2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Combining Local Trajectory Planning and Tracking Control for Autonomous Ground Vehicles Navigating along a Reference Path

Xiaohui Li, Zhenping Sun, Daxue Liu, Qi Zhu and Zhenhua Huang

Abstract—**In this paper, we develop an integrated local trajectory planning and control scheme for the navigation of autonomous ground vehicles (AGVs) along a reference path with avoidance of static obstacles. Instead of applying traditional cross track-based feedback controllers to steer the vehicle to track the reference path as closely as possible, we decompose the path following task into two subtasks. Firstly, in order to follow the reference path with smooth motions and avoid obstacles as well, we apply an efficient model-based predictive trajectory planner, which considers geometric information of the desired path, kinematic constraints and partial-dynamic constraints to obtain a collision-free, and dynamically-feasible trajectory in each planning cycle. Then, the generated trajectory is fed to the low-level trajectory tracking controller. Relying on the steady-state steering characteristics of vehicles, we develop an internal model controller to track the desired trajectory, while rejecting the negative effects resulting from model uncertainties and external disturbances. Simulation results demonstrate capabilities of the proposed algorithm to smoothly follow a reference path while avoiding static obstacles at a high speed.**

I. Introduction

The past three decades have witnessed the rapid development in the research field of autonomous driving, which attracts considerable research interest and efforts from both academia and industry. During the famous AGV competitions, the DARPA Grand Challenges and the DARPA Urban Challenge, autonomous ground vehicles had demonstrated their great potentials to improve driving safety, efficiency and comfort in both off-road and on-road environments [1]-[3]. These competitions showed significant advances over the-state-of-the-art in autonomous driving technology and stimulated extensive interest in the AGV research field as well. Recently, partially automated control functions have already been applied in the driver assistant systems and several automotive companies even make their research and development plans for producing future autonomous cars. Nevertheless, there is still a number of challenges to be faced for developing truly reliable and robust fully-autonomous driving vehicles to handle various realistic situations in the real world.

As is well-known that the development and application of AGVs require the integration of the-state-of-the-art technologies, ranging from perception, localization to navigation and control [4]. As the core modules, both local motion planner and path tracking controller play a significant role in guaranteeing safety and improving driving comfort.

In order to track the reference path accurately and robustly, many researchers developed Lyapunov-based feedback control laws by considering vehicle kinematics and dynamics, such as sliding model control, backstepping control and so on. To adapt high speed and varying terrain conditions and improve control accuracy and robustness, some researchers explore cascaded or multi-tiered control strategies and focusing on minimizing lateral errors in the outer loop, and stabilizing yaw motions via steering actions in the inner loop [5]. To reject model uncertainties and external disturbances, tire sideslip angles and inertial effects are taken into account. However, most of these controllers formulate the tracking control problem into a regulation problem, which primarily concerns with current cross-track error information (such as lateral and heading deviations from the reference path) to generate the immediate control action instead of a sequence of optimal or sub-optimal control actions within a finite horizon. In addition, state and control constraints are often ignored. Therefore, it may easily result in abrupt steering actions when vehicles deviate far from the reference path or negotiate a tight turn. Some researchers utilizes the gain-scheduling or variable structure control approaches to avoid abrupt control actions to achieve graceful motions at the expense of reducing tracking accuracy or error convergent speed [6]. Based on the comparison of a variety of path tracking controllers, [7] concluded that path tracking control performance is strongly dependent on both vehicle dynamics and smoothness of the reference path. Essentially, most of these kinematics-based and/or dynamics-based controller primarily focused on eliminating the current cross-track errors instead of taking advantage of the predictive information ahead to integrally optimize a sequence of control actions and its corresponding trajectory, which smoothly regulates the vehicle from the current state onto the sampling states aligned with the reference path ahead.

There exists a large amount of research on integrated planning and control approaches for AGVs by applying optimization techniques. One of the most attractive methods is the Model Predictive Control (MPC), which is capable of formulating the vehicle navigation problem into a finite-horizon constrained optimization control problem [8]-[10]. MPC approach uses the vehicle kinematic or dynamic model to predict its future state evolution based on the current measured states. In each control cycle it generates a sequence of control actions, which minimizes a specific objective function within a finite horizon and satisfies the control constraints as well. Then, the first control action is issued to be executed by the low-level actuator. The process is repeated at subsequent time steps. The MPC scheme has the capabilities to systematically deal with system state and control constraints. However, it often assumes the reference path is known and the speed is constant over a short-term finite horizon ahead at each

* The research is supported by NSFC Grants No. 61075072 and No. 91220301.

X. Li, Z. Sun, D. Liu, Q. Zhu and Z. Huang are with the College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073 China (e-mail: xiaohui_lee@outlook.com; sunzhenping@outlook.com; 13508495034@139.com; zhq_cs@126.com; huangzhenhua_001@163.com).

X. Li is a visiting Ph.D. student in the Electrical and Computer Engineering Department, The Ohio State University, Columbus, OH, USA.

978-1-4799-6078-1/14/$31.00 ©2014 IEEE

time. When vehicles drive either in on-road or off-road dynamic environments, due to the existence of unexpected obstacles and localization errors, the collision-free assumption of the reference path may be impractical. Solving the optimization problem via MPC scheme may involve non-convex constraints when obstacles are considered. In this case, computational burden and limited on-board computational resources may become a barrier preventing MPC approaches from generating a collision-free and feasible trajectory in real time.

To deal with obstacle avoidance and achieve safe and graceful motions along the reference route, we introduce a computationally efficient local trajectory planner between the reference path and the path following controller based on a hierarchical framework. To generate a sufficiently long, collision-free and curvature-continuous trajectory in real-time, the planner is required to consider the reference route and surrounding environmental information from on-board perceptual system, as well as system model and constraints. To some extent, the local trajectory planner performs the function of bridging the gap between high-level rough reference path planning and low-level tracking control. In addition, it enables the vehicle to handle dynamic environments deliberatively and reactively. In addition, corresponding control inputs of the planned trajectory could be used as feedforward control commands by the low-level tracking controller and allow the feedback controller to focus on handling model uncertainties and disturbances.

To solve the local trajectory generation problem, several well-known sampling-based motion planning approaches have been extensively studied. Most of them follow a discrete optimization scheme. More specifically, a set of trajectory candidates is generated via forward simulation based on the system model. Then a best trajectory is selected according to an objective function. This sampling-based motion planning scheme can be roughly classified into two categories, one is control space sampling-based method and the other is state space sampling-based method [11].

The former scheme discretizes the control input space (such as constant-curvature arcs [12], clothoids [13], or concatenation of these short-term motions [14]) to generate a set of trajectory candidates via numerical forward integration of differential equations which governs vehicle kinematics or dynamics [15]. Therefore, the generated trajectory candidates are inherently drivable. Due to its simplicity and computational efficiency, the scheme has been widely applied for the local navigation purpose, especially suitable for finding a collision-free path in the far less constrained environment. Based on the *symmetric* nature of mechanical system, some researchers generate a motion primitive library offline in the body-centered coordinate and use them online via ration and translation [16]. Nonetheless, since the motion primitives are generated by discretizing the control input space, the motion primitives are often not well-separated. A large amount of computational resource will be consumed on exhaustive collision-test and evaluation process.

By contrast, instead of sampling discrete control inputs, the state space sampling-based motion planning scheme sample a set of terminal states by using the information of guidance path and environments. It not only considers the position constraints, but also accounts for heading and curvature states constraints imposed by the reference path. To be aligned with the reference path and obtain a collision-free and relatively smooth trajectory, a set of terminal states are sampled laterally offset along the reference path. Several approaches have been proposed to generate trajectories, which connect the vehicle current state with the terminal states aligned with the reference path. Based on cubic Bézier curves, [17] developed an efficient and analytical path smoothing algorithm to generate continuous-curvature path, which considers an upper-bound curvature constraint. Regarding the reference path as a baseline, [18] introduced a geometric method to generate multiple path candidates. The baseline is required to be smooth enough to ensure the smoothness of generated candidates. Instead of using geometric methods, [19] and [20] proposed local dynamically-feasible trajectory planners involving both close-loop control laws (kinematic-based nonlinear control law and pure pursuit control law, respectively) and system, state and control constraints, regardless of the smoothness of the baseline. [21] presented a model predictive trajectory generation scheme, which transforms the local trajectory generation problem into a two-point boundary value problem (BVP), subject to high-fidelity vehicle dynamic constraints. Due to use of the numerical solving method, it achieves a high degree of efficiency and generality. The low-level controller applies an open-loop control strategy for tracking control.

The remainder of the paper is organized as follows: in Section II we introduce the structure of the proposed framework. Then, Trajectory generation method and low-level path tracking controller are described in detail in Section III and Section IV respectively. Section V presents the simulation results. Followed by concluding remarks and future work in Section VI.

II. FRAMEWORK

The typical system architecture for an autonomous ground vehicle is illustrated in Fig.1. The perception and localization system obtained from the on-board sensors provide the environmental information surrounding the vehicle along with the vehicle position and pose information. The high-level mission planning is decided by the allocated task. While the behavioral planners reasons about road conditions, traffic regulations and other rules to issue safe and feasible behaviors to the motion planners. While the motion planning level often consists of reference path planning and trajectory planning, which primarily focus on generating a collision-free path or trajectory in accordance with the high-level intentions. The low-level tracking controller refers to tracking the generated trajectory as accurately as possible.

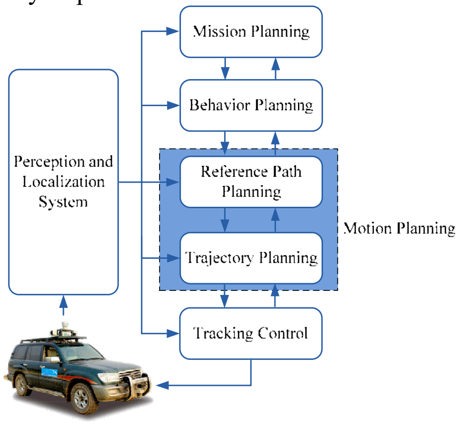


Figure 1. A typical hirarchical framework for AGVs.

We assume that a rough reference route is known as prior information. The assumption is practical because the global route could be obtained offline (for instance, global waypoints) or from online high-level path planner, such as graph-search algorithm, random sampling-based motion planner, *et al*; or extracted from the structured environment such as road lanes. Meanwhile, we do not require the reference path to be curvature-continuous and collision-free. In this paper, we mainly focus on local trajectory planning and tracking control problem. To ensure path curvature-continuity, avoid static obstacles as well as achieve graceful motions, we combine local trajectory planning and tracking control rather than following the reference path by only using a feedback controller.

Inspired by previous navigation framework in [18] and [22], which separated the navigation task into trajectory generation and subsequent tracking control via feedback control, we divide the navigation task into outer-loop trajectory generation planner and inner-loop trajectory tracking controller using a two-freedom control framework in a receding horizon manner. More specifically, the trajectory generation planner issues a dynamically-feasible trajectory and corresponding nominal control inputs in each planning cycle, while the inner-loop controller tracks the generated trajectory based on the corresponding nominal control inputs in each control cycle, which executes at a higher frequency. In order to deal with the model uncertainties and external disturbance, we introduce an inner model control strategy combining a feedforward controller with the feedback controller.

III. Trajectory Planning

For the local trajectory generation task, we develop an efficient state space sampling-based trajectory planning algorithm following the idea presented in [21] via a discrete optimization scheme. In each planning cycle, a set of trajectory candidates are generated aligned with the reference path. Based on an objective function, The best trajectory is selected and issued for low-level tracking

A. Terminal States Sampling based on a Reference Path

Through observing on-road human drivers' behaviors, it can be found that human drivers often steer the vehicle to be aligned with road lanes rather than minimizing time and energy [22]. In light of this, instead of using the absolute inertial coordinate framework, we employ the curvilinear coordinate framework relying on the reference path to express the reference path. To satisfy the constraints imposed by road geometry, we sample terminal states aligned with it. In addition, the bias-sampling scheme significantly reduces the computational complexity as well as prevents the vehicle from entering into dangerous states.

To avoid abrupt steering actions and ensure curvature-continuity of the generated trajectory, we use four dimensional state-space representation, i.e. position (x, y), heading θ and curvature κ. In order to obtain a collision-free and smooth trajectory, the terminal states are required to be sampled in a high-resolution state space. However, due to limited onboard computational resources, we have to restrict sampling density and range. So we adopt an efficient low-dispersion sampling strategy as shown in Fig. 2. In the longitudinal direction, to guarantee the minimal crash distance, and overcome the delay resulting from the actuators and vehicle inertial effects to ensure the stability of control, a minimal look-ahead distance is often set as a function of the vehicle's current speed. We set lateral offset bounds as well. The sampling density and range can be adjusted according to the environment (on-road or off-road) and computational resources available.

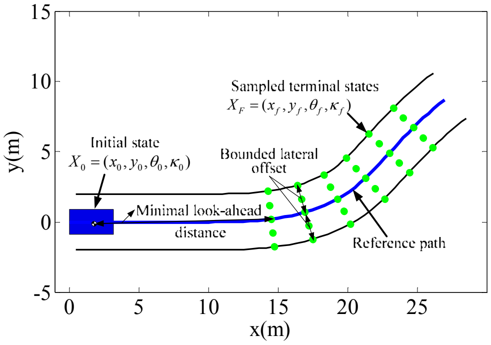


Figure 2. Terminal states are uniformly sampled along the reference path.

As shown in Fig. 2, a set of terminal states X_F (x_f, y_f, θ_f, κ_f) are uniformly sampled in both the lateral and longitudinal direction along the reference path. To be aligned with the reference path, the sampled heading state is set to be the same as that of the corresponding nearest point on the path. For simplicity, the trajectory is represented in the vehicle local coordinate framework, so vehicle current states are transformed to be (0, 0, 0, κ_0) via rotation and translation. Since we consider the initial curvature in each planning cycle, the smoothness of steering actions can be improved.

B. Model Predictive Path Generation

High-fidelity dynamic model can be applied to precisely predict evolution states of the vehicle. However, it refers to various time-varying parameters, online identification methods are required to precisely estimate these parameters in real time. Even so, the parameter variations caused by the interaction between the tire and ground are difficult to predict. Instead, we utilize the vehicle kinematic model to predict future state evolution of the system. Dynamic effects, such as sideslip angles and actuator dynamics, could be accounted for to improve the accuracy of the prediction.

$$\dot{x}(t) = v\cos(t),\ \dot{y}(t) = v\sin(t),\ \dot{\theta}(t) = v\kappa,\ \kappa = u(t) \quad (1)$$

It can be seen that a trajectory can be explicitly defined as a vector valued function of time t. However, a trajectory tightly couple the spatial path and velocity. It imposes strict velocity constraints on the path. In fact, we can track a geometric path without specifying velocity laws. Through integrating the time, the states can be represented as a function of the arc-length s instead of time t.

$$\begin{aligned} &x(\mathrm{s}) = \int_0^{s_f} \cos(\theta(\mathrm{s}))ds,\ \ y(\mathrm{s}) = \int_0^{s_f} \sin(\theta(\mathrm{s}))ds \\ &\theta(\mathrm{s}) = \int_0^{s_f} \kappa(\mathrm{s})ds,\ \ \kappa(s) = u(s) \end{aligned} \quad (2)$$

In this way, the trajectory planning could be decomposed into the geometric path planning and longitudinal velocity planning. The time-dependent vehicle model is transformed into a spatial-dependent model, which allows velocity to remain unspecified. Hence, we decompose the trajectory generation task into two sequential subtasks: geometric path generation and velocity planning. Solving the path generation problem involves computing a geometric path, which satisfies the current and sampled terminal states boundary constraints, the differential constraints represented by (2), and other state and control input constraints. Because of the nonlinear differential equations, it is nontrivial to solve the nonlinear constrained problem by using nonlinear programming methods in the continuous control space. In order to make the nontrivial inverse problem solvable, we follow the idea in [21], and parameterize the control input space and formulate the trajectory generation problem into a two-point boundary value problem (BVP). More precisely, we parameterize the control input space by using a cubic polynomial spirals model. Although it reduces the control input space, while it retains the property to express complicate maneuvers and limits the unknown parameters as well. Hence, the generated paths will be cubic polynomial spirals.

$$\kappa(s)=\kappa_0+\kappa_1 s+\kappa_2 s^2+\kappa_3 s^3 \qquad (3)$$

In this way, the BVP is transformed into solving the unknown control parameters $P=[\kappa_1, \kappa_2, \kappa_3, s_f]^{\mathrm{T}}$.

We apply a computationally efficient numerical nonlinear optimization technique, Newton's method described in [21], to solve the above BVP. The control parameters matrix P is iteratively solved via using the Newton's method, as represented in (4). For each BVP, the iterative procedure ends till the terminal states errors $\Delta X_{\mathrm{F}}(P)$ is less than a user-specified threshold. The initial guess of parameter P affects the speed of convergence. In order to obtain a good initial guess for online application and reduce the number of iterations, we use a l pre-computed ookup-table of initial parameters guess by sampling densely in the high-resolution state space.

$$\begin{aligned}
&\Delta X(P(k))=X_{FS}-X_F(P(k))\\
&\left.\frac{\partial \Delta X(P)}{\partial P}\right|_{P(k)} \Delta P=(\Delta X(P(k))) \qquad (4)\\
&P(k+1)=P(k)+\Delta P
\end{aligned}$$

Here, we use small perturbations to calculate the matrix of the first-order partial derivatives of a vector-valued function.

During the trajectory generation stage, to alleviate tire sideslip effects and reduce the control efforts for yaw motion stabilizing control, we consider lateral acceleration limits (as described in (5)) based on the current velocity and road conditions (such as road friction coefficient). In this way, we efficiently limit the tire slip angles and prevents the force of vehicle tires from entering into nonlinear saturated zone. Moreover, it can significantly improve vehicle stability and generate safer and more comfortable trajectories, though at the expense of reducing solution space.

$$|u(\mathrm{s})| \le \kappa_{\max},\ \kappa_{\max}=f(v,\mu) \qquad (5)$$

Since the numerical integration method is applied for the forward simulation of the system, control constraints can be easily handled.

Since the vehicle kinematic model is explicitly considered, all of the generated path candidates are kinematically-feasible. Even though, we acknowledge that the model predictive trajectory planner has some limits. Due to finite sampling terminal states and limited expressiveness of a single cubic spiral, it cannot always guarantee to obtain a collision-free and feasible solution. [23] proposed a more complex trajectory generation approach based on the spatiotemporal lattice to deal with more challenging scenarios at the expense of computational efficiency. Once none of feasible and collision-free paths could be generated by the mentioned approach, the emergent control-space sampling method or powerful graph-search path planning method will be evoked.

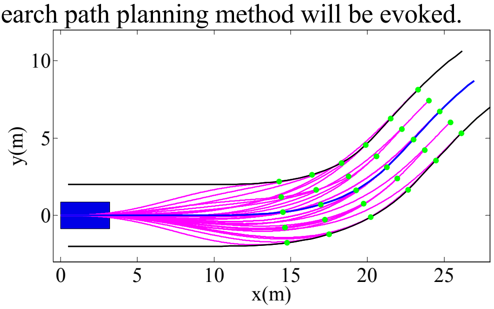


Figure 3. Path generation results.

C. Collision Check and Evaluation

To choose the best path among path candidates, we design a specified objective function for optimization criterion.

Firstly, the collision check performs to trim the paths in collision with obstacles. Since the vehicle shape is often rectangle, it could not be simplified as a mass-point. We refer to an efficient method proposed in [24]. As shown in Fig. 4, several circles are used to approximately represent envelop of the vehicle shape. To ensure collision avoidance, the distance between the obstacles and any center of these circles are required to be larger than the radius.

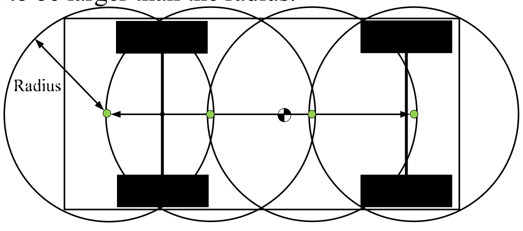


Figure 4. Circle decompostion of vehicle shape.

Then, the remaining collision-free path candidates are evaluated via a user-specified objective function. Here, we design an objective function consisting of five weighted cost terms, which are the obstacle proximity J_{o}, deviation from the reference path J_{d}, smoothness J_{s}, preview distance J_{p}, and subsequent consistency J_{c}. Each cost term contains a weighted factor, which are ω_{o}, ω_{d}, ω_{s}, ω_{p}, ω_{c}, respectively.

More precisely, the cost term J_{o} denotes the proximity to static obstacles. For instance, in off-road environments, an occupancy grid cost-map C_{o} can be pre-computed at the beginning of each planning cycle relying on the perception information. Each cell of the grid-map is assigned a cost value. Hence, J_{o} could be computed via accumulating the cost of the cells covered by the vehicle body, as described in (6), where τ_i (i=1,...,N) denotes a generated path. The cost term J_{d} describes the deviation from the reference path. $D(\tau_i(s))$ denotes the deviation distance of the generated path from the reference path. In order to improve the smoothness of the generated path, we define the smoothness criterion as J_{s}, which is obtained by integrating the path curvature. Besides, we define the cost term J_{p}, which reflects the preference of longer feasible paths. As represented in (6), $L_{\max}$ is the maximal look-ahead distance and l is the arc-length along the reference path. During the replanning process, the discrepancy of consecutive plans can easily result in overshoots, oscillations, or even instabilities of vehicle movement. To minimize the inconsistency, the discrepancy between the current evaluated path and the previous planning path is taken into account. The cost term J_c is computed by integrating the Euclidean distance $d(\tau_i(s))$ between them along the reference path. Considering all of these cost terms mentioned above, an optimization criterion could be defined as follows:

$$\arg\min_{i=1}^{N}\{\omega_o J_o(\tau_i) + \omega_d J_d(\tau_i) + \omega_s J_s(\tau_i) + \omega_p J_p(\tau_i) + \omega_c J_c(\tau_i)\}$$

$$= \{\frac{\omega_o}{s_f}\int_0^{s_f} C_o(\tau_i(s))ds + \frac{\omega_d}{s_f}\int_0^{s_f} \frac{|D(\tau_i(s))|}{D_{\max}}ds + \tag{6}$$

$$\frac{\omega_s}{s_f}\int_0^{s_f}\left|\frac{\kappa(\tau_i(\mathrm{s}))}{\kappa_{\max}}\right|ds + \omega_p\frac{L_{\max} - l(\tau_i)}{L_{\max}} + \frac{\omega_c}{s_f}\int_0^{s_f}\frac{d(\tau_i(s))}{d_{\max}}ds\}$$

In practice, the weighted factors can be flexibly tuned according to driving conditions. As illustrated in Fig. 5, the optimal path (green curve) is selected from the path candidates (red curves) and tracked by the low-level controller.

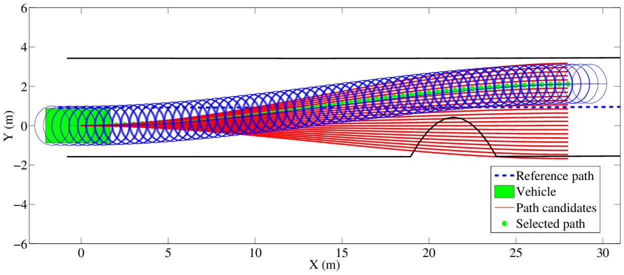


Figure 5. The evaluation of path candidates.

D. Velocity Profile Planning

The path states (x, y, θ, κ) are represented as a function of curvilinear abscissa s, which is the integration of velocity with respect to time t. In order to improve trajectory tracking control performance, the velocity state should be explicitly determined. Therefore, we plan the velocity profile after the spatial path generation.

For simplicity, we discretize the spatial path along the curvilinear abscissa. In order to improve the driving safety and comfort, we deduce a velocity upper-bound on each discrete point via considering several constraints as follows.

- Maximal allowed speed V_{limit1} :

$$v(s_i) \le V_{\mathrm{limit1}}(s_i) \tag{7}$$

V_{limit1} can be determined by the high-level behavioral planner by reasoning about road conditions, traffic rules and so on.

- Maximal allowed lateral acceleration $Acc_{lateral}$:

$$v(s_i) \le \sqrt{\frac{Acc_{lateral}}{|\kappa(s_i)|}} \tag{8}$$

To prevent the vehicle tire force from entering into the nonlinear saturated zone and reduce control efforts for the yaw motion stabilizing, we set the lateral acceleration limit.

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

$$v_{\min} \le v(s_i) \le \sqrt{v^2(s_0) + 2Acc_{lon}s_i}$$

$$v_{\min} = \begin{cases}\sqrt{v^2(s_0) + 2Dec_{lon}s_i} & if\ v^2(s_0) + 2Dec_{lon}s_i > 0 \\ 0 & otherwise\end{cases} \tag{9}$$

Acc_{lon} and Dec_{lon} are set to avoid abrupt acceleration and deceleration motions.

- Maximal braking deceleration limit :

$$v(s_i) \le \sqrt{v^2_{\mathrm{terminal}} + 2Dec_{\max}(s_N - s_i)} \tag{10}$$

Where Dec_{max} denotes the maximal braking deceleration. In order to guarantee safety, the vehicle is required to stop or reach a certain speed (e.g. it can be determined by reasoning about road conditions) at the end of the trajectory.

Based on the aforementioned constraints, the maximal velocity limits along the path can be determined. To enhance the driving speed, we assign the maximal velocity along the planned path to be the minimal value of the four speed bounds calculated by (7)-(10). Then, we generate a trapezoidal linear velocity profile connecting the current velocity with the terminal velocity, while satisfying aforementioned constraints. After that, to obtain an acceleration-continuous velocity profile, we use a parametric polynomial spline to smooth it.

IV. Low-Level Tracking Control

The object of the tracking controller is to track the generated trajectory as accurately as possible. There exists substantial work on this subject. Since path planning and velocity planning have been decoupled in the trajectory planning process, it is able to decompose the trajectory tracking control task into longitudinal control and lateral control. For the longitudinal control, we use a proportional and internal model control cascade controller (P-IMC) to generate throttle and brake control commands to track the desired velocity profile [25]. In this section, we primarily focus on the lateral control.

Since velocity and curvature profiles could be easily derived from the generated trajectory, it means that the yaw rate commands could be obtained as well. Therefore, steering control could be transformed into yaw motion stabilizing control. The desired yaw rate commands could be used to obtain feed-forward steering control inputs.

When vehicle navigates at a low speed, a geometric steering control law could be applied to track the desired curvature profile.

$$\kappa = \tan\delta / L \tag{11}$$

Where δ is the front-wheel steering angle, L is the wheelbase. It represents the geometric relation between the steering angle and curvature of the rear wheel at low speeds. However, when the vehicle drives at a high speed, large lateral force is required for steering. In this situation, tire sideslip effects cannot be neglected, the relation between the steering angle and the instant responsive curvature exhibits strong nonlinearities. In this case, in order to achieve high steering control performance, vehicle lateral dynamics has to be accounted for. Previous work on vehicle dynamics control shows that it is possible to define the steady-state steering characteristics when the vehicle negotiates a constant-curvature curve at a constant speed in a non-time-varying condition [26].

As shown in Fig. 6, the angles α_f and α_r are the sideslip angles of front and rear tires respectively. Assuming the vehicle is driving on a flat plane and the steering radius of the center of gravity (C.G.) R is much greater than the wheelbase L, i.e. $R >> L$, we can easily obtain

$$\delta \approx L / R + \alpha_f - \alpha_r \tag{12}$$

When the vehicle reaches steady state, lateral forces acting on front and rear tires generate the centripetal acceleration as

$$F_{yf} + F_{yr} = mv_x^2 / R \tag{13}$$

Where m is the vehicle mass, F_{yf} and F_{yr} are the front and rear tire lateral forces respectively, and v_x is the longitudinal velocity. According to the yaw moment equilibrium, we obtain

$$F_{yf} l_f = F_{yr} l_r \tag{14}$$

When sideslip angles are small (in the trajectory planning stage, we consider the maximal lateral acceleration in order to limit sideslip angles of the tires), lateral forces can be approximately estimated to be linear with sideslip angles.

$$F_{yf} = C_{\alpha f}\alpha_f \qquad F_{yr} = C_{\alpha r}\alpha_r \tag{15}$$

Where $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness of front and rear tires respectively. According to (12)-(15), the steady-state relationship between front-wheel steering angle and the expected curvature is characterized as

$$\delta = \kappa(L + K_v v_x^2), \;\; K_v = (\frac{l_r}{C_{\alpha f}} - \frac{l_f}{C_{\alpha r}})\frac{m}{L} \tag{16}$$

Where K_v is named understeer gradient. Since the cornering stiffness may vary in different road conditions, online system identification methods can be used.

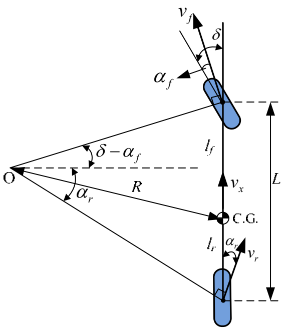


Figure 6. Bicycle model considering sideslip effects.

By using (16), the feed-forward steering control commands can be employed to track the desired yaw rate profile. In practice, due to modeling uncertainties (such as varying yaw dynamics, actuator dynamics) and external noises (such as bank angle), using the feed-forward control law may inevitably cause control errors. In order to reject the external disturbance and achieve a high degree of robustness, we develop a lateral control law based on the inner model control strategy. As shown in Fig. 7, the proposed inner model control strategy combines the feed-forward control with the feedback compensator. The discrepancy of the measured curvature (or yaw rate) and the expected curvature from the reference model can be compensated by the feedback compensator. In this way, it can significantly reduce feedback control efforts and allow the feedback controller to focus on compensating for the tracking errors resulting from the model uncertainties and external disturbances.

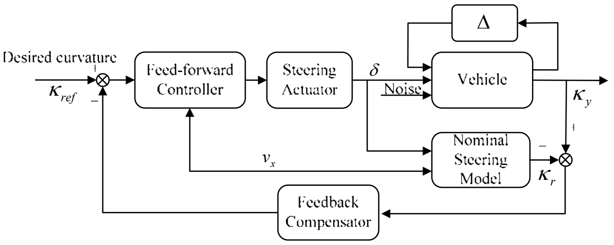
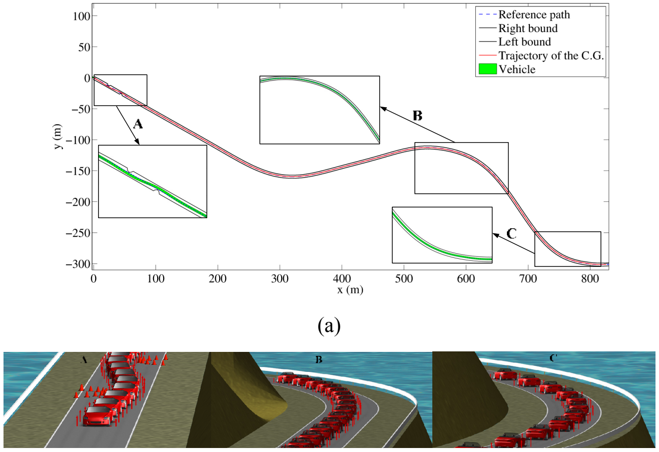


Figure 7. The inner model control framework for steering control.

V. Simulation Results

To validate the effectiveness of the proposed method, we conduct simulations on a simulated environment established by coupling Matlab with Carsim®, which is a commercially-available high-fidelity full-vehicle dynamics solver. In the simulation, we use a coastal highway scenario and a full-size B-class passenger vehicle, which are provided by the Carsim. We set traffic cones along the road as static obstacles, while dynamic traffic participants are not considered in this paper. The initial vehicle speed is set to be 60km/h and the maximal speed is limited to be 100km/h. The maximal absolute value of the longitudinal and lateral acceleration are limited to be 3m/s^2 and 5m/s^2 respectively. The cycle time of trajectory planning and low-level tracking control are set to be 100ms and 20ms respectively. The maximal look-head distance is set to be 50m. The response of low-level actuators (steering, throttle and brake systems) are considered as first-order delay process.

Fig. 8 (a) shows entire tracking result of the vehicle along a reference road course with two bounds. The vehicle is capable of avoiding the static obstacles while keeping in the corridor when it navigates at a high speed. The snapshots of scenario A, B and C in Fig. 8 (b) depicts some details when the vehicle is avoiding static obstacles and negotiating two tight turns. Fig. 9 illustrates the snapshots of the local path planning scheme in the scenarios above. Fig. 10 depicts the corresponding lateral tracking error of the center of gravity (C.G.) of the vehicle. It can be seen that the lateral tracking errors are less than 1m during the entire course. As shown in Fig. 11, the longitudinal is adjusted according to the reference path geometry in order to guarantee the safety and comfort while adhering to the speed limits in different road segments.



(a)

(b)

Figure 8. The entire vehicle path tracking result along a reference corridor with two bounds, A, B and C illustrates details of three scenarios.

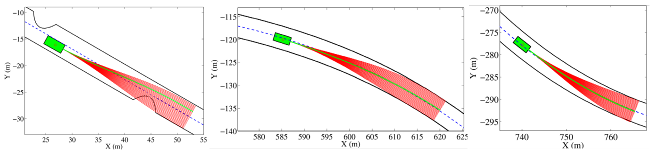


Figure 9. Snapshots of local trajectory planning in three scenarios.

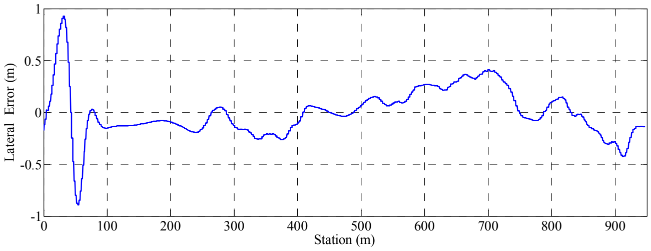


Figure 10. The lateral tracking error with respect to the reference path.

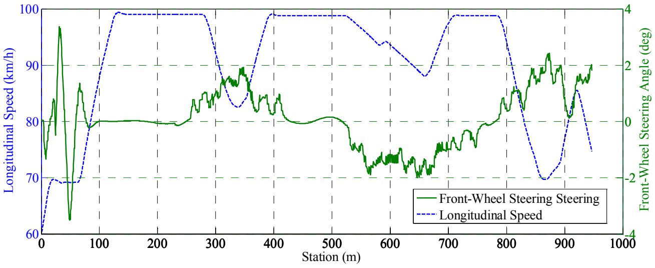


Figure 11. Vehicle speed and front-wheel steering angle

VI. Conclusions and Future Work

This paper has proposed a local navigation strategy for autonomous ground vehicle driving along a reference path by combining the local trajectory planning and tracking control in a unified framework. In order to smoothly follow the reference path and handle unpredictably changing environments reactively as well as improve the driving safety and comfort, we develop a sampling-based trajectory planner following a discrete optimization scheme in the trajectory planning stage. To ensure the dynamic-feasibility of the planned trajectory, the planner considers geometry information of the reference path, static obstacle avoidance, vehicle kinematic and partial dynamic constraints as well. During the low-level tracking control stage, vehicle steady-state steering characteristics is employed to generate the steering commands, which are used to track the desired yaw rates derived from the selected optimal trajectory.

The simulation results demonstrate the capabilities of the proposed integrated local trajectory planning and tracking control framework to follow a reference path while avoiding static obstacles at a high speed. In the future, we will continuously work on the extension of the proposed framework. For instance, one possible improvement is to integrate the local path planning strategy with the graph-search path planner in the cluttered environments and generate dynamically-feasible trajectories in continuous space instead of in the discrete state space. Strongly nonlinear vehicle dynamics should be further accounted for to handle various road conditions. Besides, understanding the scenarios and interacting with other traffic participants (such as other cars, cyclists and pedestrians) like human drivers are also required to be taken into consideration in future.

References

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

[2] Ü. Özgüner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: the DARPA Grand Challenge experience," *Proceedings of IEEE*, vol. 95, no. 2, pp. 397–412, 2007.

[3] K. Redmill, Ü. Özgüner and et al., "Ohio state university experiences at the DARPA Challenges," *SAE Int. J. Commer. Veh.*, vol. 1, no. 1, pp. 527–533, 2009.

[4] J. Levinson, et al., "Towards fully autonomous driving: systems and algorithms," *IEEE Intell. Veh. Symp.*, pp. 163–168, 2011.

[5] J. B. Derrick and D. M. Bevly, "Adaptive steering control of a farm tractor with varying yaw rate properties," *J. Field Robotics*, vol. 26, no. 6–7, pp. 519–536, 2009.

[6] X. Min and M. A. Minor, “Variable structure backstepping control via a hierarchical manifold set for graceful ground vehicle path following,” *IEEE Int. Conf. on Robotics and Auto.*, pp. 2826–2832, 2013.

[7] J. M. Snider, "Automatic steering methods for autonomous automobiles," Tech. Rep. CMU-RI-TR-09-08, Robotics Institute, Pittsburgh, PA. Feb. 2009.

[8] P. Falcone, et al., “Predictive active steering control for autonomous vehicle systems,” *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, 2007.

[9] S. J. Anderson, et al., “An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios,” *Int. J. Vehicle Autonomous Systems*, vol. 18, no. 2–4, pp. 190–215, 2010.

[10] C. E. Beal and J. C. Gerdes, “Model predictive control for vehicle stabilization at the limits of handling,” *IEEE Trans. on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013.

[11] T. M. 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.

[12] 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.

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

[14] A. Lacaze, et al., “Path planning for autonomous vehicles driving over rough terrain,” In *Proc. of IEEE Intl. Symp. Intell. Control*, 1998.

[15] A. Kelly and T. Stenz, "Rough terrain autonomous mobility – part2: An active vision and prediction control approach," *Autonomous Robots*, vol. 5, pp. 163–198, 1998.

[16] E. Frazzoli, M. A. Dahleh and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guid. Control Dynam.*, vol. 25, no. 1, pp. 116–129, 2002.

[17] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Trans. on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.

[18] S. Thrun, et al., “Stanley: The robot that won the DARPA grand challenge,” *J. Field Robotics*, vol. 23, no. 9, pp. 661-692, 2006.

[19] 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.

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

[21] 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.

[22] 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.

[23] M. McNaughton, et al., “Motion planning for autonomous driving with a conformal spatiotemporal lattice,” in *IEEE Int. Conf. on Robotics and Auto.*, pp. 4889-4895, September 2011.

[24] J. Ziegler, et al., “Trajectory planning for BERTHA- a local, continuous method,” *IEEE Intell. Veh. Symp.*, pp. 450–457, 2014.

[25] J. Wang, et al., “Adaptive speed tracking control for autonomous land vehicles in all-terrain navigation: an experimental study,” *J. Field Robotics*, vol. 30, no. 1, pp. 102–128, 2013.

[26] R. Rajamani, Vehicle Dynamics and Control. New York: Springer, 2006, ch. 2–3.