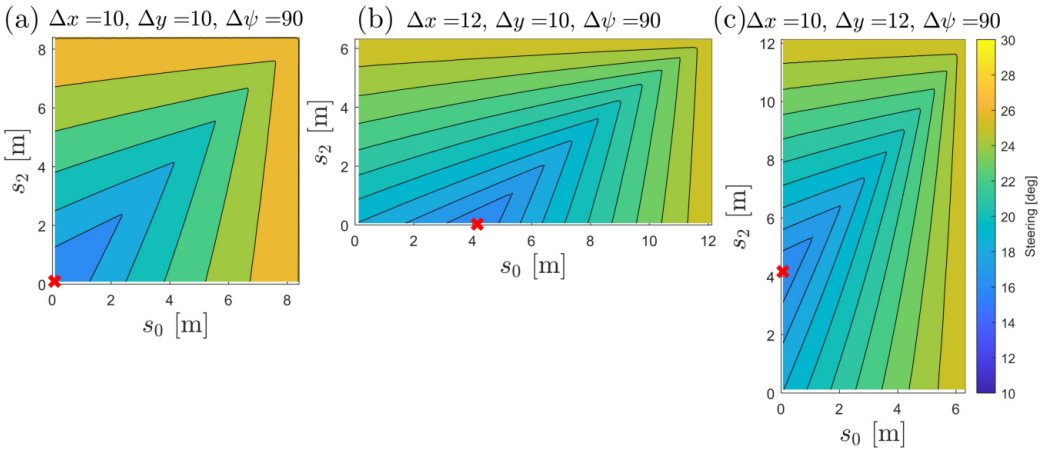

Fig. 5. Contours of maximum steering angle for a 90 degree left turn for the three shape classes: (a) $a_1 = a_2$, (b) $a_1 > a_2$ and (c) $a_1 < a_2$. The point S^{nec} is indicated with red cross in each case.

on the lateral acceleration when generating the longitudinal plan. These result in

$$\begin{aligned} a_{\min} \le a \le a_{\max}, \\ 0 \le v(s) \le \overline{v}(s), \end{aligned} \tag{15}$$

where $\overline{v}(s)$ encodes the steering rate constraint and the lateral acceleration constraint as derived below.

Once the curvature of the path $\kappa(s)$ is determined from lateral planning, one may limit the lateral acceleration by limiting the velocity according to

$$|\kappa(s)| v(s)^2 \le a_{\text{lat,max}}. \tag{16}$$

Here we use the value $a_{\text{lat,max}} = 3$ m/s^2. Moreover, the steering rate constraint in (2) can be translated to a velocity constraint taking the time derivative of (3):

$$\dot{\gamma} = l\, v(s)\, \kappa'(s) \cos^2 \gamma = \frac{l\, v(s)\, \kappa'(s)}{1 + \tan^2 \gamma}, \tag{17}$$

and substituting (3) results in

$$\dot{\gamma} = \frac{l\, v(s)\, \kappa'(s)}{1 + l^2 \kappa^2(s)}, \tag{18}$$

which yield

$$\left| \frac{l v(s)\, \kappa'(s)}{1 + l^2 \kappa^2(s)} \right| \le \Omega_{\max}. \tag{19}$$

The constraints (16) and (19) can be encoded in the second row of (15) with

$$\overline{v}(s) = \min \left\{ \sqrt{|\kappa(s)|\, a_{\text{lat,max}}},\ \Omega_{\max} \frac{1 + l^2 \kappa^2(s)}{l\, |\kappa'(s)|} \right\}. \tag{20}$$

Thus, the longitudinal planning problem is formulated as follows.

Problem 3. Given a continuous path with curvature $\kappa(s), s \in [0, s_{\text{f}}]$, we shall obtain a solution $v(s)$ for (13) with boundary conditions (14), subject to the constraint (15,20).

This problem can be solved in many different ways. Here we us a method which minimizes the final time T based on the idea presented in [13]. This consists of three iteration processes: forward iteration, backward iteration, and joint iteration. In order to execute these the arclength s_{f} is divided into n intervals of length $h = s_{\text{f}}/n$ (chosen to be 0.01 m in our example). Note that this step size fine enough, but one can choose even smaller size without sacrificing computation time since the complexity is linear to the number of intervals.

Forward iteration propagates forward along the path using $a_{\max}$ starting from v_{i}. (If v_{i} is not given an arbitrarily value is used). This generates the solution profile

$$\begin{aligned} v_0^{\text{F}} &= v_{\text{i}}, \\ v_j^{\text{F}} &= \min \left\{ \sqrt{\left(v_{j-1}^{\text{F}}\right)^2 + 2h\, a_{\max}}, \overline{v}_j \right\}, \quad j = 1, \ldots, n. \end{aligned} \tag{21}$$

Backward iteration propagates backward along the path using $a_{\min}$ starting from v_{f}. (Again, if v_{f} is not given an arbitrarily value is picked). This generates the solution profile

$$\begin{aligned} v_n^{\text{B}} &= v_{\text{f}}, \\ v_j^{\text{B}} &= \min \left\{ \sqrt{\left(v_{j+1}^{\text{B}}\right)^2 - 2h\, a_{\min}}, \overline{v}_j \right\}, \quad j = 0, \ldots, n-1. \end{aligned} \tag{22}$$

Finally, the joint iteration process chooses the minimum value between the forward and backward iteration result:

$$v_j = \min \left\{ v_j^{\text{F}}, v_j^{\text{B}} \right\}, \quad j = 0, \ldots, n. \tag{23}$$

As $n \to \infty$ the generated profile approaches a continuous one while the time complexity of the algorithm increases linearly with the discretization number n [13].

This method allows one to generate a velocity profile for each path in real time resulting in a large set of trajectories to choose from. Among the multiple plans (now containing both path and velocity), one may chose the "best" according to his/her objective. In the upcoming case studies, the minimal time to reach (TTR) is used to pick the desired trajectory.

V. Case Study

We consider a left turn to demonstrate the methods developed above. First we study the feasibility of the next waypoint, followed by the design of multiple trajectories. Finally, the trajectory with the shortest path is selected.

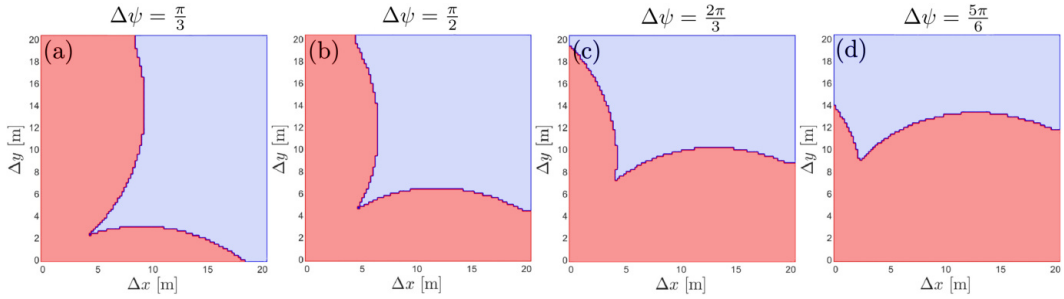

Fig. 6. Geometric feasibility domains (shaded blue) in the $(\Delta x, \Delta y)$ plane for different $\Delta\psi$ values as indicated.

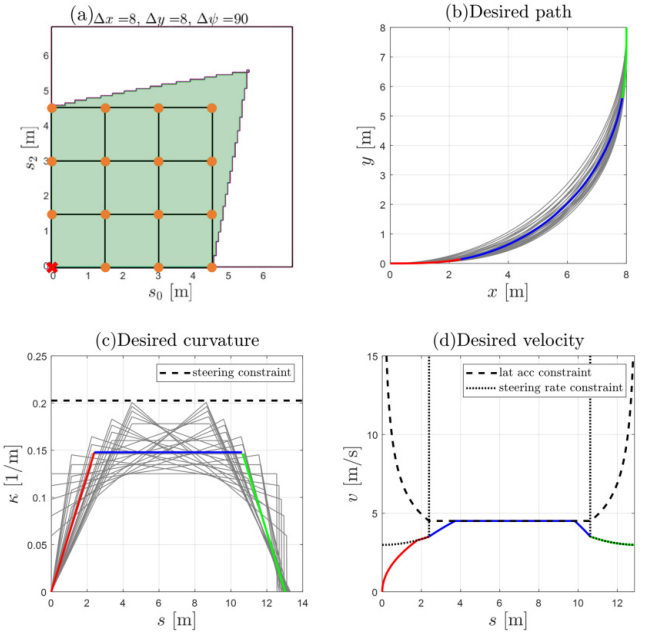

Fig. 7. Motion planning results for a 90 degree left turn. (a) Points selected in the truncated parameter domain shaded green. (b) Multiple feasible paths are shown as gray and the three segments of the selected one are colored red, blue, and green. (c) Curvatures for the feasible paths with the steering angle constraint indicated. (d) Velocity plan along the chosen path with the lateral acceleration and steering rate constraints indicated.

The feasibility domain is depicted in Fig. 6 in the configuration space. Since the configuration space is three dimensional, we show slices in the $(\Delta x, \Delta y)$ plane for different yaw angles $\Delta\psi$ as indicated. The feasible region is shaded as light blue while the infeasible region is shaded light red.

The planned trajectories for a 90 degree left turn are shown in Fig. 7 while using $(\Delta x, \Delta y, \Delta\psi) = (10\text{m}, 10\text{m}, \pi/2)$. We assume that the initial velocity is fixed to zero while the final velocity is not fixed. Panel (a) shows the selections of the (s_0, s_2) parameter pairs (red points) from the truncated parameter domain indicated by green shading. Panel (b) depicts the paths while panel (c) shows the curvatures along the paths.The total computation time to generate the 16 different paths is 0.33sec in Matlab. The velocity profile in (d) is shown for the trajectory with the fastest TTR and the corresponding path is highlighted with color in panels (b) and (c). Notice that this path consists of two clothoids connected by a constant curvature arc in the middle.

VI. Conclusion

A G^2 smooth local motion planner for automated vehicles was presented. The method separated the lateral path generation and the longitudinal velocity profile generation, handled the kinematic constraints, and was fast enough to run on board in real time. The algorithm generated multiple feasible plans to be able to choose best among them with little computational burden. In the future, the algorithm shall be extended to the case with arbitrary curvatures at the waypoints and implemented in a rolling horizon fashion.

References

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[3] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual Symposium on Foundations of Computer Science*. IEEE, 1979, pp. 421–427.

[4] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[5] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[6] E. Schmerling, L. Janson, and M. Pavone, "Optimal sampling-based motion planning under differential constraints: the driftless case," in *IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 2368–2375.

[7] 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*. IEEE, 2013, pp. 5054–5061.

[8] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5041–5047.

[9] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[10] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 441–448, 1995.

[11] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 583–601, 2003.

[12] E. Bertolazzi and M. Frego, "On the G^2 hermite interpolation problem with clothoids," *Journal of Computational and Applied Mathematics*, vol. 341, pp. 99–116, 2018.

[13] L. Consolini, M. Locatelli, A. Minari, and A. Piazzi, "A linear-time algorithm for minimum-time velocity planning of autonomous vehicles," in *24th Mediterranean Conference on Control and Automation*. IEEE, 2016, pp. 490–495.