2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

A Sampling-Based Local Trajectory Planner for Autonomous Driving along a Reference Path

Xiaohui Li, Zhenping Sun, Arda Kurt, and Qi Zhu

Abstract—**In this paper, a state space sampling-based local trajectory generation framework for autonomous vehicles driving along a reference path is proposed. The presented framework employs a two-step motion planning architecture. In the first step, a Support Vector Machine based approach is developed to refine the reference path through maximizing the lateral distance to boundaries of the constructed corridor while ensuring curvature-continuity. In the second step, a set of terminal states are sampled aligned with the refined reference path. Then, to satisfy system constraints, a model predictive path generation method is utilized to generate multiple path candidates, which connect the current vehicle state with the sampling terminal states. Simultaneously the velocity profiles are assigned to guarantee safe and comfort driving motions. Finally, an optimal trajectory is selected based on a specified objective function via a discrete optimization scheme. The simulation results demonstrate the planner's capability to generate dynamically-feasible trajectories in real time and enable the vehicle to drive safely and smoothly along a rough reference path while avoiding static obstacles.**

I. INTRODUCTION

Autonomous ground vehicle has become a rapidly growing research area nowadays, which has received considerable attention from both academic and industrial communities [1]. A series of autonomous ground vehicle competitions sponsored by DARPA have demonstrated its potential to improve driving safety and comfort [2], [3].

Typically, developing an autonomous vehicle requires various state-of-the-art technologies, such as perception, localization, planning and control. One of core techniques for autonomous driving is motion planning algorithm, which aims to generate a collision-free and drivable path or trajectory for the low-level path tracking controller to follow. In complex outdoor environments, to enable the autonomous vehicle to behave both deliberatively and reactively in real time with limited on-board computational resources, a hierarchical motion planning framework is often applied [4]. More specially, the high-level motion planner generally involves in computing a long-term collision-free geometric reference path, while the low-level motion planner refers to generating relatively short-term but dynamically-feasible trajectories subject to vehicle kinodynamic and environmental constraints. Most of the well-known high-level motion planners employ potential field methods or graph-search sampling-based planning algorithms, such as hybrid-A* that relies on the regularly discrete grid map [5], Anytime Dynamic A* search based on multi-resolution state lattices [6], and random sampling-based algorithm—rapidly-exploring random trees (RRT) [7]. Since these high-level path planning algorithms are often computationally demanding, so they may not be activated in a fixed cycling time to react to the dynamic environment in real time. They are often invoked when the global path is not available or collision with obstacles, or when complex motions are demanded, such as parking and U-turn. In addition, planning results from high-level motion planner are typically composed of a sequence of concatenated short-term motions based on low-fidelity vehicle model, which may not satisfy curvature-discontinuous or vehicle kinodynamic constraints. Therefore, neither the off-line global reference path nor the planned path from the high-level motion planner is guaranteed to be curvature-continuous and let along satisfy system constraints. So tracking the reference path directly could easily result in overshoots, oscillations or even crash, which is especially critical when the vehicle drives at high speeds or in cluttered environments. In order to refine the reference paths, we incorporate a path smooth algorithm based on support vector machine (SVM) approach. To further handle the environmental and system constraints and generate a dynamically-feasible trajectory in real time, a local trajectory planner is introduced. It plays a significant role in ensuring safety and reliability of the entire control system. Also, its existence fills the gap between collision-free path planning and low-level tracking control, as referred in [8]. In this paper, we are primarily focusing on the reference path optimization and local trajectory generation for autonomous driving along a rough reference path.

A. Related Work

There exists rich research work on local trajectory planning in the autonomous vehicle community. One of well-known approaches to local trajectory planning is sampling-based motion planning method. It could be roughly divided into two categories, which are discrete control space sampling-based method and state space sampling-based method [9]. In order to obtain the optimal trajectory, a discrete evaluation scheme based upon a specified evaluated function is followed in the post-processing step.

As for the control space sampling-based approach, a finite set of trajectories are generated by forward integration of differential equations, which describes vehicle kinematic and/or dynamic constraints. Therefore, the generated trajectories are inherently feasible and satisfying the vehicle kinematic and/or dynamic constraints. To simplify the expression of solution space, typically a parametric model of control input space is introduced, such as circular arcs in [10] [11], clothoids in [12] [13]. Then, the generated local motion can-

*The research is supported by NSFC Grants No. 61075043 and No. 61375050.

X. Li, Z. Sun, and Q. Zhu are with the College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073 China (e-mail: xiaohui_lee@nudt.edu.cn; sunzhenping@outlook.com; zhq_cs@126.com). Currently, X. Li is a visiting Ph.D. student in the Electrical and Computer Engineering Department, The Ohio State University, Columbus, OH, USA.

A. Kurt is with the Electrical and Computer Engineering Department, The Ohio State University, Columbus, OH, USA (e-mail: kurta@ece.osu.edu)

didates are evaluated by a cost evaluation function. Finally, an optimal trajectory is chosen to be fed to the low-level trajectory tracking controller. The control space sampling-based approach is easily to be applied. However, it does not explicitly take environmental constraints into account during the trajectory generation process, as a matter of face, most of the generated trajectories will be eventually trimmed, which may consume large amount of unnecessary computing resources.

Comparing to the control space sampling-based method, the state space sampling-based approach does not only consider kinodynamic constraints, but also accounts for the constraints imposed by environments. Aligning the terminal sates of generated paths with the reference path could prevent the vehicle from entering unsafe end pose and provide an efficient sampling scheme as well. Some authors establish a curvilinear coordinate based on a curvature-continuous reference path firstly. Then, a set of terminal states are determined by sampling laterally offsetting the reference path at a finite horizon [8], [14], [15], [16]. The heading angle of sampling terminal states are set to coincide with the reference path. To improve expressiveness of the trajectories, make full use of vehicle's maneuverability and maximize the probability of finding a collision-free and smooth path, different sampling strategies are proposed. In [14], multiple path candidates are generated by changing the lateral offset and its varying rates. While in [7] and [8], a closed-loop control law is employed to ensure the system-compliance nature of generated trajectory candidates. In [16], authors deal with the problem by decoupling tangential and normal motions by using quintic polynomials with respect to time. Then, a cost quadratic evaluate function is developed to obtain smooth trajectories by minimizing the jerk of motions. An referred in [17], the author formulates the trajectory generation problem into a two-point boundary value problem (BVP) subject to vehicle kinematic and dynamic constraints. Control inputs are parameterized by using a polynomial model. The numerical method is applied to solve the BVP problem, which also improve its generality. Even though due to the non-invertibility of the vehicle kinematic model, it is nontrivial to theoretically prove its convergence property, online applications have demonstrated its efficiency.

B. Motivation

Notice that the performance of most of existing local trajectory planners is strongly dependent on the quality of the reference path. In order to refine the reference path, a global path planner is developed to smooth the global reference path with bounds on the curvature and its first-derivative [18]. However, it is preprocessed without considering the dynamic environments. In addition, the planned path may achieve short path length at the cost of hugging obstacles, which is especially crucial when the vehicle navigates in cluttered environments. In order to mitigate these negative effects on the reference path, in [6], a complicated conjugate gradient optimization method is presented to smooth the prior global returning path. However, it is computationally demanding, in addition, the velocity profile planning is also not considered.

C. Contributions

In this paper, we primarily focus on the local motion planning for autonomous navigation. We assume that a reference path has been obtained via the high-level path planner. Based on the previous research work, a local trajectory generation and optimization framework is developed by considering both the environmental constraints and vehicle kinematical constraints, which is similar to [19] in spirit. The proposed two-level local motion planning architecture is depicted as follows:

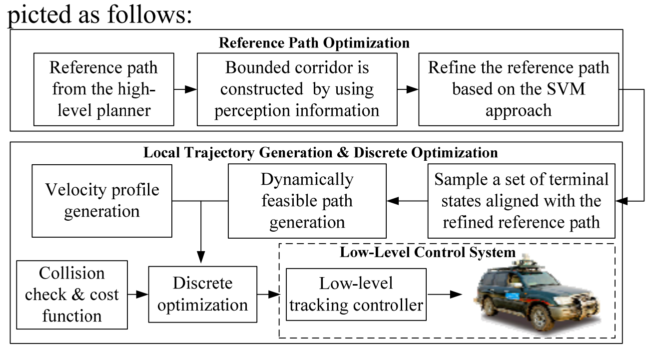

Figure 1. Reference path optimization based on the nonlinear SVM.

In the first step, the reference path is refined by using the SVM method, which maximizes the lateral distance to the road boundaries while guaranteeing the curvature-continuity. In the second step, model predictive path generation method is developed to generate dynamically-feasible trajectories aligned with the refined reference path, as in [17]. Simultaneously we generate velocity profiles subject to several constraints in order to achieve safe and comfort driving motions, as in [19]. Then, a discrete optimization scheme is applied to select the optimal trajectory, which is executed by the low-level control system.

II. Reference Path Optimization Based on SVM

The global reference path is typically represented in the form of waypoints. While, tracking the reference path directly may unduly restrict the vehicle motions. In order to avoid the vehicle's motions being strongly constrained by one-dimensional waypoints, we employ a ribbon-shaped road model, which is defined by the right and left boundaries [11].

A. Definition of the Reference Corridor

Through statistically analyzing the human driving behavior, it is not difficult to observe that human drivers typically steer the vehicle to control it within a safe corridor rather than obsessively track with the one-dimensional geometric reference path to minimize the lateral distance and heading errors. In light of this, we develop a local trajectory planner to generate several trajectory candidates aligned with the reference corridor. Instead of tracking with the reference path directly, the vehicle is controlled to run within a bounded corridor, which relaxes the constraints imposed by the reference path, but still subject to the environmental constraints. It makes the behaviors of the vehicle more like human drivers'. Besides, it will alleviate the negative effects that result from the localization errors and reference path discontinuities, still generating relatively smooth steering control inputs. In order to further improve the path tracking performance, the reference path along the corridor should be smooth and also deviate from the right and left boundaries as far as possible. Here we define the reference corridor used in the paper firstly. It is defined by the right and left boundaries. As shown in Fig. 1 (a), the binary image denotes the grid map of local environment. The black area is occupied by obstacles and the other area is collision-free area. The discrete green points denote the resulting global path obtained by using a graph search path planning algorithm. It can be seen that the resultant path approach obstacles at some points. So when the vehicle tracks the one-dimensional path directly, it may easily result in abrupt steering maneuvers or even inevitable collision. In order to mitigate these negative effects and improve the control performance, a ribbon-shaped corridor is generated, as shown in Fig. 1 (b). The blue points denote the right and left boundaries of the corridor, which are generated by using the collision check algorithm. The boundaries are generated by taking collision-free constraint and maximal corridor width constraint into account.

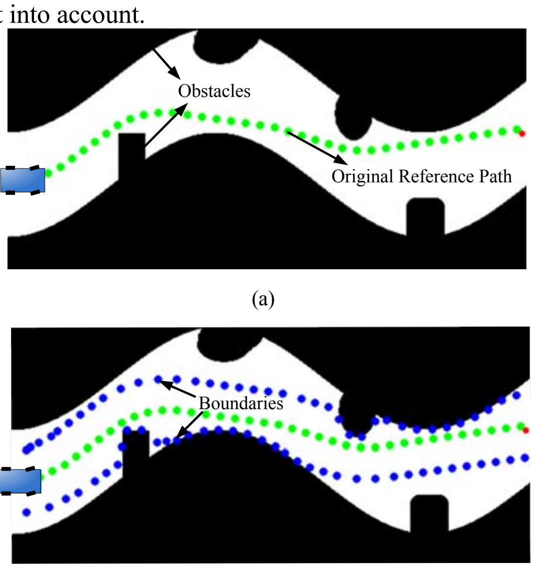

Figure 2. The generation of reference corridor. (a) The original reference path is obtained by using graph search path planning algorithm; (b) The reference path corridor is generated by using the collision test method.

In order to maximize the safe distance to the corridor boundaries and smooth the reference path along the ribbon-shaped corridor as well, we utilize SVM method as in [20] and [21]. The SVM is one of the powerful pattern classification techniques, which has been extensively and successfully applied in the field of pattern recognition and machine learning [22]. One of the significant properties for the SVM approach is that it has abilities to maximize the margin between the samples of different classes and ensure the smoothness and continuity of the separating surface. Therefore, it is suitable for robotic vehicle path planning and optimization.

B. The Linear Support Vector Machine

Let (x_i, y_i), $x_i \in \mathcal{R}^m$, $y_i=\{+1,-1\}(i=1,\ldots,N)$ denote the binary training samples. By using SVM method, the classification problem could be formulated into computing a separating hyper-plane $g(x) =< \tilde{w}, x > +\tilde{b}$ based on margin maximal principle. If the positive and negative training samples are linearly separable, then:

$$\begin{cases} < \tilde{w}, x_i > +\tilde{b} > 0, if\ y_i = +1 \\ < \tilde{w}, x_i > +\tilde{b} < 0, if\ y_i = -1 \end{cases} (i = 1,..., N) \quad (1)$$

So the above formula could be also written as:

$$y_i < \tilde{w}, x_i > +\tilde{b} > 0,\ (i = 1,..., N) \quad (2)$$

There exists ε, s.t. $y_i < \tilde{w}, x_i > +\tilde{b} \geq \varepsilon,\ (i = 1,..., N)$, i.e.:

$$y_i < w, x_i > +b \geq 1,\ (i = 1,..., N; w = \tilde{w} / \varepsilon, b = \tilde{b} / \varepsilon) \quad (3)$$

That means the positive and negative training samples could be separated by two hyper-planes:

$$\begin{cases} H_1 :< w, x_i > +b = 1 \\ H_2 :< w, x_i > +b = -1 \end{cases} (i = 1,..., N) \quad (4)$$

The distance between the two hyper-planes is $2/\|w\|$, and the problem is reformulated as a quadratic programming problem by minimizing following objective function.

$$\min \Phi(w,b) = \|w\|^2 / 2 \quad (5)$$

Lagrange function is employed to solve the convex optimization problem as follows:

$$L(\mathrm{w}, \mathrm{b}; \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i (< w, x_i > +b) - 1] \quad (6)$$

The αi (i=1,…,N) are the Lagrange multipliers. The origin optimization problem can be reformulated by solving the following dual problem:

$$\begin{aligned} &\max \mathrm{W}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j < x_i, x_j > \\ &s.t. \sum_{i=1}^{N} \alpha_i y_i = 0,\ \alpha_i \geq 0;\ \ w^* = \sum_{i=1}^{N} y_i \alpha_i^* x_i \end{aligned} \quad (7)$$

According to the Karush-Kuhn-Tucker (KKT) condition, the (8) should be satisfied to obtain the optimal solution.

$$\alpha_i [y_i < w^*, x_i > +b^* - 1] = 0 \quad (8)$$

So there exists only some special training data xi with non-zero αi. These special data are either of the hyper-planes, which is called support vectors and determine the parameters of the discrimination function (9).

$$\sum_{i \in S} \alpha^* y_i < x_i, x > +b^* = 0,\ b^* = \frac{1}{2}[w^* x^+ - w^* x^-] \quad (9)$$

Where x+ is a random support vector of positive samples and x- is a random support vector of negative samples. S denotes the set of indices of support vectors.

C. The Nonlinear Support Vector Machine

In order to solve the nonlinear separating problem, i.e. the training data could not be separated by a linear hyper-plane, some researchers introduce some kernel functions. It is able to transform the non-separating problem in low-dimensional space into a separating problem in the high-dimensional space.

Let x_i and x_j represent two samples in the original space. The implicit function $\phi(x)$ can map the samples from the original space to a higher dimensional space. Some classes of kernel functions employed to calculate the inner products of $\phi(x_i)$ and $\phi(x_j)$ in the high-dimensional space.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (10)$$

With the introduction of kernel functions, the objective function can be represented as:

$$\max W(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (11)$$

The discrimination function will also be changed as:

$$\sum_{i=S}\alpha_i^* y_i K(x_i, x_j) + b^* = 0 \quad (12)$$

Since all of the necessary calculations in the higher dimensional space are the inner product calculation based on kernel function in the original space, the higher dimensional space does not need to be represented explicitly.

D. Reference Path Optimization Based on Nonlinear SVM

A reference path optimization method is developed by using the nonlinear SVM. By using this method, the path optimization problem is formulated into computing a separating surface along the ribbon-shaped corridor.

First we classify the right and left boundaries as positive and negative training samples respectively. As shown in Fig. 2, the positive and negative training samples are obtained by discretely sampling along the local reference corridor. Let (x_{r1}, y_{r1}),…,(x_{rN}, y_{rN}), x_{ri}, $\in \mathcal{R}^2$, y_{ri}={+1}(i=1,…,N) be the right boundary (the blue line marked with the diamond), and (x_{l1}, y_{l1}),…, (x_{lN}, y_{lN}), x_{li}, $\in \mathcal{R}^2$, y_{li}={-1}(i=1,…,N) be the left boundary (the blue marked with the square). We apply the RBF kernel function to compute the separating hyper-plane, as the red curve (marked with the pentagram) shown in Fig. 2.

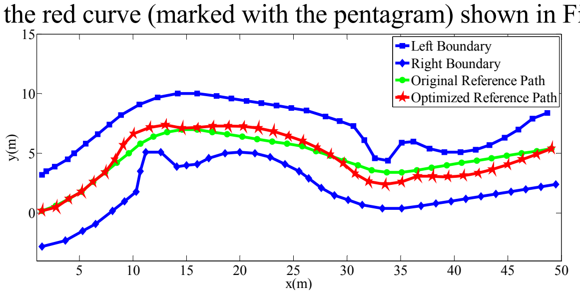

Figure 3. Reference path optimization based on the nonlinear SVM.

Comparing to the original reference path, the optimized path does not only maximize the safe distance to the boundaries, but also guarantees the curvature-continuity. However, the curvature of the separating surface cannot be guaranteed to be under a certain limit. Besides, it does not take the vehicle kinematic and dynamic constraints into account. So the resulting path might be too curvy to smoothly follow at some points. In order to alleviate the negative effects, we apply the local trajectory planner to further generate the feasible trajectory for car-like robots in the next section.

III. SAMPLING-BASED LOCAL TRAJECTORY PLANNER

The local trajectory generation layer generally aims to generate a set of collision-free and dynamically-feasible path candidates and assigned with the desired velocity profile. The local trajectory planner based on state space sampling-based planning scheme has been extensively studied and successfully implemented for both on-road and off-road autonomous driving. Different from the control space sampling-based strategy, it samples the terminal states firstly. Then the trajectory candidates are computed to connect the initial states with the terminal states

Since there is no structured information in the unstructured environment, typically the terminal sampling states are determined based upon the reference path. In order to maximize the probability to obtain a collision-free and feasible primitive, terminal states are uniformly sampled along the global reference path [14], [15] or bias sampled to adapt to the environment [9]. Then trajectory generation approaches are developed to generate multiple local trajectory alternatives by considering the vehicle kinematic and dynamic constraints in order to handle the dynamic environment or even the imminent situations. In this paper, we decouple the trajectory generation problem into the path generation and velocity profile generation problem, similar to [19].

A. Local Path Generation

We adopt the path generation method, as referred in [17], which reformulates the trajectory generation problem into a two-point boundary value problem (BVP). More specially, it generates a sequence of open-loop control commands u, which is capable of steering the vehicle from the initial state X_0 to the defined terminal state X_f, while satisfying the vehicle kinematic constraint $\dot{X} = f(X, u)$ and other constraints, such as maximal steering angle and steering rate. In essence, it is a general model predictive control method.

Instead of using the absolute world coordinate framework, we establish the curvilinear coordinate framework based on the refined reference path. So the states are represented in terms of arc-length s, i.e. ($x(s)$, $y(s)$, $\theta(s)$). Here we generate the local trajectories that connect the initial states ($x(s_0)$, $y(s_0)$, $\theta(s_0)$) with terminal states ($x(s_f)$, $y(s_f)$, $\theta(s_f)$) with satisfying the vehicle kinematic constraints. Where (x, y) is the position and θ is the heading state. Since the paper focuses on the local trajectory planner for car-like robots, the curvature is applied as the steering control inputs and it is also represented in terms of arc-length s, i.e. κ(s). Typically, the control inputs are parameterized by using the polynomial curvature spirals, which are effective to represent the most possible feasible motions with a minimum number of parameters [17]. Hence, integrated form of the vehicle kinematic model is described as follows.

$$\begin{aligned} x(s) &= \int_{s_0}^{s_f} \cos(\theta(s))ds \\ y(s) &= \int_{s_0}^{s_f} \sin(\theta(s))ds \\ \theta(s) &= \int_{s_0}^{s_f} \kappa(s)ds \\ \kappa(s) &= \kappa_0 + \kappa_1 s + \kappa_2 s^2 + \kappa_3 s^3 \end{aligned} \quad (13)$$

As shown in figure 3, we determine the terminal states by discretizing lateral offset $d(s)$ to reference path $r(s)$ and longitudinal curvilinear abscissa value s along the reference path $r(s)$. Firstly, the curvilinear abscissa value s_i, i.e. look-ahead distance, is also discretized by step Δs, with upper-bound s_{max} and lower-bound s_{min}, which are determined by the current velocity. Also, the lateral offset $d(s)$ is discretized by step Δd, with upper-bound d_{max} and lower-bound d_{min}, which are determined by the width of the reference corridor at different positions. Then the heading state $\theta(s_f)$ of the sampling point ($x(s_f, d_f)$, $y(s_f, d_f)$) is the same as the heading of the point on the reference path with curvilinear abscissa value s_f.

More precisely, a set of path candidates are computed to connect the initial states ($x(s_0)$, $y(s_0)$, $\theta(s_0)$) and terminal states ($x(s_f)$, $y(s_f)$, $\theta(s_f)$) with considering the differential constraints governed by (13). Since the closed-form solution of the inverse kinematics problem could not be solved analytically, a gradient shooting numerical optimization method, as referred in [17], is utilized to determine the unknown parameters [κ_0, κ_1, κ_2, s_f] of the control inputs. The states are propagated by forward integral of the differential equation, which describes the vehicle kinematics and the other constraints.

As illustrated in Fig. 3, a set of trajectories are generated to connect the initial state with multiple discrete sampling terminal states. Although no theoretical work is obtained to guarantee the convergence and global optimal properties, in most cases, the results are near the optimal solution from the experimental results. Furthermore, since the computation of path candidates are completely parallelizable, the quality of the optimal path could be potentially improved by using the parallel computation technique to increase the sampling density along the longitudinal and lateral directions as well.

B. Velocity Profile Generation

Following the path generation process, velocity control laws are assigned along the planned geometric path. In order to enhance the vehicle speed and avoid uncomfortable accelerated/decelerated longitudinal motions and mitigate the lateral sideslip effects, several constraints are taken into account as follows.

- Maximal allowed speed V_{limit1}:

$$v(s_i) \le V_{\text{limit1}}(s_i) \tag{14}$$

In general, it is issued from the high-level planning system by reasoning about the road conditions.

- Maximal allowed lateral acceleration $Acc_{lateral}$

$$v(s_i) \le \sqrt{\frac{Acc_{lateral}}{|\kappa(s_i)|}} \tag{15}$$

According to the differential model (13), the curvature of the returning paths at each specified position along the path could be computed explicitly.

- Maximal allowed longitudinal acceleration Acc_{lon} and deceleration Dec_{lon} :

$$v_{\min} \le v(s_i) \le \sqrt{v^2(s_0) + 2Acc_{lon}s_i}$$
$$v_{\min} = \begin{cases} \sqrt{v^2(s_0) + 2Dec_{lon}s_i} & if\ v^2(s_0) + 2Dec_{lon}s_i > 0 \\ 0 & otherwise \end{cases} \tag{16}$$

- Maximal brake deceleration Dec_{max} constraints

$$v(s_i) \le \sqrt{v^2_{\text{terminal}} + 2Dec_{\max}(s_N - s_i)} \tag{17}$$

In order to guarantee the safety motion of the vehicle, it should always have enough time to stop or reach a specified speed at the terminal position of each planned path by braking at the maximal deceleration. The s_N denotes the arc-length of the path candidate, v_{terminal} is the speed assigned to the terminal position according to the road conditions.

C. Trajectory Evaluation

As discussed in [19], it is difficult to develop absolute criteria for the trajectory evaluation. Typically the objective function is constructed by considering factors such as safety, comfort and efficiency. First, the collision check is conducted to trim the trajectories in collision with obstacles. Then remaining trajectories are evaluated according to a specified cost function J, which consists three terms. The first term J_d is designed to penalize the trajectories, which are closed to the boundaries, where d_{piL} and d_{piR} denote the nearest distance between a certain point p_i on the trajectory to the left boundary and right boundary respectively, and D is the road width; the second J_κ term penalizes the curvy trajectory; the third term J_l prefers the trajectory which can be driven for longer time. The three terms are combined in a linearly weighted sum.

$$\begin{aligned} J &= w_1 J_d + w_2 J_\kappa + w_3 J_l \\ J_d &= \frac{1}{N}\sum_{i=1}^{N}(1-\min\{\frac{d_{PiL}}{D/2},\frac{d_{PiR}}{D/2}\}) \\ J_\kappa &= \frac{1}{N\kappa_{\max}}\sum_{i=1}^{N}|\kappa(s_i)|,\ J_l = (1-\frac{s_N}{l_{\max}}) \end{aligned} \tag{18}$$

The parameters d_{max}, $\kappa_{\max}$, $l_{\max}$ are the maximal lateral distance, maximal curvature and maximal predictive distance respectively, which are used to normalize each of the cost terms within the range [0, 1]. The w_1, w_2 and w_3 are the weights for different terms, w_1+w_2+w_3=1. These weights can be adjusted to generate human-acceptance maneuvers and handle various situations according to the road conditions. As shown in Fig. 3, the path candidates of purple color are in collision with the obstacles. The candidates with lighter color imply larger cost. The green candidate is the optimal one according to the specified cost function.

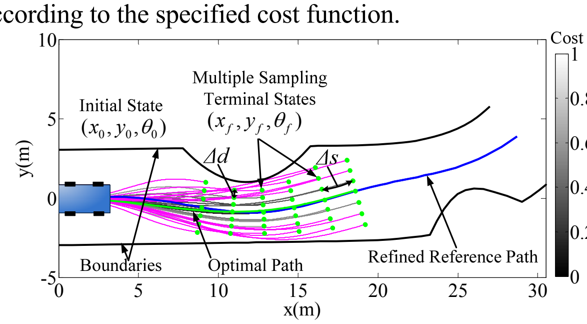

Figure 4. Sampling-based local path candinates generation and evaluation.

IV. EXPERIMENTAL RESULTS

In this section, we show the simulation experimental results to demonstrate the performance of the proposed local trajectory generation algorithms.

As shown in Fig. 4(a), a curvy corridor with multiple obstacles is constructed in the simulation environment with the static obstacles. The red path is refined based on SVM. It is generated by maximizing the lateral distance to both the right and left boundaries. Then local trajectory planner is employed to generate a set of dynamically-feasible path candidates and assigning with speed laws. Fig. 4(b) illustrates the local trajectory candidates generation and evaluation results when the vehicle reaches the position (x=15.12m, y=-1.07m), as indicated by the dashed rectangle in Fig. 4(a). It can be seen in Fig. 4(b), instead of applying the uniform sampling method, an environmental adaptive sampling scheme is employed to adjust the sampling states according to the varying width of the corridor. Parameters related to the trajectory generation algorithm are set as follows: look-ahead distance $l_p \in$[8m, 18m], Δl_p=2.5m; V_{limit1}=30km/h; $Acc_{lateral}$=3m/s^2; Acc_{lon}=3.5m/s^2; Dec_{lon} =-3.5m/s^2; Dec_{max} =-5m/s^2;$v_{terminal}$=0m/s; w_1=0.8, w_2 =0.1, w_3=0.1. Since all of the path generation candidates could be computed in parallel, the sampling horizon and density could be improved by using the parallel computation hardware. Then, an open-loop trajectory tracking strategy based on feedforward control commands is applied to track the optimal trajectory. More precisely, at the end of each planning cycle, steering and speed control commands are deduced from the optimal trajectory. The final tracking trajectory of an Ackerman-steered vehicle is shown in Fig. 4(a), and the velocity and front wheel steering angle of the vehicle along the path are displayed in Fig. 4(c) respectively.

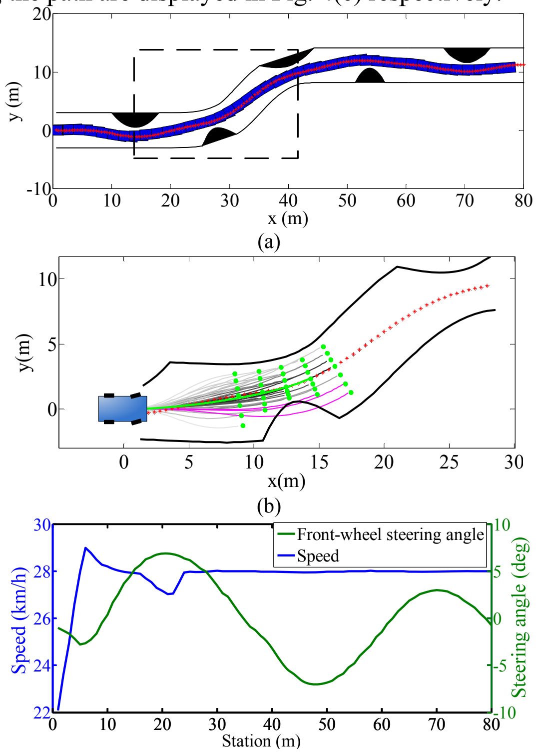

Figure 5. Experimal results for local trajectory tracking.

V. Conclusions and Future Work

This paper has presented a state space sampling-based local trajectory planning approach for autonomous vehicles driving along a reference path. A SVM-based method is proposed to refine the rough reference path by maximizing the lateral distance to the boundaries of the reference corridor while ensuring the curvature continuity. In order to reduce the computational complexity and meet environmental constraints, terminal states are sampled aligned with the refined path. A model predictive motion planner is employed based on gradient shooting optimization technique to generate dynamically-feasible trajectory candidates. Finally, the generated trajectories are evaluated by a cost function.

In the future, we will focus on implementing the proposed research on the full-size autonomous vehicle platform. Furthermore, we will handle the local trajectory generation problems by considering more practical scenarios, such as the presence of other traffic participants, or at the limits of its handling capabilities, such as driving at a high speed. In these cases, accurate motion prediction is required to ensure safety, stability and comfort.

References

[1] Ü. Özgüner, T. Acarman, and K. Redmill, *Autonomous Ground Vehicles*. Norwood, MA: Artech House, 2011.

[2] K. Iagnemma and M. Buehler, Eds., "Special issue on the DARPA grand challenge," *J. Field Robotics*, vol. 23, no. 8, pp. 461–652, 2006.

[3] K. Iagnemma and M. Buehler, Eds., "Special issue on the 2007 DARPA urban challenge," *J. Field Robotics*, vol. 25, no. 8-9, pp. 423–860, 2008.

[4] R. V. Cowlagi and P. Tsiostras, "Hierarchical motion planning with dynamical feasibility guarantees for motion robotic vehicles," *IEEE Trans. on Robotics*, vol. 28, no. 2, pp. 379–395, 2012.

[5] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, 2010.

[6] D. Dolgov, et al., "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 28, no. 8, pp. 933–945, 2009.

[7] Y. Kuwata, et al., "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. on Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, 2009.

[8] U. Schwesinger, et al., "A sampling-based partial motion planning framework for system-compliant navigation along a reference path," *IEEE Intell. Veh. Symp.*, pp. 391–396, 2013.

[9] T. Howard, et al., "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *J. Field Robotics*, vol. 25, no. 6-7, pp. 325–345, 2008.

[10] F. V. Hundelshausen, et al., "Driving with tentacles: integral structures for sensing and motion," *J. Field Robotics*, vol. 25, no. 9, pp. 640–673, 2008.

[11] Z. Sun, et al., "Ribbon model based path tracking method for autonomous land vehicle," *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 1220–1226, 2012.

[12] A. Broggi, et al., "Autonomous vehicles control in the VisLab intercontinental autonomous challenge," *Annu. Rev. Control*, vol. 36, no. 1, pp. 161–171, 2012.

[13] A. Lacaze, et al., "Path planning for autonomous vehicles driving over rough terrain," in *Proc. the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, MD 1998, pp. 50–55.

[14] S. Thrun, et al., "Stanley: The robot that won the DARPA Grand Challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.

[15] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1599–1616, 2012.

[16] M. Werling, et al., "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, 2011.

[17] T. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Int. J. Robot. Res.*, vol. 26, no. 2, pp. 141–166, 2007.

[18] J. Villagra, "Smooth path and speed planning for an automated public transport vehicle," *Robot. Auton. Syst.*, vol. 60, no. 2, pp. 252–265, 2012.

[19] T. Gu, et al., "Focused trajectory planning for autonomous on-road driving," *IEEE Intell. Veh. Symp.*, pp. 547–552, 2013.

[20] J. Miura, "Support vector path planning," *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, pp. 2894–2899, 2006.

[21] Q. Chen, et al., "Local path planning for an unmanned ground vehicle based on SVM," *Int. J. Robot. Syst.*, vol. 9, no. 246, pp. 1–13, 2012.

[22] V. N. Vapnik, *Statistical Learning Theory*. New York, NY: John Wiley & Sons, 1998.