Toward Efficient Trajectory Planning based on Deterministic Sampling and Optimization

Yang Wang
China North Vehicle Research Institute
Unmanned Ground System Research Center
Beijing China
yangwangeducn@gmail.com

Shengfei Li
China North Vehicle Research Institute
Unmanned Ground System Research Center
Beijing China
lishengfeizdh@126.com

Wen Cheng
China North Vehicle Research Institute
Beijing China
wencheng.edu.cn@gmail.com

Xing Cui
China North Vehicle Research Institute
Beijing, China
shugu_bit@126.com

Bo Su
China North Vehicle Research Institute
Beijing, China
bosu@noveri.com.cn

2020 Chinese Automation Congress (CAC) | 978-1-7281-7687-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/CAC51589.2020.9327252

***Abstract*—The solution of optimization-based planners depends heavily on a good initialization and their run-time is often non-deterministic, especially in dense obstacle fields. Sampling-based planning, whether probabilistic or deterministic, is a well-established method for exploring the search space. However, the downsides are also obvious: potentially intractable computational overhead, the curse of dimensionality and the sub-optimality due to discretization. Motivated by this observation, this paper introduces a real-time trajectory planning algorithm based on the combination of sampling and optimization approaches, which is applicable to autonomous vehicles operating in highly constrained environments. A maximum corridor width region and initial drivable path are firstly extracted from deterministic sampling. Then the initial path is further optimized through a splined-based quadratic programming and appended with a speed profile. This planner is scalable to both high-speed off-road scenarios and structured urban driving.**

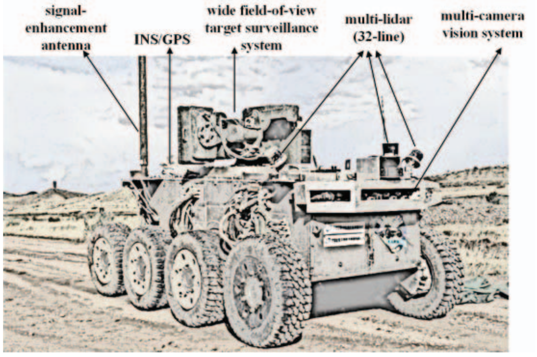
Keywords—Autonomous vehicle, real-time trajectory planning, quadratic programming (QP)

I. Introduction

In July 2019, an all-terrain vehicle Mission-Terminator (Fig. 1) completed a task-based off-road Unmanned System Challenge fully autonomously and won the first place[1]. The vehicle demonstrated the ability to complete an integrated task in desert grassland environment, while safely interacting with other human driven vehicles.

In this paper, we describe the design and implementation of an efficient and reliable trajectory planning which is used for this challenge, based on sampling and optimization. To cope with moving obstacles and several dynamic constraints, speed profile is further generated. In particular, we present a hybrid planning approach by generating a coarse initial path via drivable edge search on a 2-D corridor-like graph constructed from global reference centerline, and then locally optimizing with a spline-based QP optimizer. This centerline represents the result of a global planning algorithm for unstructured environments, which is not the scope of this paper.

II. Related work



signal-enhancement antenna
INS/GPS
wide field-of-view target surveillance system
multi-lidar (32-line)
multi-camera vision system

Fig. 1 Mission-Terminator is equiped with several LIDARs, muti-camera vision system, and a high-accuracy ING/GPS system.

Since the 2004 DARPA challenges, steadily increasing research efforts, from academia, industry, and governments, have been devoted to developing driverless vehicle technology. These developments have been fueled by recent advances in sensing and artificial intelligent (AI). Notable examples include the Intelligent Vehicle Future Challenges from 2009 to 2013 [2], the Bertha-Benz-Memorial-Route autonomous drive [3], the google self-driving car, Tesla's Autopilot system and Baidu Apollo project [4]. Our focus in this paper is on the trajectory planning aspects for autonomous vehicles.

Several methods for trajectory planning have been proposed [5],[6],[7] that find a feasible trajectory by taking into account all constraints from vehicle non-holonomic, dynamics and obstacles detected by on-board sensors. These methods can mainly categorized into two fundamental classes: the optimization-based and the sampling-based. Ziegler [3] proposed a local continuous approach by using SQP to solve a constraint-based non-linear optimization problem. But special attention has been paid to the initial guess of the trajectory and careful obstacle constraints handling. The lack of global awareness of optimization-based methods depends heavily on a good initialization trajectory, and often only return a suboptimal solution that is close to the initialized trajectory. Indeed, the solution of pure optimization-based approaches may even cause the planner to converge to a wrong local minimum [8]. The iteration process of optimization with complex objective and cost functions also implies a large computation overhead.

A challenge task for autonomous driving is to consider aspects related to optimality such as operational limits of vehicles (trajectory curvature, deviation from speed limit, accelerations, comfort, energy efficiency and various environmental elements, etc.). As a result, optimization-based approaches are intrinsically more suitable than sampling-based ones in finding optimal trajectories in continuous space [9].

Sampling-based method typically discretizes the continuous space to convert planning into a graph search problem and generate a large set of candidate trajectories by probabilistically or deterministically sampling. Probabilistic methods build the search graph by adding nodes according to a certain statistical distribution, whereas the deterministic methods perform sampling in a predefined way. The probabilistic roadmap (RPM), the rapid-exploring random tree (RRT) and some modification variants of them represent one of the successful algorithms due to their probabilistic completeness [10], [11]. Their randomization aspect, however, makes safety-critical applications and online computation efficiency very challenging. Deterministic sampling-based methods are good alternatives to the probabilistic ones. Theoretical results from [12], [13] even demonstrate the superior performance of using low-dispersion deterministic sampling strategies. On the other hand, the drawbacks are also obvious : the potentially intractable computational overhead, the curse of dimensionality and the sub-optimality due to discretization.

The two classes have both strengths and limitations. A possible solution of utilizing the benefits of both approaches is very promising. The on-road planning in [14] sampled a set of candidate trajectories by edge-augmented search-based path nudging and then used pure-pursuit controllers to optimize the selected coarse path. However, the path nudging cost terms depend on neighboring states, which introduces strong non-convexity and non-Markovian process. Although a directed acyclic graph (DAG) can be constructed by edge-augmented formulation, the full sampled configuration space is also time-consuming. An integration of a sampling and an optimization method was also proposed in [15], which sampled various behavioral trajectory candidates, picked the best one based on a performance index and resampled the behavioral trajectory for initial state to reduce the computation of the final trajectory optimization SQP step. The numerous sampling evaluation and collision-checking along with the complex SQP formulation causes parameter tunability and stability issues.

In this work, we resort to a hybrid trajectory planning scheme where retaining the ability to converge to an exact optimum of the optimization-based planner as well as the global awareness of the sampling-based method. Specifically, a maximum corridor width region and initial drivable path are firstly extracted from deterministic sampling. Then the initial path is further optimized through a splined-based QP and appended with a speed profile. In particular, the QP refines a solution within a drivable region around the current trajectory. Hence, there is no inherent inaccuracy due to limited resolution of search space, and the QP formulation is computationally efficient.

III. Spatial Region Sampling in the Frenet Frame

An important part of planning is to determine the drivable space (i.e. the equivalent obstacle-free space) within a bounded corridor, which is typically modelled by a sequence of waypoints representing the centerline of the corridor. The centerline waypoints are defined in the global frame. For drivable space abstraction, all obstacles information is defined in the local body frame and need to be projected on the Frenet frame. Static obstacles can be directly projected, while the moving obstacles can be described with an predicted moving trajectory. Fig. 2 shows different information expressed in these three frames.

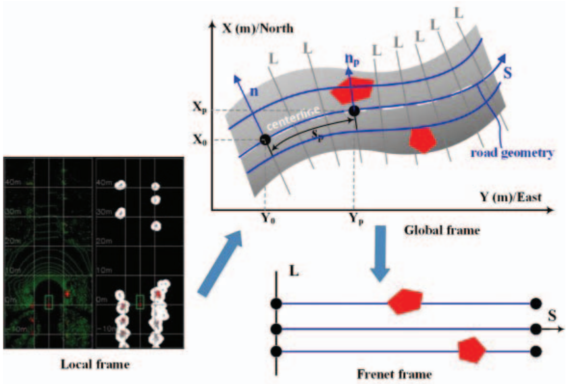
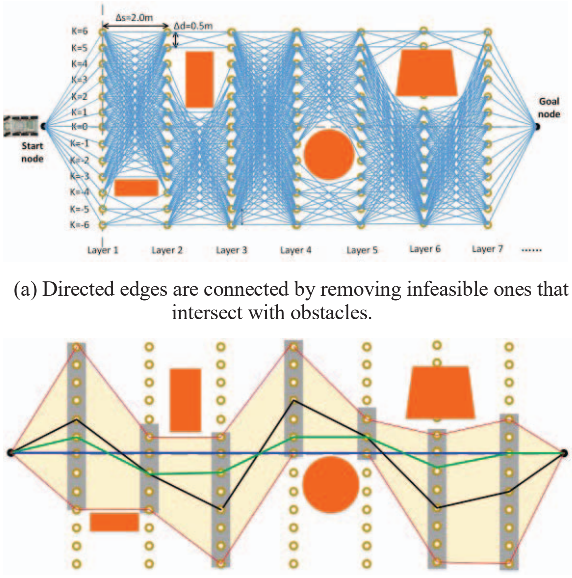


Fig. 2. The vehicle local frame is an integration of odometry, INS/GPS and environmental perception information. The centerline (reference path) is defined in the global frame. All obstacles and environment information are projected on the Frenet frames. Trajectory planning problem is formulated in the dynamic Frenet frame.



(a) Directed edges are connected by removing infeasible ones that intersect with obstacles.

(b) Maximum corridor width region and initial drivable path determination.

Fig. 3. Illustration of searching space segmentation and drivable region abstraction. The initial path is selected within drivable region based on simple cost computation.

A discrete graph consisting of an array of multiple node layers is constructed by laterally shifting the waypoints from the centerline (Fig. 3). Infeasible spatial edges that cause collision with any obstacles, will be removed. The four parameters that define the graph construction are listed below:

- Longitudinal interval ΔS: 2m.
- Lateral range L: 6m
- Lateral interval ΔL: 0.5m.
- Longitudinal planning distance S: 50m.

These specific values are in general determined by trial and error process. But the longitudinal planning distance is closely related to the sensing horizon.

There are multiple topologically distinct sub-graphs that allows the planning to navigate freely within them. Evaluation of these candidate homotopic sub-graphs is performed using geometric intuition. The optimal drivable space can be selected primarily according to maximum corridor width criteria, which is a good indicator of traversability (Fig. 3(b)). This drivable space forms a feasible tunnel, which is the boundary constraints used for later optimization. The solution of optimization-step within this tunnel often yield a result that is closer to a global optimum.

It is natural to use deterministic sampling to find an initial path the belongs to the drivable space. Two simple aspects considered here for shape the path: (1) the closeness to the centerline and (2) the distance between current node and the obstacle. The overall path smoothness can be left for subsequent optimization. The optimal sequence of nodes $\{x_i^*\}$is defined to be optimal such that the following cost function is minimized :

$$\{\boldsymbol{x}_i^*\} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}} \textstyle\sum_i w_{cl} \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_{i0}\|^2 + w_{obs} \cdot d \qquad (1)$$

where $\boldsymbol{x}_i$ is the node within the drivable space, $\boldsymbol{x}_{i0}$ represents each waypoint along the centerline, and d is the distance associated with obstacles (Fig. 4). Denote $d_m = \|\boldsymbol{x}_i - \boldsymbol{x}_{obs}\|$ as the distance between the current node position and obstacle center. d_n is set to the obstacle-expanded zone width for safety considerations, and d_c is radius of the obstacle zone.

$$d = \begin{cases} 0, & d_m > d_n \\ d_m - d_c, & d_c \le d_m \le d_n \\ d_{collision} & d_m < d_c \end{cases} \qquad (2)$$

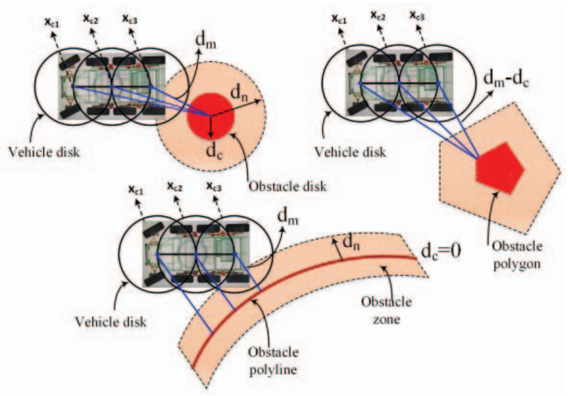


Fig. 4. The calculation of the distance associated with obstacles. The cases with respect to disk and polyline are relatively easy to compute. The more general case with respect to polygon can be calculated in an easy approximated way or by introducing a pseudo distance that has continuously differentiable in [3].

The $d_{collision}$ in (2) is the collision distance, which is set to a very large value that helps to eliminate infeasible paths.

The optimal cost-to-go searching problem can be efficiently solved by performing dynamic programming in a backward recursive fashion. The goal is to generate a candidate path that keeps safety distance from obstacles and tracks the road centerline, leading to a tradeoff between centerline deviation weight w_{cl} and obstacle repulsion weight w_{obs}. The weights should be tuned neither too low, nor too high. As shown in Fig. 3(b), different weights cause different paths. In fact, a simple rule-based selection will find an satisfactory solution in most cases, such as always selecting the middle-node in each layer of the drivable space. Hence, in this case, no graph search algorithm is necessary.

The path generated so far is coarse piecewise linear trace which is not guaranteed to be dynamically feasible and non-smooth. The points between different layers are then smoothly connected by cubic Bezier curves. An analytical continuous-curvature path-smoothing algorithm [16] is adopted here. This smoothing process is based on parametric cubic Bezier curves, and the algorithm naturally satisfies non-holonomic constraints. Several characteristic lengths define the cubic Bezier spiral segments (Fig. 5), which are given as:

$$\left\{\begin{gathered} d_1 = d_2 = d, \beta = \gamma/2 \\ c_1 = 7.2364 \\ c_2 = 2\left(\sqrt{6} - 1\right)/5 \\ c_3 = (c_2 + 4)/(c_1 + 6) \\ g_b = g_e = c_2 c_3 d \\ h_b = h_e = c_3 d \\ k_b = k_e = [(6c_3 \cos\beta)/(c_2 + 4)]d \end{gathered}\right. \qquad (3)$$

where the design parameter d can be determined by the maximum curvature constraint k_{max} as follows:

$$d = d_{kmax} = \frac{c_4 \sin\beta}{k_{max} cos^2 \beta} \qquad (4)$$

where $c_4 = (c_2 + 4)^2/54c_3$.

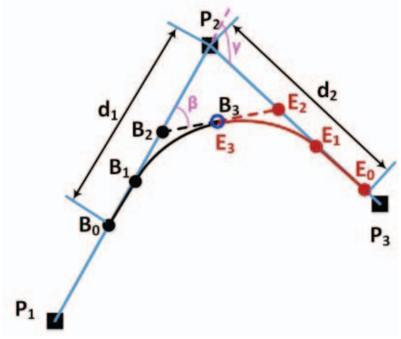


Fig. 5. Illustration of two geometrically symmetrical cubic Bezier curves and characteristic lengths.

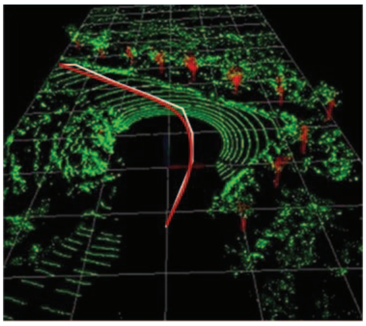


Fig. 6. Initial piecewise linear path (white curve) and smoothed path that satifying the upper bounded curvature constraint (red curve).

Fig. 6 shows the result of path-smoothing algorithm generating a continuous curvature path and satisfying the upper maximum curvature constraint. The white piecewise linear path in Fig. 6 is the initial result from the search problem defined in (1), and the red path represents the smoothed result.

The first step, described in this section, uses search-and-smooth that produces an initial feasible path. While lacking theoretical optimality guarantees, this initial path typically lies in a neighborhood of the global optimum and hence in practice even this initial path can also be directly used for some scenarios.

IV. Path Optimization with Boundary Constraints

The second step, described in this section, uses a splined-based QP method to locally improve the quality of the previous solution, shaping a path that is close to natural driving style of human-driven behavior. In Section III, the selected drivable space forms a feasible tunnel (Fig. 7), within which an optimal path is generated by the following QP step.

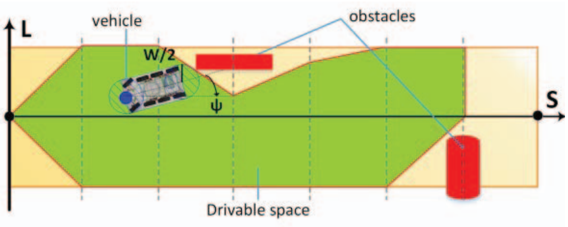


Fig. 7. The optimal path must obey a set of contraints, among which drivable space imposes the boundary constraints.

The QP step computes an optimal path $\left(s, f(s)\right)^T$ for the rear axle center point of the vehicle. The optimization programming to solve the optimal solution by minimizing the integral

$$J\left(f(s)\right) = \int_{s_0}^{s_0+S} L(s, \dot{s}, \ddot{s}, \dddot{s}) ds, \tag{5}$$

with

$$L = j_{offs} + j_{head} + j_{curv} + j_{dcurv}$$

where the arc length s_0 is the current position and S is the preview horizon, which is chosen according to sensing range. The individual summands of the integrand L are expressed as

$$\begin{aligned} j_{offs} &= w_{offs}|f(s) - f_0(s)|^2 \\ j_{head} &= w_{head}\left|\dot{f}(s)\right|^2 \\ j_{curv} &= w_{curv}\left|\ddot{f}(s)\right|^2 \\ j_{dcurv} &= w_{dcurv}\left|\dddot{f}(s)\right|^2 \end{aligned} \tag{6}$$

where $f_0(s)$ is the initial path from section III, j_{offs} is the term to penalize deviation from the initial path, j_{head}, j_{curv} and j_{dcurv} are related to the heading, curvature and derivation of curvature. The objective function describes the balance between the navigation and smoothness. The cost function in (6) is similar to that used by [3] who showed the driving style was described as pleasant and natural.

The optimal path must minimize the cost function (6), but, at the same time, satisfy a set of constraints. The constraints include boundary constraints and dynamic feasibility.

Boundary constraints result from the vehicle kinematics. It is not sufficient to only consider a feasible range for L $= f(s)$. The heading of current vehicle also matters. As shown in Fig.7, we add two half circles on the front and rear axle of the vehicle. Denote the distance from front axle to rear axle as l_d and the vehicle width as l_w. Then, the lateral position of the front-left corner is given as:

$$l_{front-left-corner} = f(s) + l_d \sin\psi + l_w/2$$

where ψ is the heading difference between vehicle and road direction. Since ψ is generally small, $\sin\psi \leq tan\psi \approx \dot{f}(s)$, the constraint can be approximated by the following inequality:

$$\begin{aligned} f(s) + l_d \sin\psi + l_w/2 \leq f(s) + l_d\dot{f}(s) + l_w/2 \\ \leq l_{front-left-bound} \end{aligned} \tag{7}$$

Dynamic feasibility constraints imposed by vehicle's initial condition, final condition and intermediate condition on their derivatives. Since $f(s)$ is piecewise polynomials, it requires continuous derivative matching at joint knots. These constraints can be written as equality constraints.

The spline curve $f(s)$ involves at least third-order derivative information in cost function. In order to avoid over-constraint problem, the polynomial order m is set to 5, i.e. **quintic** polynomials. It is convenient to write $f(s)$ as piecewise polynomial functions over n arc intervals as:

$$f(s) = \begin{cases} \sum_{i=0}^{m} p_{i1}s^i & s_0 \leq s < s_1 \\ \sum_{i=0}^{m} p_{i2}s^i & s_1 \leq s < s_2 \\ \vdots & \vdots \\ \sum_{i=0}^{m} p_{in}s^i & s_{n-1} \leq s < s_n \end{cases} \tag{8}$$

The coefficient vector is denoted as $\mathbf{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots \boldsymbol{p}_n)^T$, where $\boldsymbol{p}_j = \left(p_{0j}, p_{1j}, \cdots, p_{mj}\right)^T$. We can formulate the problem as a quadratic programming by writing $\mathbf{p}$ as a $4(m+1)n \times 1$ vector c with decision variables c $= \{p_{ij}\}$:

$$\begin{aligned} &\underset{c}{\text{argmin}}\ \mathbf{p}(f) = \textstyle\sum_{k=0}^{3} w_k \boldsymbol{p}_k(f) \\ s.t.\ &L(s) \leq f(s) + l_d\dot{f}(s) \leq U(s) \\ &Ac = b \end{aligned} \tag{9}$$

Here the objective function incorporates the minimization of the functional expressed in (5), while the first constraints represent the boundary constraints in (7) and the equality constraint $Ac = b$ expresses piecewise polynomials must be connected smoothly.

The functional $\mathbf{p}(f)$ defined in (9) has a quadratic form, and the constraint functional has a linear form with respect to parameter $\{p_{ij}\}$. Let us step further to gain this insight. Each $\boldsymbol{p}_k(f)$ can be represented as a summation of integrations:

$$\begin{aligned} \boldsymbol{p}_k(f) &= \textstyle\sum_{j=0}^{n-1} \int_{s_j}^{s_{j+1}} \left(f^{(k)}\right)^2 ds \\ &= \textstyle\sum_{j=0}^{n-1} \int_{s_j}^{s_{j+1}} p_j^T s_{(k)} {s_{(k)}}^T p_j ds \\ &= p^T S_{(k)} p \end{aligned} \tag{10}$$

where $s_{(k)}$ is the k-th-order derivative coefficient of f with respect to parameter p at s ∈ $[s_j, s_{j+1})$, which is given as:

$$\begin{aligned} s_{(0)} &= (1, \tilde{s}, \tilde{s}^2, \dots, \tilde{s}^m)^T \\ s_{(1)} &= (0,1,2\tilde{s}, \dots, m\tilde{s}^{m-1})^T \\ s_{(2)} &= (0,0,2, \dots, m(m-1)\tilde{s}^{m-2})^T \\ s_{(3)} &= (0,0,0, \dots, m(m-1)(m-2)\tilde{s}^{m-3})^T \end{aligned}$$

$S_{(k)}$ is defined as the block diagonal matrix:

$$S_{(k)} = Diag\left(\int_{s_0}^{s_1} s_{(k)}{s_{(k)}}^T ds, \dots, \int_{s_{n-1}}^{s_n} s_{(k)}{s_{(k)}}^T ds\right)$$

Thereby, the functional $\mathbf{p}(f)$ has a quadratic form and all constraints are linear with respect to decision variables. Optimization with a quadratic convex objective functional and linear constraints can be solved very fast, which guarantees the real-time planning.

The path optimization formulated is a single weighted cost function. The continuous nature of the cost terms and objective functions, and use the QP optimizer makes it easier to observe a gradual change of path shape due to weights tuning. Insights from classic linear quadratic (LQ) optimal control can be used to guide the weights selection. In practice, the weights listed in (6) are physically interpretable and thus can be easily tuned to avoid overshoot or oscillation.

V. Speed Planning

The speed planning generates a speed profile subjected to several dynamic constraints. With the previous sampling-and-optimization algorithm, we obtain a continuous path, which is made up of a functional $f(s)$ parameterized by arc-length s. The goal of speed planning is to determine a scalar speed function v parameterized by s, i.e. $v(s)$. Speed profile is essential for the driving quality and safety. Several dynamic constraints will be imposed on the profile, including maximum allowed speed, maximum lateral and longitudinal acceleration, limited jerk, and so on.

To facilitate incremental implementation of velocity profile generation, each constraint will be represented in finite-difference forms. For each iteration, the maximum allowed lateral acceleration a_{max}^{lat} constraint is applied firstly to bound the speed at each point along path:

$$v_i \leq min\left\{\sqrt{a_{max}^{lat}/|\kappa_i|}\,, v_{max}\right\} \tag{11}$$

where κ_i is the curvature information extracted from the previous planner, v_{max} is the maximum speed extracted from the digital map of high-level behavioral planner.

The lateral acceleration must be limited to a level that preserves lateral stability. The longitudinal acceleration and jerk also exert considerable influence on the riding quality and steering controllability. Preferred longitudinal acceleration a_{max}^{lon} and deceleration d_{max}^{lon} are given as follows:

$$-d_{max}^{lon} \leq \frac{v_i^2 - v_{i-1}^2}{2|s_i - s_{i-1}|} \leq a_{max}^{lon} \tag{12}$$

The next step is to apply the jerk constraint to each speed candidate. Excessive longitudinal jerk may induce frequent high-intensity braking, which can lead to dynamical impact and fatigue damage. Therefore, a maximum longitudinal jerk j_{max}^{lon} should be further enforced. In order to calculate current jerk, we use a parametric velocity represented by a quadratic polynomial of arc-length :

$$\begin{gathered} v = \alpha \cdot s^2 + \beta \cdot s + \gamma \\ a = 2\alpha \cdot sv + \beta \cdot v \\ j = 2\alpha \cdot (v^2 + sa) + \beta \cdot a \end{gathered} \tag{13}$$

where s is the arc-length, the speed v is derived through derivation of s w.r.t. time, we can further obtain the analytical form of acceleration a and jerk j by taking the first and second derivative of velocity.

Now, assuming that we have three speed points v_{i-2}, v_{i-1}, v_i at their corresponding arc-length points s_{i-2}, s_{i-1}, s_i, we can easily calculate all the coefficients α_i, β_i and γ_i by solving the following linear equation:

$$\begin{bmatrix} s_{i-2}^2 & s_{i-2} & 1 \\ s_{i-1}^2 & s_{i-1} & 1 \\ s_i^2 & s_i & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha_i \\ \beta_i \\ \gamma_i \end{bmatrix} = \begin{bmatrix} v_{i-2} \\ v_{i-1} \\ v_i \end{bmatrix} \tag{14}$$

The numerical approximation of current point jerk j_i can be uniquely solved. Then the current velocity point v_i is adjusted to satisfy the bounded jerk constraint, which is given as:

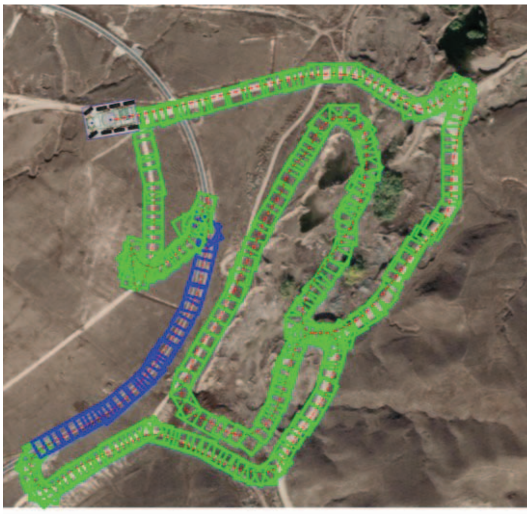
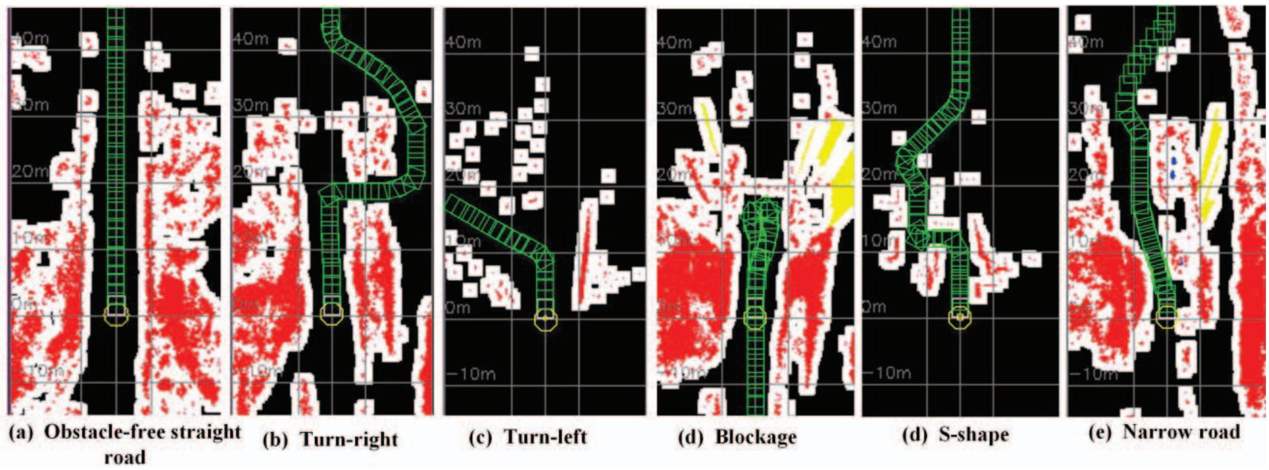


Fig. 8. The performance of the proposed trajectory planning and the integrated unmanned system of Misssion-Terminator were also fully tested on a competition event named “Integrated Autonomous task”. The task route ploted above includes 17% paved road (blue trajectory) and 83% off-raod (green trajectory).



(a) Obstacle-free straight road (b) Turn-right (c) Turn-left (d) Blockage (d) S-shape (e) Narrow road

Fig. 9. Examples of trajectories generated by the planner and driven by the autonomous vehicle in “Integrated Autonomous task” competition. The six snapshots represented the typical scenarios for planning. All the snapshots show a map, the vehicle and the traversable trajectory.

$$|j_i| \leq j_{max}^{lon} \quad (15)$$

The proposed speed-constraint-based planning approach is very computationally efficient. The speed profile is iteratively modified by taking all constraints into account at every cycle until the profile no longer changes.

VI. Results

Various field tests on our autonomous vehicle have been conducted. We used the following parameters for our planner: the perception grid map was of size 65m × 30m with 0.2 resolution; trajectory planning algorithm used a grid of size 50m × 6m with 0.5m lateral resolution, 2m longitudinal resolution, which is sufficient in unstructured terrain. Typical runtimes for a full planning cycle involving the deterministic sampling, continuous-curvature path-smoothing, QP optimization, and speed planning were on the order of 15ms-50ms.

Fig. 8 depicts an integrated autonomous task trajectory driven by Mission-Terminator in the National Unmanned System Challenge. This competition consisted of positive & negative obstacles, human-driven vehicles, curvy S-shape road, blockage road, parking lot, each focusing on testing different capabilities. Coordinated with global planning and behavior decision, the proposed algorithm was not tuned to any specific test area in this competition, showing the generality of the approach. Mission-Terminator was one of the five vehicles that completed all missions, finishing with the shortest time and the highest score.

The occupant grid map is used to model the environmental information, where the color grids represent obstacle data from Lidar-camera fusion. Fig. 9 shows that the trajectory planning generates a set of trajectories that attempt to follow the global centerline while also allowing for local obstacle avoidance and re-planning. Several typical scenarios are chosen to illustrate the capabilities of the proposed planner. The planner is able to generate obstacle-free trajectory in various roads.

Note that the lateral range of the sampling-space can be modified according to environmental characteristics. For example, the lateral range can be tuned to increase searching space when in dense obstacle fields. Shown in Fig. 9 (d) is a situation taken from S-shape off-road environment, the most difficult passage of the competition. The perception map displays complex obstacles forming a high dimensional constraint space, in which direct optimization-based approach does not work. Note that it is not always necessary to take exactly the every step of the planner. Generally, in simple straight road or other obstacle-free scenarios, the sampling and smoothing steps is sufficient. Mission-Terminator demonstrated flawless performance in complex general path-planning tasks, such as navigating parking lots, executing flexible turns, dealing with blocked roads and intersections with human-driven vehicles.

VII. Conclusions

A capable trajectory planner plays a crucial role in autonomous driving development. This paper presented a hybrid planning framework that uses the combination of sampling-based and optimization-based methods to preserve the benefits of both: sampling-based for initialization and avoiding local optimality, and optimization-based for local refinement. The proposed main steps are computationally efficient., including deterministic sampling with finite resolution, an analytical path-smoothing, and optimization with a quadratic convex objective and linearized constraints. This trajectory planning algorithm has proven very capable of tackling the various challenges encountered in a competition event named “Integrated Autonomous task”. The field tests demonstrate the safety and execution-feasible trajectory generation showing promise of this approach as a real-time planner scheme.

Acknowledgment

This work is supported by the National Natural Science Foundations of China (Grant No. 91748211).

References

[1] Wang, Yang , et al. "Path tracking of eight in-wheel-driving autonomous vehicle: controller design and experimental results." 2019 IEEE International Conference on Unmanned Systems (ICUS) IEEE, 2019.

[2] J. Xin, C. Wang, Z. Zhang, and N. Zheng, “China future challenge: Beyond the intelligent vehicle,” IEEE Intell. Transp. Syst. Soc. Newslett, vol. 16, pp. 8–10, 2014.

[3] Ziegler, Julius , et al. "Trajectory Planning for BERTHA - a Local, Continuous Method." IEEE Intelligent Vehicles Symposium IEEE, 2014.

[4] Fan, Haoyang , et al. "Baidu Apollo EM Motion Planner." (2018).

[5] Paden, Brian , et al. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles." IEEE Transactions on Intelligent Vehicles (2016).

[6] Badue, Claudine , et al. "Self-Driving Cars: A Survey." (2019).

[7] D. González, et al. "A Review of Motion Planning Techniques for Automated Vehicles." IEEE Transactions on Intelligent Transportation Systems 17.4(2016):1135-1145.

[8] Gu T. Improved Trajectory Planning for On-Road Self-Driving Vehicles Via Combined Graph Search, Optimization & Topology Analysis[D]. Carnegie Mellon University, 2017.

[9] Qian, Xiangjun , et al. "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective." 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (2016):205-210.

[10] Sertac Karaman, and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. Sage Publications, Inc. 2011.

[11] Du, Mingbo , et al. "An improved RRT-based motion planner for autonomous vehicle in cluttered environments." 2014 IEEE International Conference on Robotics and Automation (ICRA) IEEE, 2014.

[12] Lavalle, Steven M. , M. S. Branicky , and S. R. Lindemann . "On the Relationship between Classical Grid Search and Probabilistic Roadmaps." International Journal of Robotics Research 23.7-8(2004):673-692.

[13] Lucas Janson, Brian Ichter, and Marco Pavone. Deterministic Sampling-Based Motion Planning: Optimality, Complexity, and Performance. Robotics Research. 2018.

[14] Gu, Tianyu , et al. "Tunable and stable real-time trajectory planning for urban autonomous driving." IEEE/RSJ International Conference on Intelligent Robots & Systems, 2015:250-256.

[15] Lim, Wonteak , et al. "Hierarchical Trajectory Planning of an Autonomous Car Based on the Integration of a Sampling and an Optimization Method." IEEE Transactions on Intelligent Transportation Systems 19.2(2018):613-626.

[16] Yang, K. , and S. Sukkarieh . "An Analytical Continuous-Curvature Path-Smoothing Algorithm." IEEE TRANSACTIONS ON ROBOTICS 26.3(2010):561-568.