Speed Profile Planning in Dynamic Environments via Temporal Optimization

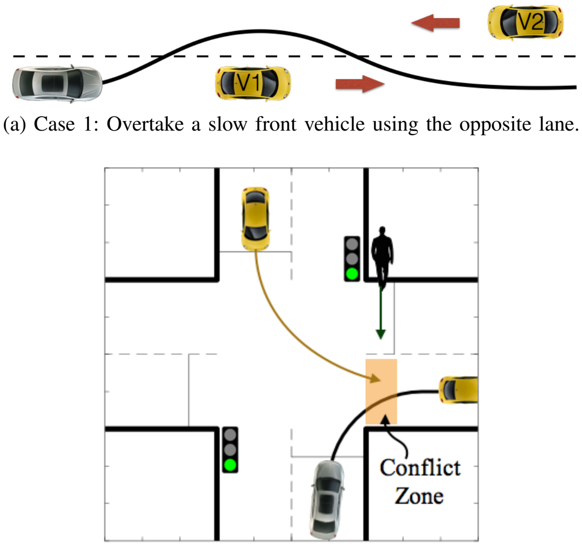
Changliu Liu, Wei Zhan and Masayoshi Tomizuka

***Abstract*—To generate safe and efficient trajectories for an automated vehicle in dynamic environments, a layered approach is usually considered, which separates path planning and speed profile planning. This paper is focused on speed profile planning for a given path that is represented by a set of waypoints. The speed profile will be generated using temporal optimization which optimizes the time stamps for all waypoints along the given path. The formulation of the problem under urban driving scenarios is discussed. To speed up the computation, the non-convex temporal optimization is approximated by a set of quadratic programs which are solved iteratively using the slack convex feasible set (SCFS) algorithm. The simulations in various urban driving scenarios validate the effectiveness of the method.**

I. Introduction

Autonomous driving is widely viewed as a promising technology to revolutionize today's transportation system. However, it is still challenging to plan collision-free, time-efficient and comfortable trajectories for an automated vehicle in dynamic environments such as urban roads, since the vehicle needs to interact with other road participants. For example, how to overtake a slow front vehicle safely using the opposite lane as shown in Fig.1a and how to turn safely at an intersection when pedestrians are crossing and vehicles in the opposite lane are turning left as shown in Fig.1b.

To respond to other road participants, the automated vehicle can resort to either spatial maneuvers such as detours or temporal maneuvers such as slowing down or speeding up. Regarding these two kinds of maneuvers, there are two planning frameworks in literature, e.g. the integrated framework and the layered framework. The integrated framework relies on spatiotemporal planning, which considers the spatial and temporal maneuvers simultaneously. Typical spatiotemporal planning methods include: 1) search-based methods which rely on spatiotemporal lattice as discussed in [1] and [2]; and 2) optimization-based methods which comprehensively consider driving quality as well as feasibility and safety constraints as discussed in [3]. However, the computational complexity for these methods is comparatively high [4]. On the other hand, the layered framework separates the considerations on the spatial and temporal maneuvers by planning a path first and then generating a speed profile along the path. Computational complexity can be greatly reduced by employing the layered framework as discussed in [4], [5], [6], [7], and [8]. Moreover, in the cases when there is not much freedom for an automated vehicle to choose alternative paths, nudging the speed profile is the only choice to respond to other road participants.



(a) Case 1: Overtake a slow front vehicle using the opposite lane.

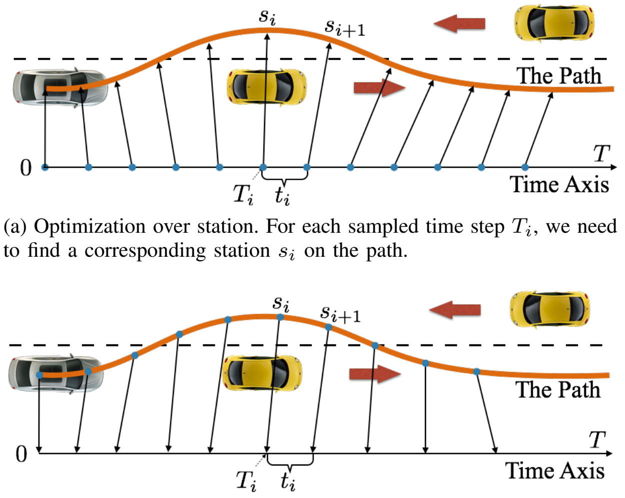
(b) Case 2: Right turn at an intersection.

Fig. 1: Urban Driving Scenarios.

In literature, speed profiles for a smooth path can be obtained in the following ways: 1) by selection from a pool of diverse speed profile candidates as discussed in [4], [5] and [6]; 2) by A* search [9]; 3) by quintic Bézier curves [10]; or 4) by direct optimization [7]. The discrete nature of the candidate pool and the state lattice may affect the optimality of trajectories. Meanwhile, it is hard to consider collision avoidance using the quintic Bézier curves. This paper will focus on direct optimization to obtain safe, time-efficient, smooth and comfortable speed profiles.

A speed profile is a one-to-one mapping between the time domain and the distance domain of the path. Optimization can be performed either over station as shown in Fig.2a or over time as shown in Fig.2b. Optimization over station requires an analytical parameterization of the path as discussed in [4], [5], [6], [7] and [8]. However, globally continuous analytical parameterization of a complicated path is difficult, which usually requires approximation. For a curvy path, the approximation may introduce infeasibility, e.g. curvature exceeding the vehicle's kinematic limits. Moreover, the complexity of the optimization problem may increase when complicated expressions of the parameterization enter the objective function. On the other hand, if we optimize over time as discussed in [11], only a sequence of way points are needed instead of a continuous parameterization of the path.

C. Liu, W. Zhan and M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: changliuliu, wzhan, tomizuka@berkeley.edu).



(a) Optimization over station. For each sampled time step T_i, we need to find a corresponding station s_i on the path.

(b) Optimization over time. For each sampled station s_i, we need to find a corresponding time stamp T_i on the time axis.

Fig. 2: Two optimization schemes to obtain the speed profile.

Therefore, this paper will focus on speed profile planning via temporal optimization which optimizes the time stamps for all waypoints along a given path.

Nonetheless, since the velocity, acceleration and jerk of a vehicle depend nonlinearly on the time stamps, the temporal optimization problem is still nonlinear and non-convex. To speed up the computation, we approximate the problem by a sequence of quadratic programs using the slack convex feasible set (SCFS) algorithm [12].

The contributions of the paper lie in: 1) novel formulation of the speed profile planning problem in urban environments by temporal optimization; 2) introduction of an efficient algorithm to obtain speed profiles in real-time by quadratic approximation of the non-convex temporal optimization. The remainder of the paper is organized as follows: nomenclature is presented in Section II; Section III formulates the temporal optimization problem for speed profile planning; Section IV discusses the quadratic approximation of the temporal optimization problem; Section V illustrates the performance of the proposed method through several case studies; and Section VI concludes the paper.

II. Nomenclature

p	position of the center of the rear axis $\in \mathbb{R}^2$;
s	station along a path $\in \mathbb{R}$;
v	velocity $\in \mathbb{R}^2$;
a	acceleration $\in \mathbb{R}^2$;
j	jerk $\in \mathbb{R}^2$;
θ	vehicle heading $\in [0, 2\pi)$;
τ	longitudinal direction $\in \mathbb{R}^2$;
η	lateral direction $\in \mathbb{R}^2$;
t	relative time interval $\in \mathbb{R}$;
T	absolute time stamp $\in \mathbb{R}$;
h	horizon of the problem $\in \mathbb{N}$.

III. Problem Formulation

To generate a desired speed profile, the vehicle needs to consider driving quality (such as time-efficiency, speed limit, longitudinal and lateral comfort), feasibility (such as longitudinal and lateral acceleration limits) and safety (collision avoidance). All these factors will be evaluated in the optimization. In this section, the mathematical problem underlying speed profile planning will be discussed first by introducing the $s-T$ graph. Then the temporal optimization problem for speed profile generation will be formulated.

A. Representing Speed Profile Using the $s-T$ graph

Denote the path of the center of the rear axis of a vehicle as $\mathbf{p} \subset \mathbb{R}^2$, which is a one dimensional manifold that can be parameterized using only one parameter. Denote the parameter as s, e.g. $\mathbf{p}(s) \in \mathbb{R}^2$. The parameter s is also called the station along the path. One intuitive parameterization is the distance along the path. In this case, the path starts at $\mathbf{p}(0)$; the tangent vector $\dot{\mathbf{p}}(s) := \partial\mathbf{p}/\partial s$ has unit length; and the norm of $\ddot{\mathbf{p}}(s) := \partial^2\mathbf{p}/\partial s^2$ represents the curvature of the path. In urban environment, another parameterization is the distance along the lane.

Path $\mathbf{p}$ does not contain any speed information. In order to determine the speed at each station, a time axis needs to be introduced. The $s-T$ graph is shown in Fig.3a where the vertical axis is the one dimensional parameterization of the path and the horizontal axis is the time axis. A speed profile $\mathcal{V}$ is equivalent to a monotone curve on the $s-T$ graph. Since the curve is monotone, there are two ways to obtain the speed profile. The first way is to find a mapping from the T axis to the s axis, e.g. fix a sequence of time steps $\{T_i\}$ and choose the desired s_i for each T_i as shown in Fig.2a. The second way is to find a mapping from the s axis to the T axis, e.g. fix a sequence of sampled stations $\{s_i\}$ and choose the desired T_i for each s_i as shown in Fig.2b. The first way is the optimization over station as s_i's are the decision variables, while the second way is the optimization over time as T_i's are the decision variables. In either case, the speed profile is represented by the curve $\mathcal{V} = \{(T_i, s_i)\}_i$ in the $s-T$ graph.

B. Computing Speed, Acceleration and Jerk

Consider the speed profile $\mathcal{V} = \{(T_i, s_i)\}_i$. Let $p_i := \mathbf{p}(s_i)$. Denote the time interval between p_i and p_{i+1} as $t_i = T_{i+1} - T_i$. When the time interval t_i and the distance between p_i and p_{i+1} are not too large, the velocity, acceleration and jerk at p_i can be approximated by

$$v_i = \frac{p_{i+1} - p_i}{t_i}, a_i = \frac{2(v_i - v_{i-1})}{t_i + t_{i-1}}, j_i = \frac{3(a_i - a_{i-1})}{t_i + t_{i-1} + t_{i-2}}. \tag{1}$$

Denote the heading of the vehicle at point p_i as $\theta_i := \arctan(\dot{\mathbf{p}}(s_i))$. The longitudinal and lateral directions at p_i are denoted as $\tau(\theta_i) := [\cos\theta_i, \sin\theta_i]$ and $\eta(\theta_i) := [\sin\theta_i, -\cos\theta_i]$ respectively. Define the longitudinal velocity as $v_i^\tau = v_i \cdot \tau(\theta_i)$, the lateral velocity as $v_i^\eta = v_i \cdot \eta(\theta_i)$, the longitudinal acceleration as $a_i^\tau = a_i \cdot \tau(\theta_i)$, the lateral

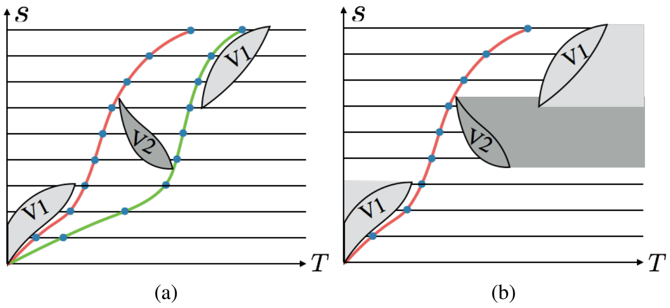


Fig. 3: Illustration of the constraints and the topological trajectories on the $s-T$ graph. (a) Multiple topological trajectories (or homotopy classes) given the original constraint. (b) The new constraint for the chosen homotopy class.

acceleration as $a_i^\eta = a_i \cdot \eta(\theta_i)$, the longitudinal jerk as $j_i^\tau = j_i \cdot \tau(\theta_i)$ and the lateral jerk as $j_i^\eta = j_i \cdot \eta(\theta_i)$.

Note that both p_i and θ_i depend on station s_i. If we optimize over station, an analytical expression of $\mathbf{p}(s)$ is indispensable. Then the drawbacks discussed in the introduction section will make the problem hard to solve. On the other hand, if we optimize over time, s_i is predefined and p_i and θ_i are fixed for all i. The analytical expression of $\mathbf{p}(s)$ is not necessary. Moreover, in a lot of cases, the paths are represented by a sequence of points instead of an analytical expression. Hence temporal optimization is desired.

C. Speed Profile Planning via Temporal Optimization

As discussed earlier, given a path that is represented by a list of points $\{p_i\}_1^{h+1}$, we need to plan a speed profile along the path such that potential collision with other road participants is avoided and the driving quality is optimized. We do so by optimizing over the time interval t_i's between any two consecutive points of the path, which is equivalent to optimize over the time stamps T_i's for points along the path.

The safety constraint for the optimization problem is illustrated in the $s-T$ graph in Fig.3a. The shaded area represents moments that another road participant is occupying part of the path (e.g. the conflict zone) so that the ego vehicle cannot enter. The constraints in Fig.3a are generated regarding the case shown in Fig.1a where the slow front vehicle is denoted as V1 and the vehicle in the opposite lane is denoted as V2. As the ego vehicle can choose to enter the conflict zone before or after another road participant, there are more than one topological trajectory (or homotopy class), hence more than one local optimum as shown in Fig.3a. The red trajectory corresponds to the case that the ego vehicle overtakes V1 before V2 passes by, while the green trajectory corresponds to the case that the ego vehicle overtakes V1 after V2 goes away. It is assumed that which homotopy class to choose is determined by a high level planner [13]. In temporal optimization, the speed profile planner only tries to find the local optimum inside the chosen homotopy class as shown in Fig.3b. This implies that the high level planner would specify a constraint $[T_i^{min}, T_i^{max}]$ for each absolute time stamp T_i.

Regarding the driving quality, a^τ and j^τ should be minimized for longitudinal comfort, while a^η and j^η should be minimized for lateral comfort. Moreover, a reference longitudinal speed v^r should be tracked for time efficiency. Hence, the following optimization problem can be posed.

Problem 1 (The Temporal Optimization Problem)**.**

$$\begin{aligned}\min_{t_1,\cdots,t_h}\ & w_1 \sum |a_i^\tau|^2 + w_2 \sum |a_i^\eta|^2 + w_3 \sum |j_i^\tau|^2 \\ & + w_4 \sum |j_i^\eta|^2 + w_5 \sum (v^r - v_i^\tau)^2 \qquad (2a)\\ s.t.\ & T_i \in [T_i^{min}, T_i^{max}] \qquad (2b)\\ & a_i \in \Omega, \qquad (2c)\end{aligned}$$

where (2a) is the cost function that considers driving quality. w_1, w_2, w_3, w_4 and w_5 are positive weights. Equation (2b) is the safety constraint. Equation (2c) is the feasibility constraint where Ω represents acceleration limits.

In this paper, considering the passenger comfort, we set the acceleration limits to be $|a^\tau| \leq \bar{a}$ and $|a^\eta| \leq \bar{a}$ where $\bar{a} := 2.5m/s^2$. Note that as a_i, v_i and j_i all depend nonlinearly on t_i, Problem 1 is highly nonlinear and non-convex.

IV. Quadratic Approximation of the Temporal Optimization

To solve Problem 1 efficiently, we transform it into a sequence of quadratic programs in the framework of the SCFS algorithm [12].

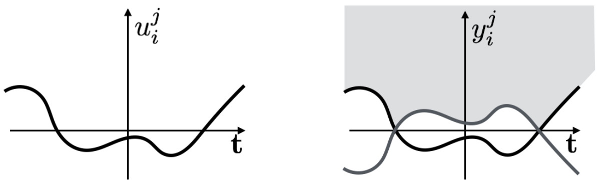
A. Rearranging the Problem

Define $\mathbf{t} := [t_1, \cdots, t_h]$, $u_i := [a_i^\tau, a_i^\eta, j_i^\tau, j_i^\eta, v^r - v_i^\tau]$ and $\mathbf{u} := [u_1, \cdots, u_h]$. For simplicity, let $u_i^j \in \mathbb{R}$ be the j-th entry in u_i. Then Problem 1 can be rewritten as

$$\begin{aligned}\min_{\mathbf{t}}\ & \mathbf{u}^T R \mathbf{u} \qquad (3a)\\ s.t.\ & A\mathbf{t} \leq b, -\bar{a} \leq u_i^j \leq \bar{a}, \forall i, \forall j = 1,2 \qquad (3b)\\ & f_i^j(\mathbf{t}) + h_i^j(\mathbf{t}) u_i^j = 0, \forall i, j, \qquad (3c)\end{aligned}$$

where $R = diag(w_1, w_2, w_3, w_4, w_5, w_1, \cdots, \cdots, w_5)$,

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 1 & \cdots & 1 & 1 \\ -1 & 0 & \cdots & 0 \\ -1 & -1 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ -1 & \cdots & -1 & -1 \end{bmatrix}, b = \begin{bmatrix} T_1^{max} \\ \vdots \\ T_h^{max} \\ -T_1^{min} \\ \vdots \\ -T_h^{min} \end{bmatrix}.$$



(a) The original constraint in (3c).
(b) The relaxed constraint in (4c).

Fig. 4: The relaxation of the optimization problem.

Moreover, from (1), we have

$$\begin{aligned}
f_i^1(\mathbf{t}) &= 2\left[t_i dp_{i-1}^{\tau_i} - t_{i-1} dp_i^{\tau_i}\right]\\
h_i^1(\mathbf{t}) &= t_i t_{i-1}(t_i + t_{i-1})\\
f_i^2(\mathbf{t}) &= 2\left[t_i dp_{i-1}^{\eta_i} - t_{i-1} dp_i^{\eta_i}\right]\\
h_i^2(\mathbf{t}) &= t_i t_{i-1}(t_i + t_{i-1})\\
f_i^3(\mathbf{t}) &= 6[(t_{i-1} + t_{i-2})t_{i-1}t_{i-2}dp_i^{\tau_i} - (t_i + 2t_{i-1}\\
&\quad + t_{i-2})t_i t_{i-2} dp_{i-1}^{\tau_i} + (t_i + t_{i-1})t_i t_{i-1} dp_{i-2}^{\tau_i}]\\
h_i^3(\mathbf{t}) &= t_i t_{i-1} t_{i-2}(t_i + t_{i-1})(t_i + t_{i-1} + t_{i-2})\\
f_i^4(\mathbf{t}) &= 6[(t_{i-1} + t_{i-2})t_{i-1}t_{i-2}dp_i^{\eta_i} - (t_i + 2t_{i-1}\\
&\quad + t_{i-2})t_i t_{i-2} dp_{i-1}^{\eta_i} + (t_i + t_{i-1})t_i t_{i-1} dp_{i-2}^{\eta_i}]\\
h_i^4(\mathbf{t}) &= t_i t_{i-1} t_{i-2}(t_i + t_{i-1})(t_i + t_{i-1} + t_{i-2})\\
f_i^5(\mathbf{t}) &= v^r t_i - dp_i^{\tau_i}\\
h_i^5(\mathbf{t}) &= t_i,
\end{aligned}$$

where $dp_i = p_{i+1} - p_i$, $dp_i^{\tau_j} = dp_i \cdot \tau(\theta_j)$, and $dp_i^{\eta_j} = dp_i \cdot \eta(\theta_j)$. By definition, $h_i^j \geq 0$ for all i and j. This property will be exploited to relax the problem. In order to compute the acceleration and jerk at $i = 1, 2$, constants p_0, p_{-1}, t_0 and t_{-1} are pre-defined according to the initial velocity v_0 and the initial acceleration $a_0 = 0$.

B. Quadratic Approximation Using the SCFS Algorithm

The SCFS algorithm is designed for problems with convex costs and nonlinear equality constraints. The idea is to 1) relax the nonlinear *equality* constraints to a set of non degenerating nonlinear *inequality* constraints using slack variables by exploiting symmetry of the problem, and 2) approximate the relaxed problem using quadratic programs. A set of nonlinear *inequality* constraints are non degenerating if they do not imply any nonlinear *equality* constraint.

Since Problem 1 is symmetric with respect to $\mathbf{u}$ in the cost function (3a) and in the feasibility constraints (3b), we define $\mathbf{y} \geq |\mathbf{u}|$ to be the slack variable. Then the relaxed problem is formulated below.

Problem 2 (The Relaxed Problem)**.**

$$\min_{\mathbf{t},\mathbf{y}} \ \mathbf{y}^T R \mathbf{y} \tag{4a}$$

$$s.t. \ A\mathbf{t} \leq b, y_i^j \leq \bar{a}, \forall i, \forall j = 1, 2 \tag{4b}$$

$$f_i^j(\mathbf{t}) + h_i^j(\mathbf{t}) y_i^j \geq 0, f_i^j(\mathbf{t}) - h_i^j(\mathbf{t}) y_i^j \leq 0. \tag{4c}$$

Geometrically, the nonlinear *equality* constraint shown as the nonlinear curve in Fig.4a is relaxed to two *inequality* constraints shown as the shaded area in Fig.4b. Problem 2 is equivalent to Problem 1 in the sense that: if $(\mathbf{t}^o, \mathbf{y}^o)$ is a local optimum of Problem 2, then $\mathbf{t}^o$ is a local optimum of Problem 1; and if $\mathbf{t}^o$ is a local optimum of Problem 1, then $(\mathbf{t}^o, \mathbf{y}^o)$ is a local optimum of Problem 2 with $(y_i^j)^o := |[h_i^j(\mathbf{t}^o)]^{-1} f_i^j(\mathbf{t}^o)|$.

The relaxed problem is solved iteratively. Suppose at the k-th iteration, we have a solution $\mathbf{t}^{(k)}$ and $\mathbf{y}^{(k)}$. Then at the $(k+1)$-th iteration, (4c) is convexified with respect to $\mathbf{t}^{(k)}$ and $\mathbf{y}^{(k)}$ [12]. In this paper, we simply linearize (4c) and obtain the following approximated quadratic program.

Problem 3 (The Approximated Quadratic Program)**.**

$$\min_{\mathbf{t},\mathbf{y}} \ \mathbf{y}^T R \mathbf{y} \tag{5a}$$

$$s.t. \ A\mathbf{t} \leq b, y_i^j \leq \bar{a}, \forall i, \forall j = 1, 2 \tag{5b}$$

$$\begin{aligned}
&\left[\nabla_{\mathbf{t}} f_i^j(\mathbf{t}^{(k)}) + \nabla_{\mathbf{t}} h_i^j(\mathbf{t}^{(k)})(y_i^j)^{(k)}\right](\mathbf{t} - \mathbf{t}^{(k)})\\
&+ h_i^j(\mathbf{t}^{(k)}) y_i^j + f_i^j(\mathbf{t}^{(k)}) \geq 0
\end{aligned} \tag{5c}$$

$$\begin{aligned}
&\left[\nabla_{\mathbf{t}} f_i^j(\mathbf{t}^{(k)}) - \nabla_{\mathbf{t}} h_i^j(\mathbf{t}^{(k)})(y_i^j)^{(k)}\right](\mathbf{t} - \mathbf{t}^{(k)})\\
&- h_i^j(\mathbf{t}^{(k)}) y_i^j + f_i^j(\mathbf{t}^{(k)}) \leq 0.
\end{aligned} \tag{5d}$$

Solving the above quadratic program, we obtain $\mathbf{t}^o$ and $\mathbf{y}^o$. Define $\mathbf{t}^{(k+1)} := \mathbf{t}^o$ and $(y_i^j)^{(k+1)} := |[h_i^j(\mathbf{t}^o)]^{-1} f_i^j(\mathbf{t}^o)|$. Then we iterate until the solution converges, e.g. $\|\mathbf{t}^{(k+1)} - \mathbf{t}^{(k)}\| \leq \epsilon$ for ϵ small. $\mathbf{t}^{(0)}$ and $\mathbf{y}^{(0)}$ is initialized as

$$t_i^{(0)} = \frac{dp_i^{\tau_i}}{v^*}, (y_i^j)^{(0)} := |[h_i^j(\mathbf{t}^{(0)})]^{-1} f_i^j(\mathbf{t}^{(0)})|, \tag{6}$$

where v^* can be the reference v^r or the initial speed v_0^τ.

It is shown in [12] that if (5c) and (5d) are subsets of (4c), then the iteration described above will converge to a local optimum of Problem 1. In practice, the iteration converges even if the condition does not hold, which requires further analysis in the future. Note that the idea of iteratively solving a quadratic subproblem is similar to that of the sequential quadratic programming (SQP). However, if we directly apply SQP on Problem 1 (directly linearizing the equality constraint (3c)) instead of relaxing Problem 1 to Problem 2, we may not approach a local optimum of Problem 1 without line search in each iteration. The key advantage of the SCFS algorithm is that no line search is needed. Thus the computational complexity is lowered and the computation time is reduced.

V. Performance

In this section, the performance of the proposed algorithm will be illustrated in several scenarios including the two cases in Fig.1. The parameters in Problem 1 were chosen to be $w_1 = 1$ and $w_2 = w_3 = w_4 = w_5 = 10$. The simulations were run in Matlab on a Macbook of 2.3 GHz using Intel Core i7. In the SCFS algorithm, Problem 3 for each iteration was solved using the `quadprog` function. For comparison, Problem 1 was also solved using the SQP in the `fmincon` function. The SCFS algorithm and the SQP algorithm terminated if the step size (difference between the two consecutive solutions) was less than 10^{-6}. Note that the computation time of the algorithms shown in this paper is

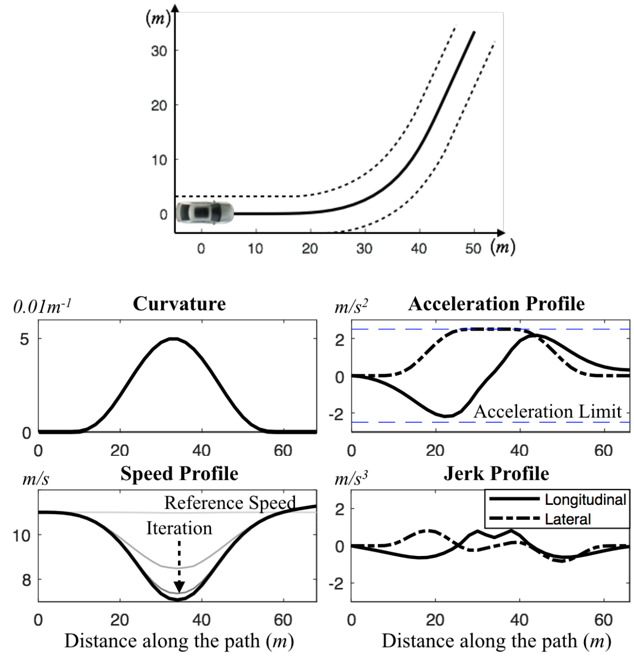


Fig. 5: The scenario and the result in case 0.

just for comparison. The computation time may be greatly reduced if we run the algorithm in more efficient languages such as C++, which will be pursued in the future. In this paper, the efficiency of the proposed algorithm can also be demonstrated though the reduced number of iterations required for convergence. For better illustration, the speed profiles after every iteration are shown in grayscale (the later in the iteration, the darker) together with the optimal speed profile in all three cases.

A. Case 0: Speed Profile for a Curvy Road

In this scenario, the automated vehicle needs to pass a curvy road shown in Fig.5. The reference speed is $v_r = 11m/s$. The vehicle's initial speed is $v_0^\tau = 11m/s$ and $v_0^\eta = 0m/s$. The path is sampled every 2m and 35 points are chosen. The SCFS algorithm (iteratively solving Problem 3) converges after 5 steps with total computation time 0.185s. The optimal speed, acceleration and jerk profiles are shown in Fig.5. The horizontal axes in the plots represents the traveling distance along the path. The vehicle decelerated first in order to meet the acceleration constraint in the lateral direction. It then accelerated after the maximum curvature was reached. The SQP algorithm converges to the same speed profile after 80 iterations with total computation time 54.479s.

B. Case 1: Speed Profile for Overtake

The scenario is shown in Fig.1a where the automated vehicle wants to overtake the slow front vehicle using the opposite lane. The reference speed is $v_r = 11m/s$. The vehicle's initial speed is $v_0^\tau = 10m/s$ and $v_0^\eta = 0m/s$. $\mathbf{t}^{(0)}$ is initialized using v_0^τ. The path is sampled every 2m and 34 points are chosen. The SCFS algorithm converges at iteration 5 with computation time 0.308s. The optimal speed, acceleration and jerk profiles are shown in Fig.6. The horizontal axis in the plots represents the traveling distance along the lane. The corresponding time stamps for all stations are shown in Fig.7. Snapshots are also shown in Fig.7 where the gray rectangle represents the ego vehicle and the yellow rectangles are the surrounding vehicles. The ego vehicle slowed down first to keep a safe headway from the front vehicle V1. When it changed to the adjacent lane, it speeded up to overtake V1. Before the vehicle V2 in the opposite direction came, the ego vehicle went back to its lane. The optimal speed profile is on the boundary of the safety constraint as shown in Fig.7 and on the boundary of the feasibility constraint as shown in the acceleration profile in Fig.6. For comparison, SQP method converges to the same optimum after 48 iterations with computation time 28.513s.

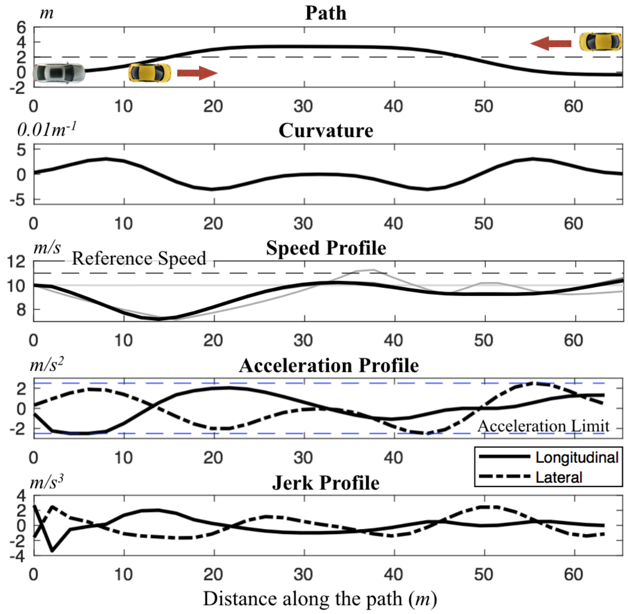


Fig. 6: The scenario and the result in case 1.

C. Case 2: Speed Profile for Right Turn

The scenario is illustrated in Fig.1b where the automated vehicle tries to turn right in green light when a vehicle in the opposite direction turns left and a pedestrian is crossing the street. The reference speed is $v_r = 5m/s$. The vehicle's initial speed is $v_0^\tau = 2.5m/s$ and $v_0^\eta = 0m/s$. $\mathbf{t}^{(0)}$ is initialized using v_0^τ. The path is sampled every 0.5m and 36 points are chosen. The strategy determined by the high level planner for the ego vehicle is to pass the conflict zone after the pedestrian at $T = 5s$ and before the left-turn vehicle at $T = 6s$. To meet the safety constraint, the ego vehicle slowed down to yield the pedestrian and speeded up to pass the conflict zone before the left-turn vehicle, as shown in the speed profile in Fig.8. For comparison, the optimal speed profile without the safety constraint is shown in Fig.9 where the ego vehicle turned smoothly. The SCFS algorithm converges after 8 iterations with computation time 0.522s. The SQP algorithm does not converge before the maximum number of iterations 100 is reached.

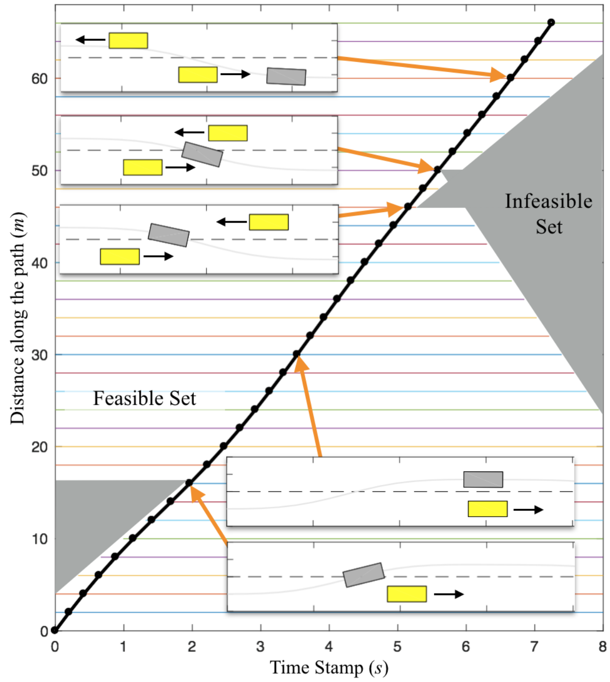


Fig. 7: The optimal time stamps in case 1.

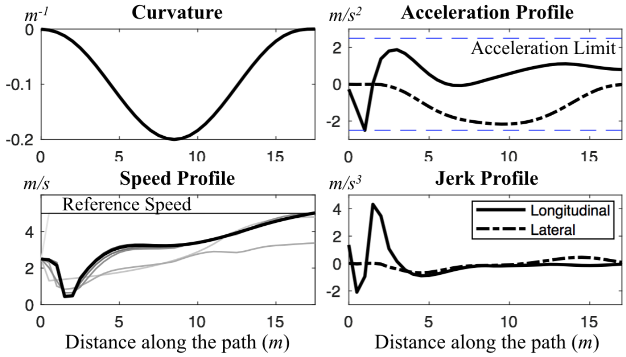


Fig. 8: The optimal speed, acceleration and jerk profiles in case 2.

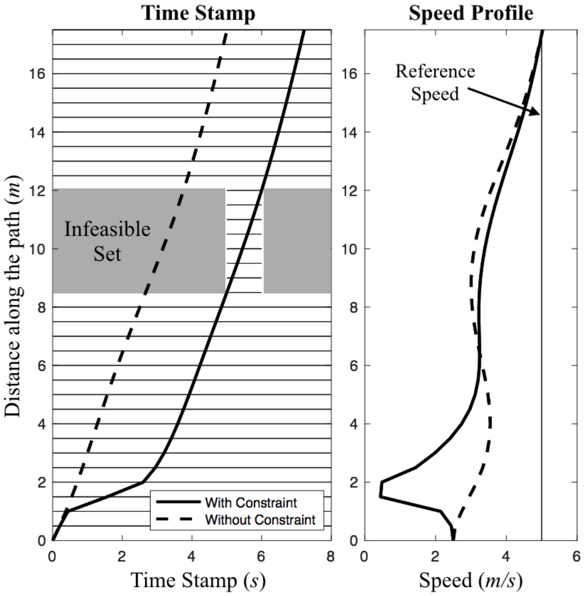


Fig. 9: Comparison between the speed profiles with and without the temporal constraint in case 2.

VI. Conclusion

In this paper, the one dimensional speed profile planning problem is considered. A temporal optimization framework is proposed which comprehensively evaluates driving quality such as time-efficiency and comfort, feasibility such as acceleration limits, as well as safety during interactions with other road participants. To speed up computation, the non-convex temporal optimization problem is approximated by a sequence of quadratic programs and then solved iteratively using the slack convex feasible set algorithm. Several case studies are provided to illustrate the effectiveness of the method. In the future, the proposed method for speed profile planning will be integrated into a layered trajectory planning framework.

References

[1] M. McNaughton, C. Urmson, J. M. Dolan, and J. W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4889–4895.

[2] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1879–1884.

[3] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha - A local, continuous method," in *IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 450–457.

[4] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 250–256.

[5] T. Gu, J. M. Dolan, and J.-W. Lee, "Runtime-bounded tunable motion planning for autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1301–1306.

[6] X. Li, Z. Sun, Z. He, Q. Zhu, and D. Liu, "A practical trajectory planning framework for autonomous ground vehicles driving in urban environments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1160–1166.

[7] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using Bézier curves and MPC," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 826–833.

[8] W. Xu, J. Wei, J. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2061–2067.

[9] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1010–1016.

[10] D. González, V. Milanés, J. Pérez, and F. Nashashibi, "Speed profile generation based on quintic Bézier curves for enhanced passenger comfort," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 814–819.

[11] J. van den Berg, "Extended LQR: locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost," in *Robotics Research*. Springer, 2016, pp. 39–56.

[12] C. Liu and M. Tomizuka, "Geometric considerations on real time trajectory optimization for nonlinear systems," *System & Control Letters*, p. in review, 2016.

[13] W. Zhan, J. Chen, C.-Y. Chan, C. Liu, and M. Tomizuka, "Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, p. to appear.