



Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles



Xiaohui Li ^a, Zhenping Sun ^{a,*}, Dongpu Cao ^b, Daxue Liu ^a, Hangen He ^a

^a College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China

^b Department of Automotive Engineering, Cranfield University, Bedford MK43 0AL, UK

ARTICLE INFO

Article history:

Received 14 July 2015

Received in revised form

29 September 2015

Accepted 9 October 2015

Keywords:

Autonomous ground vehicles

Local trajectory planning

Tracking control

Obstacle avoidance

Model-based predictive trajectory generation

ABSTRACT

This study proposes a novel integrated local trajectory planning and tracking control (ILPTC) framework for autonomous vehicles driving along a reference path with obstacles avoidance. For this ILPTC framework, an efficient state-space sampling-based trajectory planning scheme is employed to smoothly follow the reference path. A model-based predictive path generation algorithm is applied to produce a set of smooth and kinematically-feasible paths connecting the initial state with the sampling terminal states. A velocity control law is then designed to assign a speed value at each of the points along the generated paths. An objective function considering both safety and comfort performance is carefully formulated for assessing the generated trajectories and selecting the optimal one. For accurately tracking the optimal trajectory while overcoming external disturbances and model uncertainties, a combined feedforward and feedback controller is developed. Both simulation analyses and vehicle testing are performed to verify the effectiveness of the proposed ILPTC framework, and future research is also briefly discussed.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Autonomous ground vehicles (AGVs) have great potential to improve driving safety, comfort and efficiency and can be widely applied in a variety of fields, such as road transportation, agriculture, planetary exploration, military purpose and so on [1–4]. The past three decades have witnessed the rapid development of AGV technologies, which have attracted considerable interest and efforts from academia, industry, and governments. Particularly in the past decade, contributing to significant advances in sensing, computer technologies, and artificial intelligence, the AGV has become an extraordinarily active research field. During this period, several well-known projects and competitions for AGVs have already exhibited AGV's great potentials in the areas ranging from unstructured environments to the on-road driving environments [5,6].

The development and application of AGVs requires a variety of the-state-of-the-art technologies, among which, motion planning and control play a critical role in improving safety and comfort for autonomous driving. There has been substantial research on motion planning and path tracking control for AGVs [7–22]. Due to the limited on-board computational resources, most of the previous approaches address the motion planning and tracking control problems separately [8]. More

* Corresponding author. Fax: +86 073184574383.

E-mail address: sunzhenping@outlook.com (Z. Sun).

specifically, the high-level motion planner often takes advantage of powerful discrete graph search-based algorithms based upon a simplified point-mass vehicle motion model to compute a long-term and collision-free path. The speed law is often assumed to be a constant value. Then the low-level path tracking controller focuses on regulating the vehicle onto the planned path using kinematic and/or dynamic lateral controllers [12–14].

As shown in Fig. 1, most of the conventional Lyapunov-based linear and nonlinear feedback path tracking controllers derive the steering control laws by relying on cross-track errors without explicitly considering the prediction of future motions or environmental constraints. The steering control command δ is derived by minimizing the lateral error Δy and heading error $\Delta\theta$. In addition, the collision avoidance problem is also neglected. It is known that the smoothness of the reference path has a significant impact on the control performance. If the reference path is collision-free and smooth, only using gain-scheduling or sliding-model feedback control laws can achieve fairly satisfactory results. However, in practice, the reference paths from the high-level path planner are generally not smooth or even violate the vehicle kinematic and/or dynamic constraints, and tracking the reference paths directly using cross-track error-based feedback controllers can therefore easily result in response overshoot, oscillation or even instability, which is especially critical when the reference path is highly curvy or curvature-discontinuous. To alleviate these negative impacts, optimization techniques and path deformation algorithms have been developed to refine and smoothen the planned path for execution [23]. Smoothening a path with considering vehicle constraints and obstacles avoidance often involves a complex nonlinear optimization process, which makes computation prohibitively expensive. Besides, due to localization errors and unpredictably changing outdoor environments, the initially planned collision-free path may still collide with obstacles at the execution stage. Therefore, when a vehicle drives in a dynamic environment along a guidance path, it is necessary to develop a local motion planner to handle the online updating surrounding perceptual information.

The literature indicates that a better communication or integration between the high-level motion planning and the low-level tracking control would be very valuable for enhancing the overall performance of AGVs. To bridge this gap, a real-time trajectory planner is necessary and is required to be capable of taking both the information of the guidance path and vehicle motion constraints into account. Also, it should generate expressive drivable trajectories to make full use of vehicles' maneuverability to handle dynamic environments. To ensure that the generated trajectory could be smoothly and accurately tracked, instead of solely using the cross-track error, the low-level trajectory tracking controller should take advantage of the information provided by the trajectory planner to derive the feedforward control input to stabilize the vehicle and guide it along the planned trajectory as well as employ the feedback controller to overcome the model uncertainties and external noises, while guaranteeing the control stability [12–14]. In this study, we focus on developing an efficient local trajectory planner and the corresponding controller for AGVs to smoothly follow the reference path while avoiding unexpected obstacles. During the trajectory planning process, the vehicle kinematic model and control constraints are explicitly considered to generate smooth and curvature-continuous spatial paths. Furthermore, to ensure driving safety and comfort, velocity profiles are also carefully generated and assigned along the trajectory. After that, vehicle dynamics is also taken into account during the tracking control process to enhance the tracking control performance.

Since the AGV trajectory generation and tracking control problem involves dealing with constraints imposed by the vehicle control model, it can be naturally formulated into a constrained optimal control problem. Applying optimization techniques to solve these problems is not new. Very recently, some researchers employed linear and nonlinear model predictive control (MPC) approaches to address vehicle trajectory generation and tracking problems [24–27]. Due to its capabilities of systematically handling system nonlinearities, state and control constraints, MPC has become a well-known method to solve trajectory generation and tracking control problems. However, solving the constrained optimization problem in the continuous control space over a long-term prediction horizon often involves a complex optimization process, which may easily result in excessive computational burden. Besides, constraints imposed by irregularly distributed obstacles in outdoor environments are difficult to be handled. To overcome these difficulties, a great deal of research on sampling-based local trajectory planning approaches has been conducted. They are capable of generating feasible trajectories along with the corresponding control inputs. Most of these methods follow a discrete optimization scheme, i.e. generation, collision test and evaluation. These approaches can be roughly classified into two types: control-space sampling and state-space sampling [15].

The control-space sampling method refers to generating a finite control input subset based on the parameterized control input space, for instance, parameterized curvature, such as arcs, clothoids, polynomial spirals. Then, based on the sampled control inputs, the trajectories are generated through forward simulation of the differential equations, respecting vehicle system constraints. Therefore, the produced paths are intrinsically drivable. Owing to its simplicity and efficiency, the

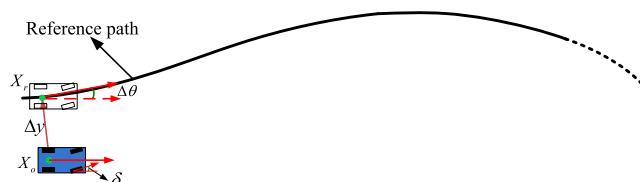


Fig. 1. Conventional Lyapunov-based feedback path tracking control strategy.

method has been widely applied for local navigation. To reduce online computational complexity, some researchers proposed approaches by generating a set of body-centered motion primitives offline, applied them via translation and rotation online, and then evaluated the primitives via the convolution of the vehicle shape along these primitives on a cost-map [15,16]. However, the control-space sampling methods also suffer from shortcomings: it does not exploit the information of environments and the reference path during the sampling process. As the purple dash dot line curves illustrate in Fig. 2, the uniformly sampled trajectories are not aligned with the reference path. As a consequence, it may result in discrepancy between the consecutive plans and cause overshoot and oscillation in the tracking control process. Secondly, due to the nonlinearities of the system constraints, though the uniform sampling performs in the control input space, the generated paths may not be well separated. Consequently, it results in computational complexity during the collision-test and evaluation process.

In contrast, instead of sampling in the control space, the state-space sampling method requires generated trajectories to satisfy the constraints imposed by the terminal states and the vehicle model. This approach samples a set of terminal states firstly, then computes the trajectories, which connect the initial state with the sampling states while respecting system constraints. Furthermore, the state-space sampling scheme is able to exploit the information of environmental structure and guidance path or from sampling or collision-test results to bias sampling toward the space, where the potential solutions are most likely to exist. In this way, it can significantly reduce unnecessary samples. Pure pursuit control is a well-known preview path tracking controller, which is essentially a simplified version of state-based sampling planning approach. As the orange dash line curves show in Fig. 2, the essential idea of the pure pursuit approach is regulating the vehicle onto the reference path with satisfying the position constraints. However, it does not explicitly consider the constraints imposed by heading and curvature states. Based upon the state-space sampling strategy [18,28,29], computed system-compliant trajectories via forward simulation using the vehicle system model and associated close-loop feedback control laws.

Howard et al. [7], formulated the trajectory generation problem into a two-point boundary value problem (BVP). Since it involves inverting nonlinear differential equations, solving the BVP directly will be nontrivial. The well-known Dubins curves transformed the BVP into a path-length optimization problem and used an analytical approach to generate path by using circular arcs and line segments. However, curvature discontinuities at the conjunction of the line segments and arcs may cause stop and reorientation actions. Another popular analytical method, Bézier curves are able to ensure the curvature continuity, but they cannot satisfy the constraints of curvature limits and its rates limits [30]. Montemerlo et al. [6] proposed a geometrical path generation method, which generated a finite set of paths with terminal states aligned with a baseline (reference path). The method is especially efficient when a curvature-continuous reference path is available. However, it did not explicitly take vehicle control constraints into account.

To explicitly deal with vehicle model and control constraints, Howard et al. [7] developed a general and efficient model-predictive trajectory generation approach via numerical linearization and optimization techniques. The approach can also be used for generating feasible edges to construct global search graphs, such as regular state lattice [17]. It has been successfully applied in a variety of mobile robotic platforms. Combining the trajectory generation method and the conformal spatio-temporal lattice, McNaughton et al. proposed a powerful Graphic Processor Unit (GPU) based motion planner to solve several challenging motion planning problems in on-road driving scenarios [16].

It can be seen from the aforementioned research, most of them deal with the trajectory generation and tracking control problem separately. In this paper, a new integrated local trajectory planning and tracking control (ILPTC) framework is developed for online autonomous driving with obstacles avoidance to bridge the gap between planning and control. The presented ILPTC framework consists of an efficient sampling-based local trajectory planning algorithm which explicitly considers the time parameterized states. Subsequently, a tightly associated tracking controller directly exploits the rich information from the trajectory planner by combining feedforward and feedback controllers.

2. The ILPTC framework and algorithms

In this paper, we assume that a rough reference path could be obtained as prior information. In practice, when AGVs drive in outdoor environments, a long-term reference path could be obtained from the high-level route planner, or extracted

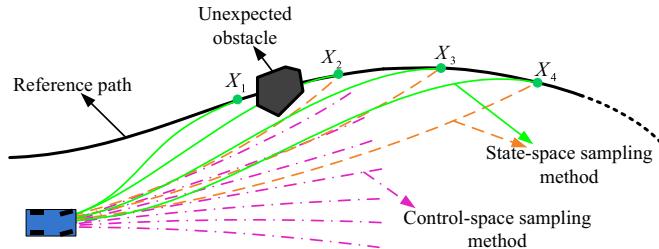


Fig. 2. Control-space sampling trajectory planner and state-space sampling trajectory planner. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

from the online perceptual information, such as road lanes, or the high-level graph-search planner. It is able to prevent the vehicle from being stuck into local minima and avoid excessively reactive actions. However, we do not require the reference path to be collision-free or curvature-continuous. To track the reference path smoothly and react to the real-time perception surrounding environment, an integrated local trajectory planning and tracking control (ILPTC) system architecture is developed, as shown in Fig. 3.

The global route planner computes the fastest route based on the missions. The behavioral planner employs the global path plan result and the online sensing information to generate informed decisions for the local trajectory planner to execute. When the vehicle drives in different scenarios, the information provided by the behavioral planner is also different. For instance, when the vehicle drives in urban environments. The behavioral planner will reason about the complex traffic situations based on the perception information and traffic rules. Then it provides the maximal allowed speed as well as discrete decision commands to the trajectory planner, such as lane keeping, lane changing, stop in front of the stopping line and so forth. Based upon these decisions, the solution space of the trajectory planner could be significantly reduced. In this study, by using the informed decisions provided by the behavioral planner, the trajectory planner could determine its target reference path, the maximal allowed speed as well as the terminal speed.

The local trajectory planner is developed based on the generation-and-evaluation scheme. We apply an informed state-based sampling scheme first, which samples a rich set of terminal states aligned with the reference path. The environmental constraints and high-level behavior planner directives can be incorporated into the sampling scheme to reduce computational complexity. Then, based on the model-predictive trajectory generation method, multiple dynamically-feasible trajectories are generated to connect the initial state with the terminal states while respecting the vehicle model and control constraints. Meanwhile, the corresponding nominal control input sequences are also obtained. After that, collision-test is performed to trim the trajectory colliding with obstacles. The remaining collision-free trajectory candidates are evaluated according to an objective function, which accounts for the proximity to obstacles, deviation from reference path, smoothness, consecutive consistency and executive speed. The best trajectory is selected and tracked by a designed

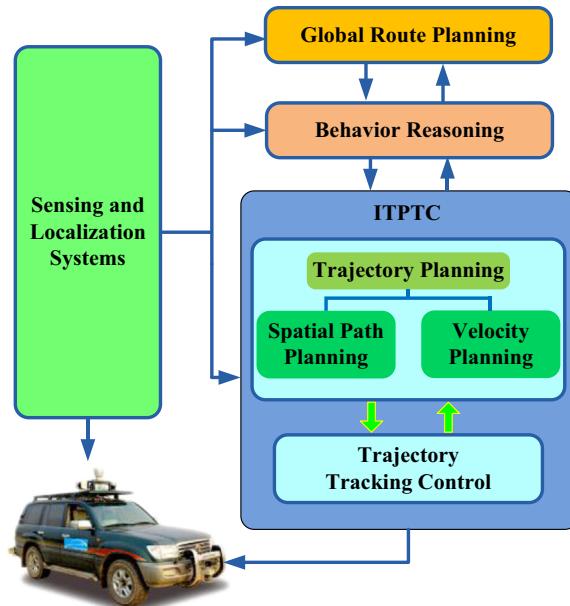


Fig. 3. A schematic navigation framework for AGVs by integrating ILPTC.

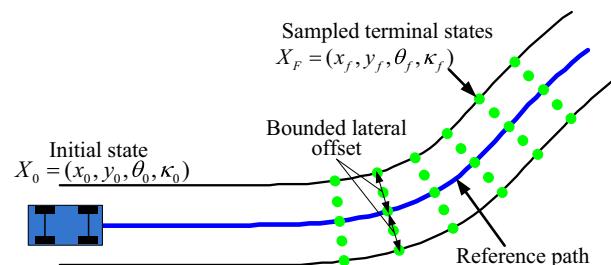


Fig. 4. A uniform state-space sampling scheme. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

combined feedforward and feedback trajectory tracking controller. The entire process is periodically running within a short planning cycle.

2.1. Local trajectory planning

This section formulates the local trajectory planning strategy, which is composed of state sampling, trajectory generation, collision-test and evaluation.

2.1.1. State-space sampling strategy

Firstly, we develop an intuitive uniform sampling strategy by using the reference path information. As illustrated in Fig. 4, the blue curve is the reference path, to avoid corner cutting and ensure smoothness of path following, four dimensions states X_i ($x_i, y_i, \theta_i, \kappa_i$) are sampled along the reference path at different preview distances. Where (x_i, y_i) is the position, θ_i is the heading, and κ_i is the curvature. To ensure control stability, the preview distance is usually tuned based on the current velocity. As represented in (1), the preview distance d_{preview} can be determined as a function of velocity v with the minimal limit d_{\min} and maximal limit d_{\max} . For instance, $f(v)$ could be a linear function of the current velocity and d_{\min} should be greater than the minimal crash distance.

$$d_{\text{preview}} = \begin{cases} d_{\min} & f(v) < d_{\min} \\ f(v) & \text{otherwise} \\ d_{\max} & f(v) > d_{\max} \end{cases} \quad (1)$$

To obtain smooth trajectories as well as avoid unexpected obstacles, multiple lateral offsets are sampled along the reference path, as the green circles shown in Fig. 4. Generally, the heading and curvature states of the sampling points are set to be same as those of the corresponding nearest points along the reference path to ensure the generated trajectories are aligned with the reference path. Fig. 4 displays a uniform sampling scheme.

In practice, if the planning time allows, the sampling density could be increased to produce more trajectory candidates. In addition, an informed sampling strategy can be applied to achieve bias sampling by exploiting the information on environmental structures, the reference path as well as high-level behavior directives.

2.1.2. Trajectory generation

Once the terminal states have been determined, the trajectory generation problem can be naturally formulated into a BVP, subject to the constraints of boundary states and vehicle predictive motion model. Inspired by the work proposed in [7], a model-predictive trajectory generation approach is developed to solve the trajectory generation problem.

In general, the vehicle motion model can be represented by a set of nonlinear differential equations along with the control constraints as follows

$$\dot{X} = f(X, U) \quad (2)$$

$$U_{\min} \leq \|U\| \leq U_{\max}, \dot{U}_{\min} \leq \|\dot{U}\| \leq \dot{U}_{\max}$$

In this study, the vehicle kinematic model is employed as the predictive motion model. Of course, the vehicle dynamic model could also be used to predict the consequences of control inputs, but, it usually involves online identification of a variety of time-varying parameters, which directly affect the accuracy of the forward evolution. To accurately predict the consequences of the control inputs, these time-varying parameters are required to be precisely identified online. Furthermore, in practice, it is difficult to predict the effects on the vehicle dynamics caused by the interactions between the terrain and the vehicle. Therefore, during the trajectory process, the vehicle kinematic model is used instead of the dynamic model for forward propagation. The vehicle kinematic model can be represented by the following differential equations.

$$\dot{x} = v \cos(\theta), \dot{y} = v \sin(\theta), \dot{\theta} = \kappa v, \kappa = u \quad (3)$$

where (x, y) is position, θ is angle of the vehicle velocity, v is velocity, κ is curvature, and u is steering control input. The kinematic model uses the simplified mass-point model, which describes the motion of the center of the rear axle. In practice, when vehicles drive at high speeds, abrupt steering actions may lead to control instability. To ensure physical feasibility of steering actions (such as steering angle and its rate limits) and improve safety and smoothness of the lateral tracking control (avoid the tire lateral forces entering into nonlinear or saturated zone and alleviate sideslip effects), we explicitly considered the curvature upper-bound κ_{\max} and the upper-bound constraints on its first-order derivative $\dot{\kappa}_{\max}$. They are defined as

$$\kappa \leq \|\kappa_{\max}\|, \quad \left\| \frac{d\kappa}{dt} \right\| \leq \dot{\kappa}_{\max} \quad (4)$$

Based on the kinematic model, the trajectory generation can be naturally decomposed into spatial path generation and velocity profile generation via the following transformation

$$\frac{d\xi}{dt} = \frac{d\xi}{ds} \frac{ds}{dt} = \frac{d\xi}{ds} v \quad (5)$$

Substituting (5) into (3), the integral formation of vehicle states and the control constraints are derived as

$$\begin{aligned} x(s) &= \int_0^{s_f} \cos(\theta(s))ds \\ y(s) &= \int_0^{s_f} \sin(\theta(s))ds \\ \theta(s) &= \int_0^{s_f} \kappa(s)ds \\ \kappa(s) &= u(s) \\ \|\kappa(s)\| &\leq \kappa_{max}, \quad \|\kappa(s)\| \leq v\kappa_{max} \end{aligned} \quad (6)$$

Note that the states of the spatial path can be represented as functions with respect to curve length s instead of time. In other words, these states become velocity-independent. So the spatial path generation problem can be transformed into a problem of determining a sequence of curvature control inputs $\kappa(s)$, which steer the vehicle from the initial state $X_0(x_0, y_0, \theta_0, \kappa_0)$ to the terminal state $X_F(x_f, y_f, \theta_f, \kappa_f)$. Considering the system model, it refers to inverting the nonlinear differential equations. Solving the problem in the continuous control space is impractical. One potential approach to address the problem is nonlinear programming. However, it is computationally demanding. In order to achieve computational efficiency, the control input curvature κ could be parameterized as a polynomial function with respect to the path length s , as referred in [18].

$$\kappa(s) = \kappa_0 + \kappa_1 s + \kappa_2 s^2 + \dots \quad (7)$$

It is capable of effectively representing the control space by using only a few parameters matrix $P[s_f, \kappa_1, \kappa_2, \dots]$, but it may be at the cost of reducing solution space and introducing sub-optimality.

Now the BVP is reformulated as solving the unknown parameters matrix P . Since the differential equations of the predictive model involve strong nonlinearities, it is nontrivial to obtain an analytical solution. A numerical method is employed to provide the planner capabilities to handle arbitrary vehicle models and control constraints. More specifically, the unknown parameters matrix P is iteratively refined by using the Newton's method until the states' differences $\Delta X_F(P(k))$ between the numerically computed terminal states $X_F(P(k))$ and the sampling terminal states X_F are less than the pre-

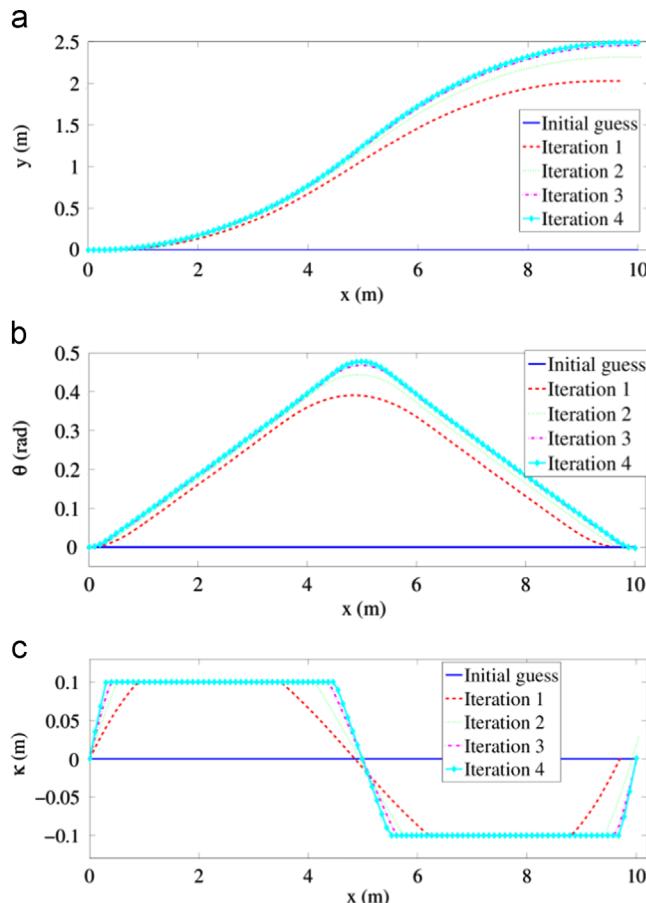


Fig. 5. An example of the iterative process of spatial path generation method: (a) generated trajectories; (b) heading angle; (c) curvature.

defined thresholds or the maximal iteration number is reached [7]. The iterative process is given as

$$\Delta X(P(k)) = X_F - X_P(P(k)) \frac{\partial \Delta X(P(k))}{\partial P} \Delta P = \Delta X(P(k)) \Delta P = \left(\frac{\partial \Delta X(P(k))}{\partial P} \right)^{-1} \Delta X(P(k)) P(k+1) = P(k) + \Delta P \quad (8)$$

In order to obtain a unique solution, the number of unknown parameters in matrix P should be equal to the number of sampling states. Hence, the curvature is set to be a cubic polynomial function of the path length s .

$$\kappa(s) = \kappa_0 + \kappa_1 s + \kappa_2 s^2 + \kappa_3 s^3 \quad (9)$$

Hence, the unknown parameters matrix P becomes $[s_f, \kappa_1, \kappa_2, \kappa_3]$, and the matrix of partial derivatives (or Jacobian matrix) is represented as

$$\frac{\partial \Delta X(P)}{\partial P} \Big|_{P(k)} = \begin{bmatrix} \frac{\partial \Delta x(P(k))}{\partial \kappa_1} & \frac{\partial \Delta x(P(k))}{\partial \kappa_2} & \frac{\partial \Delta x(P(k))}{\partial \kappa_3} & \frac{\partial \Delta x(P(k))}{\partial s_f} \\ \frac{\partial \Delta y(P(k))}{\partial \kappa_1} & \frac{\partial \Delta y(P(k))}{\partial \kappa_2} & \frac{\partial \Delta y(P(k))}{\partial \kappa_3} & \frac{\partial \Delta y(P(k))}{\partial s_f} \\ \frac{\partial \Delta \theta(P(k))}{\partial \kappa_1} & \frac{\partial \Delta \theta(P(k))}{\partial \kappa_2} & \frac{\partial \Delta \theta(P(k))}{\partial \kappa_3} & \frac{\partial \Delta \theta(P(k))}{\partial s_f} \\ \frac{\partial \Delta \kappa(P(k))}{\partial \kappa_1} & \frac{\partial \Delta \kappa(P(k))}{\partial \kappa_2} & \frac{\partial \Delta \kappa(P(k))}{\partial \kappa_3} & \frac{\partial \Delta \kappa(P(k))}{\partial s_f} \end{bmatrix} \quad (10)$$

Since it is nontrivial to analytically solve the Jacobian matrix above, again we take advantage of the numerical method to estimate it via small perturbations.

$$\frac{\partial \Delta X_{ij}(P(k))}{\partial P_j(k)} = \frac{\Delta X_{ij}(P(k) + e_j) - \Delta X_{ij}(P(k))}{e_j} \quad (11)$$

[Fig. 5](#) presents an example to illustrate the iterative process of the aforementioned spatial path generation method. Without losing generality, the vehicle initial state $X_0(x_0, y_0, \theta_0, \kappa_0)$ is given as $[0, 0, 0, 0]$, the terminal state $X_F(x_f, y_f, \theta_f, \kappa_f)$ is set to be $[10, 2.5, 0, 0]$. We consider the curvature constraint as $-0.1 \leq \kappa \leq 0.1$, and curvature rate limits as $-0.5 \leq \dot{\kappa} \leq 0.5$. An initial guess for parameters matrix P is given as $[0, 0, 0, 10]$, so the initial trajectory is a linear segment, which is shown in [Fig. 5\(a\)](#). Then by using the path generation method, the terminal state of the generated spatial paths gradually converges to the sampling terminal state after 4 iterations. As depicted in [Fig. 5\(b\)](#) and [\(c\)](#), the corresponding heading and curvature states are continuous. Meanwhile, the curvature satisfies the required constraints.

Similar to the other nonlinear optimization problems, the initial guess can affect quality and the convergent speed of solutions. A good initial guess for the parameters matrix P can effectively reduce the iterations. In practice, the pre-computed look-up table could be applied to obtain a good initial guess for the online use. As illustrated in [Fig. 6](#), based on the vehicle body-centered coordinate framework, we uniformly sample the terminal states and generate corresponding paths (for clarity, the curvature state is ignored and the sampling resolution is sparse). The parameters of these generated paths are stored in a look-up table.

As [Fig. 7](#) shows, path candidates are generated using the aforementioned spatial path generation method, corresponding to the uniform state-space sampling scheme in [Fig. 4](#). If the planning time is limited, the priority will be given to generate the path candidates with a longer preview distance along the reference path in order to avoid over-reactive actions due to being too short-sighted. As shown in [Fig. 7](#), the trajectory generation sequence is from Layer 1 to Layer 6. If the planning time allows, the sampling density can be incrementally improved. Since all of these trajectories could be parallel computed,

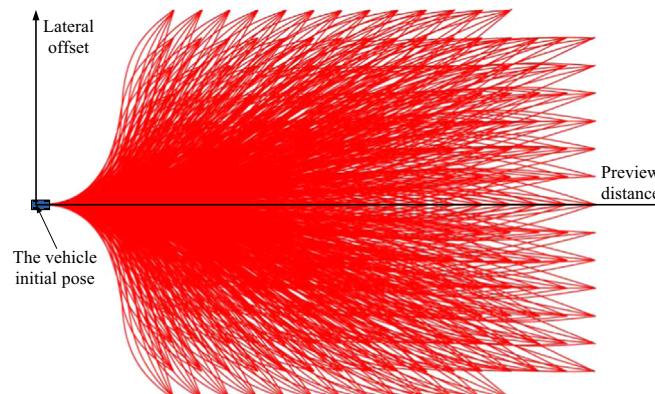


Fig. 6. Offline Look-up table generation.

the parallel computing hardware could be adopted to enhance the sampling density, to further improve the trajectory planning result.

Velocity profile generation performs after the spatial path generation. Longitudinal velocity control is essential for the driving safety and comfort. As a matter of fact, safety and smoothness requirements impose several constraints on the speed profile, such as maximal allowed speed, maximal lateral and longitudinal acceleration, and so on. Here, we propose a three-step velocity profile generation scheme. The maximal speed is determined firstly, then the trapezoidal velocity profile is adopted. After that, to improve smoothness of the longitudinal control, we utilize the polynomial parameterized method to smoothen the linear velocity profiles and generate acceleration-continuous velocity curves.

Firstly, in order to guarantee safety and improve driving comfort, we consider several constraints to determine the maximal speed as follows.

(i) The maximal allowed speed V_{limit1}

V_{limit1} could be obtained from the high-level behavioral planner by reasoning about task requirements, road conditions, traffic rules and so forth.

(ii) The maximal allowed lateral acceleration Acc_{MaxLat}

$$V_{limit2}^2 |\kappa| \leq Acc_{MaxLat} \quad (12)$$

To prevent the lateral force acting on tires from entering into the nonlinear or saturated zone, alleviate sideslip effects, as well as ensure yaw stability, the maximal lateral acceleration should be restricted.

(iii) Maximal longitudinal acceleration limit Acc_{Max_Lon} and deceleration Dec_{Max_Lon} limit.

$$\frac{V_{limit3}^2 - V_0^2}{2Acc_{Max_Lon}} + \frac{V_{limit3}^2 - v_f^2}{2Dec_{Max_Lon}} + D_{safe} = s_f \quad (13)$$

where v_0 is the current velocity, v_f is the terminal speed, and s_f represents the spatial path length. The terminal speed v_f is often set to be zero when the vehicle operates in unstructured environments. When it drives in the on-road environment, v_f is determined by reasoning about traffic conditions. For instance, considering the traffic rules, it is set to be zero when it approximates a stop-line, and it is also required to retain a certain speed by accounting for surrounding traffic participants. Besides, we consider the safe distance D_{safe} , which involves the response delay t_{delay} and safe braking distance.

$$D_{safe} = t_{delay}v_0 + g(v_f) \quad (14)$$

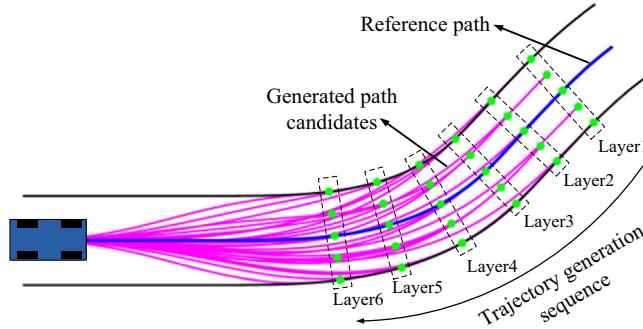


Fig. 7. Spatial path generation result.

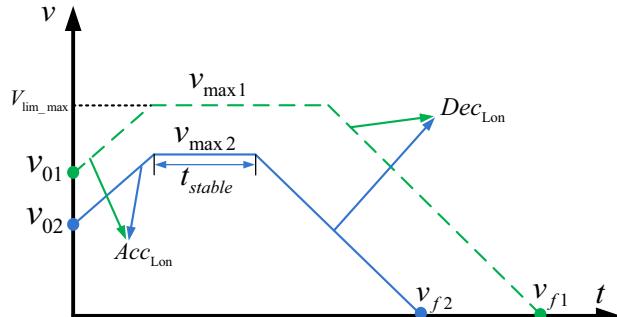


Fig. 8. Trapezoidal velocity profiles. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Based on the constraints above, the maximal velocity could be limited as

$$V_{lim_max} = \min\{V_{limit1}, V_{limit2}, V_{limit3}\} \quad (15)$$

Then, the trapezoidal velocity profile is employed for the longitudinal velocity profile generation. As shown in Fig. 8, for the sake of simplicity, we use the linear velocity profiles with constant acceleration Acc_{Lon} and deceleration Dec_{Lon} values. To ensure smoothness of the longitudinal control, the velocity is required to be kept at the maximal speed v_{max} for a certain period of time, which is equal to or greater than t_{stable} . Therefore, the maximal speed does not necessarily reach the maximal speed limits V_{lim_max} , such as the blue solid velocity curve shows in Fig. 8. Given the initial speed v_0 , terminal speed v_f , the path length s_f , the maximal speed limits V_{lim_max} , and user-specified parameters $\{Acc_{Lon}, Dec_{Lon}, t_{stable}\}$, the trapezoidal velocity profile can be analytically solved.

After that, we smoothen the piece-wise linear velocity profiles using polynomial splines to obtain acceleration-continuous velocity profiles. Inspired by the parametric velocity profile generation method in [31], the velocity can be represented as a cubic polynomial function of time.

$$v(t) = v_0 + at + bt^2 + ct^3 \quad (16)$$

Correspondingly, the acceleration $acc(t)$ and the path length $s(t)$ can be easily derived through derivation and integration of velocity respectively. Given the initial velocity v_0 and acceleration a_0 , the terminal velocity v_f and acceleration a_f , as well as the path length s_f , the unknown parameters $\{a, b, c, t_f\}$ in (17) could be analytically solved via the following equations.

$$\begin{aligned} acc(0) &= a = a_0 \\ v(t_f) &= v_0 + at_f + bt_f^2 + ct_f^3 = v_f \\ acc(t_f) &= a + 2bt_f + 3ct_f^2 = a_f \\ s(t_f) &= v_0 t_f + \frac{1}{2}at_f^2 + \frac{1}{3}bt_f^3 + \frac{1}{4}ct_f^4 = s_f \end{aligned} \quad (17)$$

As shown in Fig. 9, since the ramp-up path length s_1 , and ramp-down path length s_2 could be determined, we solve the acceleration-continuous S-shaped ramp-up velocity profile (the red solid line) and ramp-down velocity profile (the blue dash line) respectively, based upon the proposed velocity profile generation method. In this way, it does not only significantly reduce the velocity search space, but also guarantees driving safety and improve tracking smoothness.

2.1.3. Collision-test and evaluation

The collision-test was performed using the environmental perception information, which could be represented as an occupancy grid-map. As depicted in Fig. 10, to reduce the computational complexity of the collision-test, the rectangular shape of the vehicle could be approximated by a set of circles with the same radius [32]. In order to ensure safety, all of distances between the obstacles and the centers of circles are required to be greater than the circle radius. The strategy could also be extended to deal with the dynamic obstacle avoidance using the spatial-temporal collision-test strategy based on the predictive information of the moving objects in dynamic environments.

After trimming trajectories colliding with obstacles, the remaining collision-free trajectories are evaluated via an objective function comprised of five weighted cost terms as

$$i^* = \arg \min_{i=1}^N (\omega_o J_o(\tau_i) + \omega_d J_d(\tau_i) + \omega_s J_s(\tau_i) + \omega_p J_p(\tau_i) + \omega_c J_c(\tau_i)) \quad (18)$$

where $J_{\{o, d, s, p, c\}}$ are the cost terms, $\omega_{\{o, d, s, p, c\}}$ are the corresponding weighted factors, and i is the index of the trajectory candidates. The optimal trajectory is defined as the one that minimizes the cost function.

The cost term J_o represents the proximity of the trajectory to obstacles. For instance, in unstructured environments, an occupancy grid cost-map could be built upon the perception information. Each cell of the grid-map is assigned a cost value

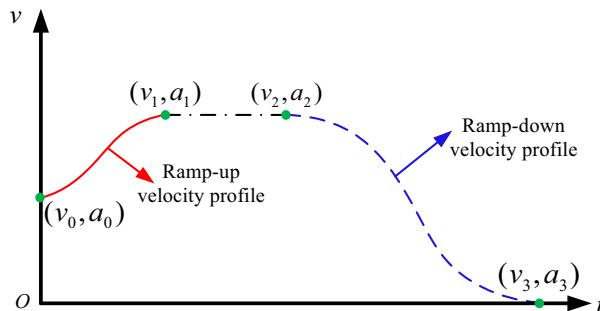


Fig. 9. Smoothing the trapezoidal velocity profile using polynomial splines. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

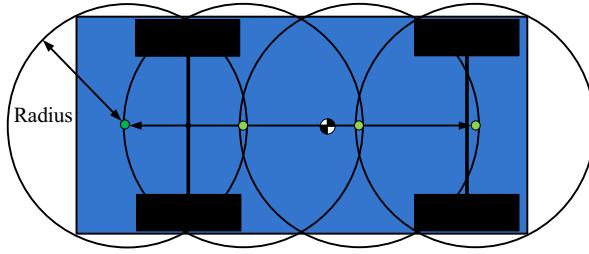


Fig. 10. Circle decomposition of the vehicle shape.

within the range $[0, 1]$, according to its distance to the nearest obstacle. Hence, J_o can be computed via the convolution of the vehicle along the trajectories.

$$J_o = \frac{1}{s_f} \int_0^{s_f} C_o(\tau_i(s)) ds \quad (19)$$

where τ_i ($i=1,\dots,N$) is the evaluated trajectory candidate and s_f is the path length. The cost value is normalized to be within $[0, 1]$.

The cost term J_d penalizes the path deviation from the reference path.

$$J_d = \frac{1}{s_f} \int_0^{s_f} \frac{|D(\tau_i(s))|}{D_{\max}} ds \quad (20)$$

where $D(\tau_i(s))$ is the deviation distance of the trajectory candidate from the reference path and D_{\max} is the maximal deviation distance. In order to avoid jerky lateral movement, we consider smoothness criterion defined by the cost term J_s , which is computed by integrating the curvature of the trajectory.

$$J_s = \frac{1}{s_f} \int_0^{s_f} \frac{\|\kappa(\tau_i(s))\|}{\kappa_{\max}} ds \quad (21)$$

The cost term J_p exhibits the preference of a longer trajectory, which can efficiently prevent aggressive maneuvers resulting from a short predictive horizon

$$J_p = \frac{L_{\max} - l(\tau_i)}{L_{\max}} \quad (22)$$

where $l(\tau_i)$ is the relative path length of the trajectory candidate τ_i along the reference path and L_{\max} is the maximal preview path length along the reference path.

In addition, we also consider the consistency of planning results during the replanning process. In practice, the inconsistency between the consecutive plans can easily result in abrupt steering actions, control overshoots or even control instability. The cost term J_c penalizes the inconsistency between current and previous trajectories. It is computed by integrating the distance between them.

$$J_c = \frac{1}{s_f} \int_0^{s_f} \frac{d(\tau_i(s))}{d_{\max}} ds \quad (23)$$

where $d(\tau_i(s))$ is the Euclidean distance between the point on the current path and the corresponding point on the previous path, and d_{\max} is the maximal value.

In each planning cycle, the optimal trajectory that minimizes the objective function is determined from the generated trajectory candidates. To achieve this, a rich set of path candidates are generated first, the path candidates colliding with road boundaries are trimmed after the collision test. Then, the remaining collision-free trajectories are evaluated based on the aforementioned optimization criteria. The best trajectory is selected and executed by the low-level controller. In practice, the weighted factors could be flexibly tuned in order to adapt to different driving conditions.

2.2. Trajectory tracking control design

In this section, we present the trajectory tracking control design, which involves generating steering and speed control inputs to accurately track the generated trajectory. In this study, we adopt a combined feedforward and feedback control strategy [33]. Since the desired velocity and curvature profiles come along with generation of the trajectory, they can be used as the feedforward control inputs to follow the reference path. More specifically, both velocity and curvature profiles are employed as desired commands. To that end, the feedforward control scheme can be applied to track the desired commands, which can significantly reduce feedback control efforts and allow the feedback controller to focus on compensating for the errors resulting from the model uncertainties and external disturbances.

For AGVs, the trajectory tracking task can be naturally decomposed into two subtasks: longitudinal (or velocity) control and lateral control. For the longitudinal control, we developed a proportional and internal model control cascade controller (P-IMC), which was discussed in detail in [34]. In this section, we primarily focus on the vehicle lateral control.

Based on the velocity and curvature profiles, we can easily obtain the sequence of the open-loop yaw rate commands [35]. For Ackerman-steered vehicles, note that there are direct relationships between the steering angle and desired yaw rate. For instance, when the vehicle navigates at a low speed v , the geometric relationship between the yaw rate r and the steering angle δ can be expressed as

$$r = v \cdot \tan \delta / L \quad (24)$$

where L is the vehicle wheelbase.

However, when the vehicle drives at a higher speed, due to the large lateral force acting on the tires, the sideslip angles rise. To accurately generate the steering controls, the tire sideslip angle (the angle between the direction of the tire and the direction of its velocity vector) should be explicitly considered. Previous research on vehicle lateral dynamics shows that it is possible to employ the vehicle steady-state steering characteristics to define the relationship between the steering angle and the steady response of the yaw rate [35]. As shown in Fig. 11, a dynamic bicycle vehicle model is considered [36]. The angles α_f and α_r are the sideslip angles of the front and the rear tires respectively. Assuming the vehicle is driving on a flat plane and negotiating a constant-curvature curve, and the steering radius R of the center of gravity (C.G.) is much greater

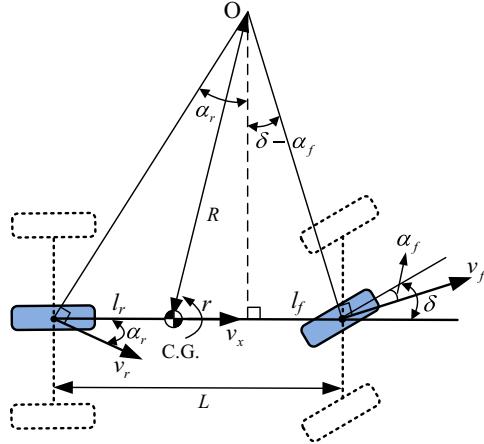


Fig. 11. Dynamic bicycle model considering sideslip effects.

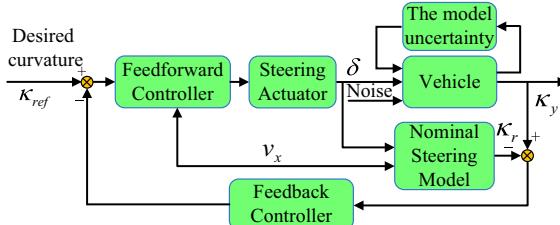


Fig. 12. A combined feedforward and feedback lateral control strategy.

Table 1

Main parameters for a full-size Ackerman-steered vehicle in the simulation environment.

Symbol	Meaning	Value
m	Vehicle mass	1370 kg
I_z	Vehicle yaw moment of inertia	4192 kg m ²
l_f	Distance of C.G. from the front axle	1.110 m
l_r	Distance of C.G. from the rear axle	1.666 m
w	Vehicle body width	1.795 m
C_{af}	Front-tire cornering stiffness	42,670 N/rad
C_{ar}	Rear-tire cornering stiffness	42,670 N/rad

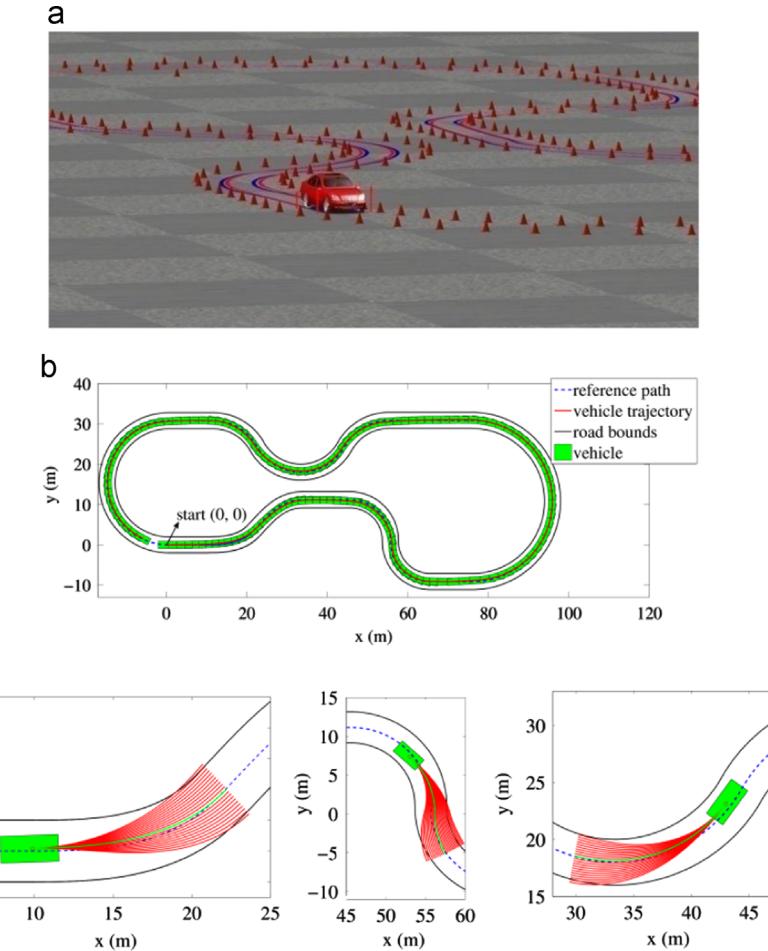


Fig. 13. Path following along the curvy reference path: (a) a snapshot from Carsim simulation environment; (b) the vehicle tracking result; (c) snapshots for local trajectory generation.

than the wheelbase L , i.e. $R > L$, according to geometric relationship, we can obtain

$$\delta \approx L/R + \alpha_f - \alpha_r \quad (25)$$

When the vehicle reaches the steady state, i.e., the vehicle steadily drives along a constant circular arc, the lateral forces F_{yf} and F_{yr} acting on the front and rear tires generate steady centripetal acceleration

$$F_{yf} + F_{yr} = mv_x^2/R \quad (26)$$

where m is the vehicle mass and v_x is the longitudinal velocity. The yaw moment equilibrium yields

$$F_{yf}l_f - F_{yr}l_r = 0 \quad (27)$$

where l_f and l_r are the distances from the centers of the front wheel and rear wheel axles to the C.G. respectively. Assuming that sideslip angles are small (in the trajectory planning process, the maximal centripetal acceleration is limited to ensure satisfaction of the assumption), lateral forces can be approximately estimated to be linear with sideslip angles

$$F_{yf} = C_{af}\alpha_f, \quad F_{yr} = C_{ar}\alpha_r \quad (28)$$

where C_{af} and C_{ar} are the cornering stiffness of front and rear tires respectively.

According to (26)–(28), the nominal relationship between the steering angle δ and the expected curvature κ can be derived as

$$\kappa = \frac{\delta}{L + K_v v_x^2}, \quad K_v = \left(\frac{l_r}{C_{af}} - \frac{l_f}{C_{ar}} \right) \frac{m}{L} \quad (29)$$

where K_v is the vehicle under-steer gradient. Since K_v involves the tire cornering stiffness, which may vary in different road conditions. The online parameter identification approaches can be used to estimate the time-varying parameters.

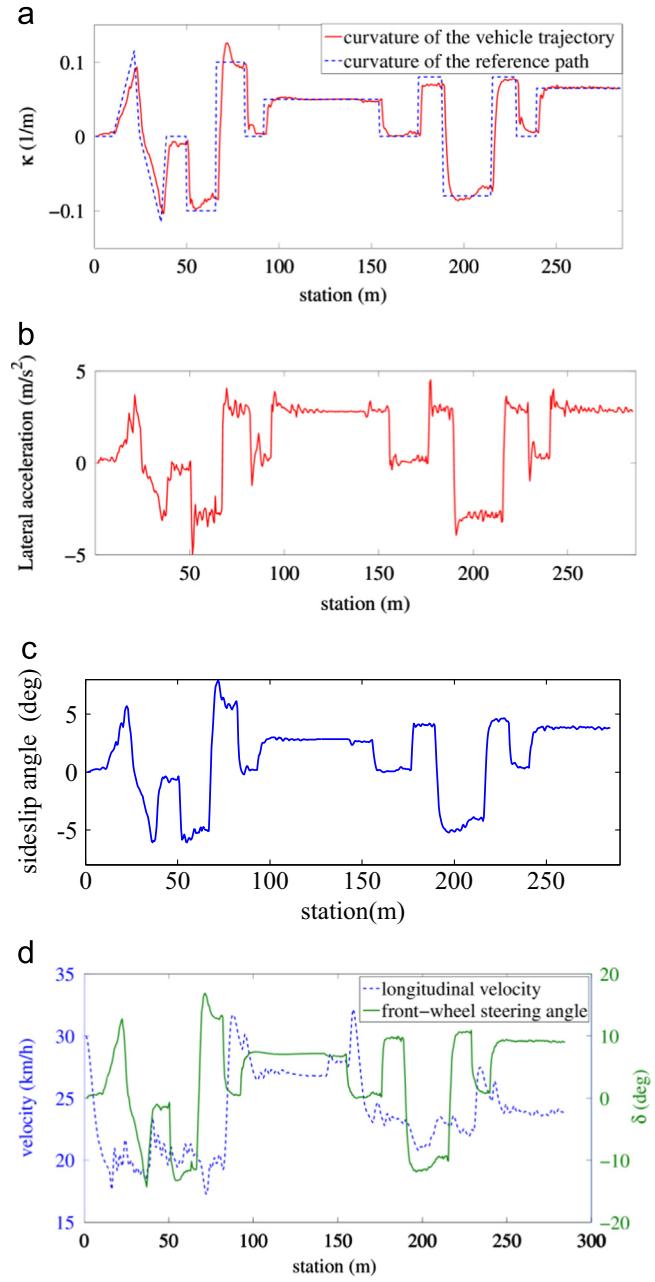


Fig. 14. Trajectory tracking results: (a) the curvature profiles of the vehicle trajectory and the reference path; (b) the lateral acceleration profile; (c) the sideslip angle of C.G.; (d) the velocity and steering angle profiles. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

From (29), the feedforward steering control law can be easily obtained. It performs well when the vehicle drives under ideal road conditions without external disturbances. However, in practice, when AGVs drive in different environments, where road conditions may vary significantly. As a consequence, it will inevitably result in model uncertainties. Besides, time delay appears in the low-level networked control system (such as driver-by-wire units) due to the inertial effects and network-induced delays [37]. Therefore, only using the steering control law in (29) may prevent the vehicle from accurately tracking the desired yaw rate.

To address this issue, we develop a combined feedforward and feedback lateral control strategy, which is shown in Fig. 12. The feedforward control law is derived from the steady-state steering characteristics to track the desired curvature profile, while the curvature tracking errors (differences between the nominal curvature value κ_r and the measured value κ_y resulting from the model uncertainties and external disturbances are compensated by using the feedback controller.

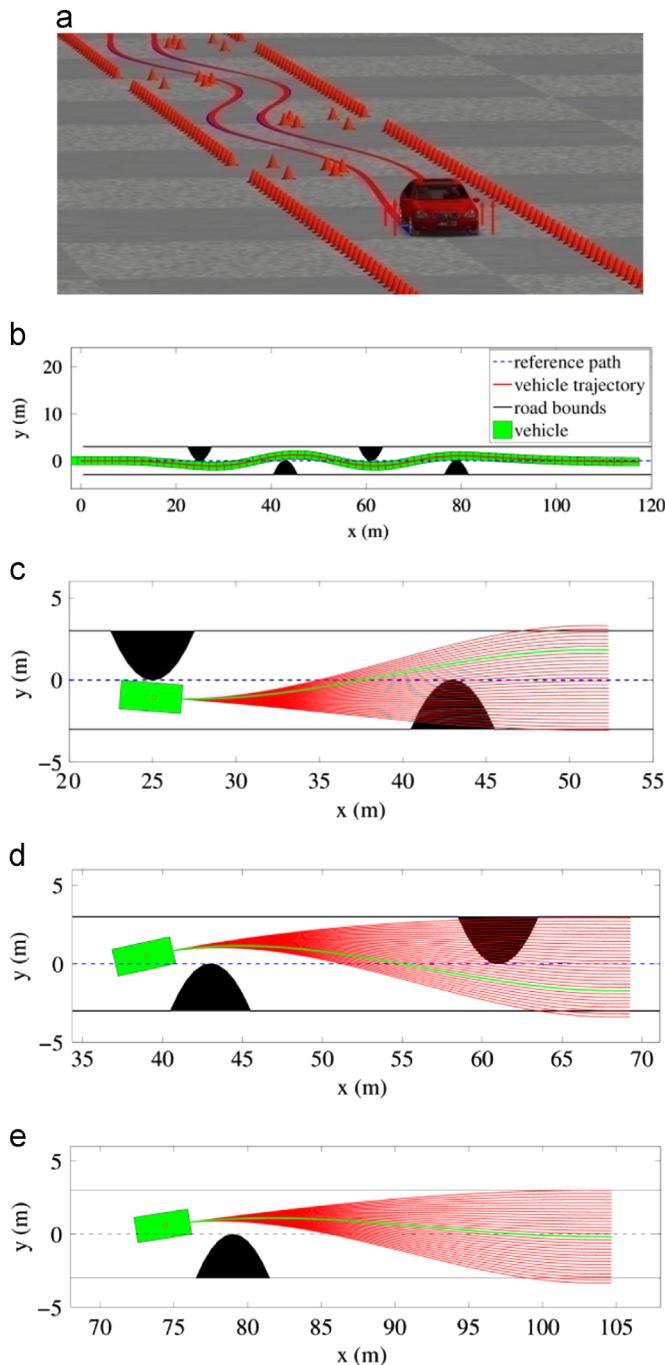


Fig. 15. Path following with static obstacles avoidance: (a) a snapshot from Carsim simulation environment; (b) the vehicle tracking result; (c–e) snapshots for local trajectory generation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3. Simulation results and discussions

To verify effectiveness of the proposed ILTPTC framework and algorithms, firstly we implement it in a simulation environment coupling Matlab with the validated high-fidelity full-vehicle dynamics constructed in Carsim®. The main parameters of a full-size Ackerman-steered D-Class Sedan are listed in Table 1.

The trajectory planning cycle is 100 ms and the low-level tracking control cycle is 20 ms. At the end of each planning cycle, the optimal trajectory is generated and tracked by the low-level controller. The maximal look-ahead distance is 60 m. The responses of the low-level actuators' dynamics are considered as a first-order delay process. The steering control inputs are restricted within the range of $[-30^\circ, 30^\circ]$.

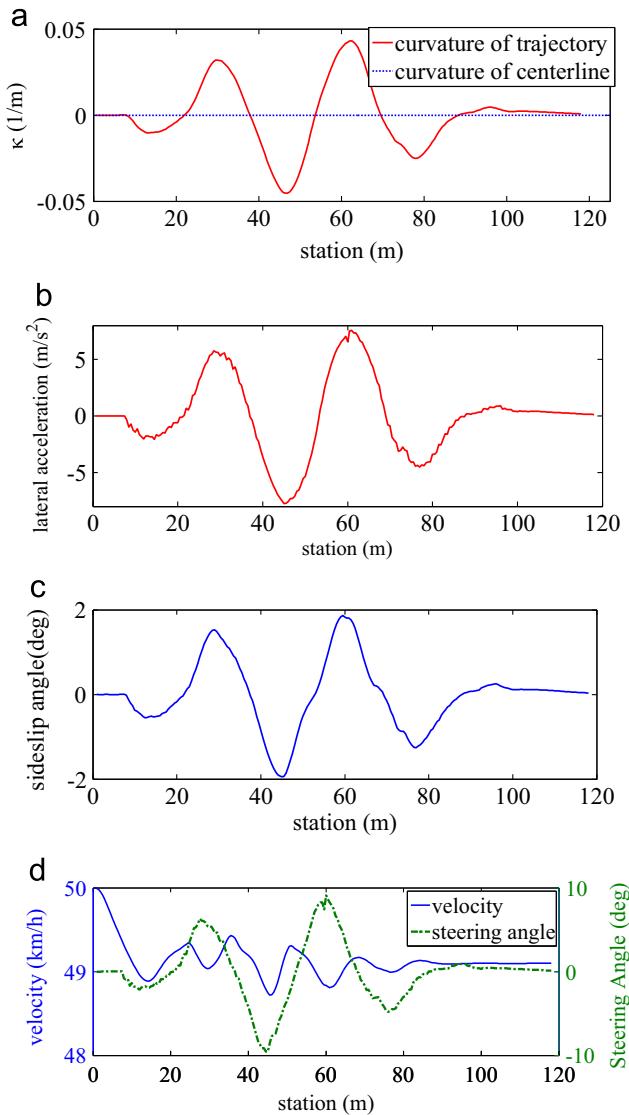


Fig. 16. Trajectory tracking results: (a) the curvature profiles of the vehicle trajectory and the reference path; (b) the lateral acceleration profile; (c) the sideslip angle of C.G.; (d) the velocity and steering angle profiles.

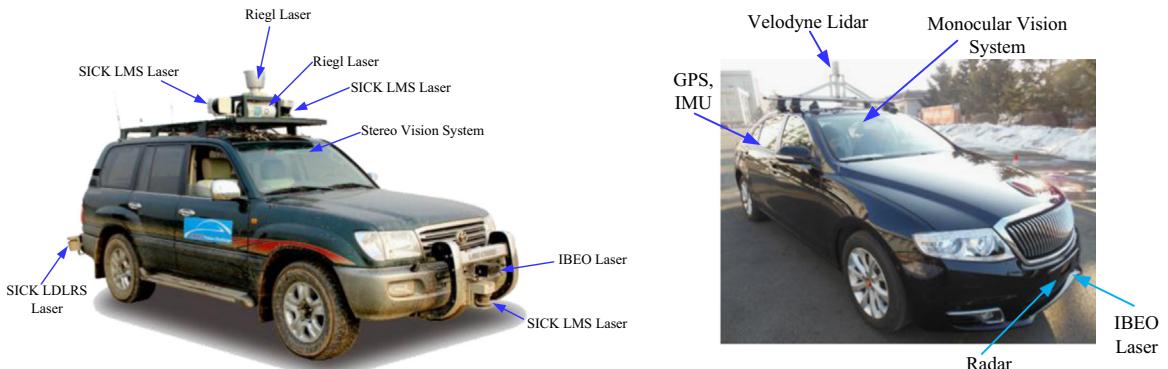


Fig. 17. Our autonomous vehicle test platform.

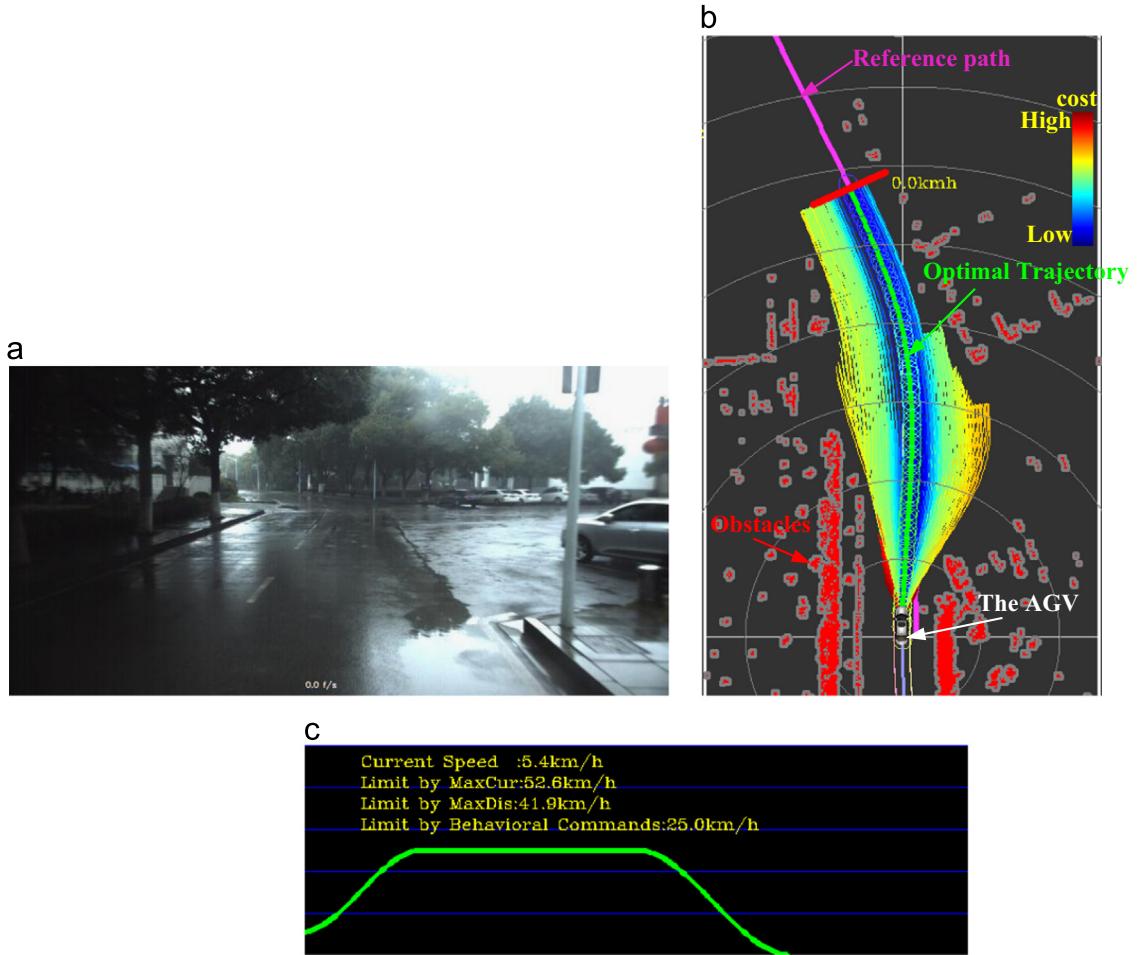


Fig. 18. A snapshot from the local trajectory generation scenario during the field experiment. (a) The image taken by the front camera of the AGV. (b) A snapshot of the local trajectory planning result. (c) the speed profile. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3.1. Curvy path following responses

As shown in Fig. 13(a), a closed-loop road course with tight curves is created to test the capability of the proposed planning and control strategy to negotiate the path with a variety of curvy turns. The road friction coefficient is 0.85 and the width of the road course is 4 m. The control purpose is to steer the vehicle to track the road centerline (reference path) closely, while avoiding colliding with road bounds. The initial C.G. position of the vehicle is ($x=0, y=0$) and the initial speed is 30 km/h. The maximal allowed longitudinal speed is set to be 35 km/h and the maximal lateral acceleration is set to be 5 m/s^2 .

Fig. 13(b) illustrates the vehicle's trajectory (the C.G. is employed as the vehicle reference point) during the entire tracking procedure. It can be seen that the vehicle is able to excellently follow the reference path. Fig. 13(c) describes three snapshots of the trajectory generation and evaluation during the path following process. For clarity, the vehicle states are represented as a function of the path length, which denotes the traveled arc length from the start point. As Fig. 14(a) shows, the curvature of the road centerline (the dash blue curve) is discontinuous and varies sharply at some points. The curvature of the vehicle trajectory is adaptively changing with the curvature of the reference path. From Fig. 14(b), it can be seen that the lateral acceleration is able to be controlled within 5 m/s^2 , it slightly violates the maximal lateral acceleration constraints due to sharp change of the curvature of the reference path at the position ($x=55.6$ m, $y=2.1$ m). As illustrated in Fig. 14(c), the peak value of sideslip angle is more than 5°, which indicates the lateral force of the vehicle tire slightly enters into the nonlinear zone, and the vehicle stability is still able to be guaranteed.

The steering angle and longitudinal vehicle profiles are shown in Fig. 14(d) separately. It can be seen that the velocity can be adaptively tuned according to the curvature of the road. The vehicle decelerates before approaching a turn, and accelerates again during leaving the turn. In addition, it is also subject to the maximal lateral acceleration constraint 5 m/s^2 .

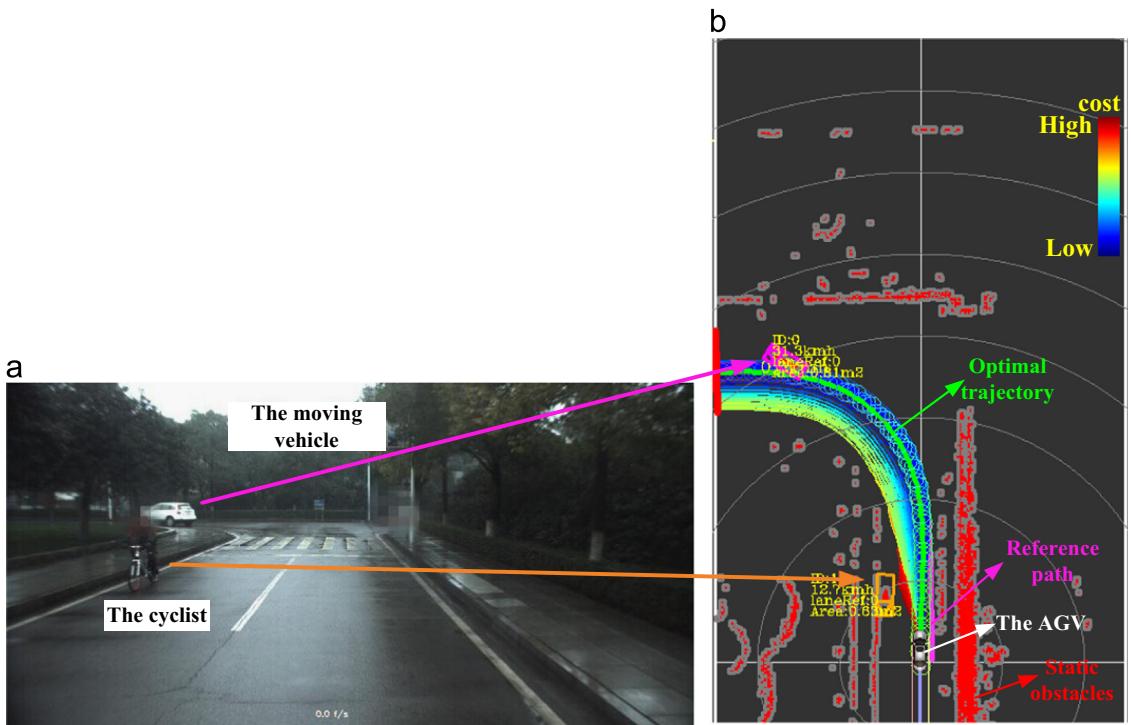


Fig. 19. A snapshot of the local trajectory generation scenario when the AGV negotiates a tight turn while interacting with both static and dynamic obstacles. (a) The image taken by the front camera of the AGV. (b) A snapshot of the local trajectory planning result.

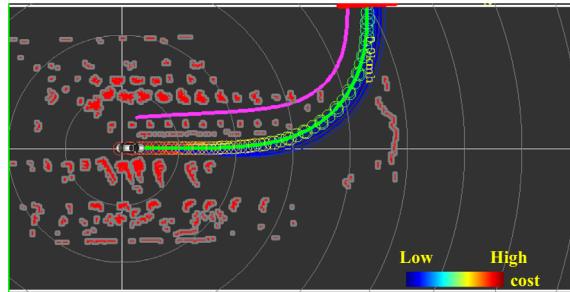


Fig. 20. A snapshot of the local trajectory generation scenario when the reference path deviate from the free zone due to poor GPS signals.

3.2. Reference path following with obstacle avoidance

In order to prove capabilities of the proposed algorithm to follow the reference path while avoiding obstacles, a road track course with unexpected obstacles is established, shown in Fig. 15(a). The road width is 6 m and the road friction coefficient is 1. The road centerline (blue dash line) is the reference path. Along the road there are obstacles, which cannot be known until they are within the sensing range. The initial speed is 50 km/h and the maximal allowed speed is set to be 50 km/h as well. In order to make better use of tire lateral forces to reactively avoid unexpected obstacles, the maximal lateral acceleration is set to be 7 m/s².

For the sake of safety, we require the vehicle to drive within the corridor without colliding with two bounds. It can be seen from Fig. 15(b), the reference path is closed to the road bounds at some points. Therefore, following the centerline without considering the obstacles will inevitably cause collisions.

The path following results in Fig. 15(b) illustrate that the vehicle is capable of keeping itself within the corridor while avoiding colliding with the unexpected obstacles. Fig. 15(c–e) depict three snapshots of local trajectory generation and evaluation scenarios. Based upon the proposed trajectory generation strategy, the vehicle is able to avoid the unexpected obstacles and obtain a collision-free and relatively smooth path to follow.

It can be seen from Fig. 16(a), though the curvature is kept at zero along the entire reference path, the vehicle is able to generate steering control inputs to avoid unexpected obstacles. Fig. 16(b) and (c) shows the vehicle lateral acceleration profile and sideslip angle of the C.G. respectively. Due to the high driving speed and reactive steering actions, the peak value of the lateral acceleration almost reaches the 7.7 m/s². However, the lateral control stability was still guaranteed. The

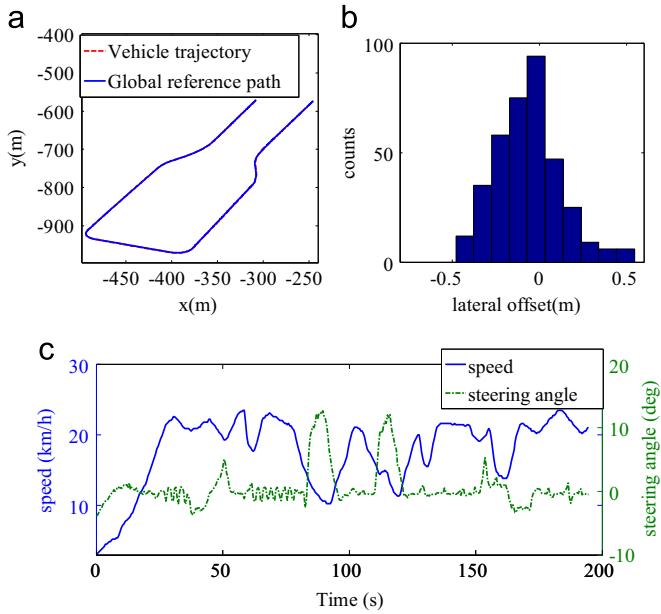


Fig. 21. Vehicle tracking results: (a) the closed-loop test field; (b) the histogram of lateral offset; (c) the speed and front-wheel steering angle curves.

velocity and steering control profiles are shown in Fig. 16(d). Since the trajectory generation algorithm is capable of producing expressive, smooth and long-term trajectories in each control cycle, the sudden braking actions can be effectively avoided during the path following process.

4. Field testing and verifications

After the algorithms have been verified in the simulation environment, we conduct field tests on our autonomous vehicle research platforms to verify the proposed ILPTC strategy further as shown in Fig. 17. The steering, brake, throttle and gear shift actuators of the vehicle have been fully modified to be drive-by-wire. All of these actuators could be controlled via the electronic signals through the CAN Bus. The appropriate sensors, like Lidars, cameras and radar, have been mounted to sense the 360° and 80 m far away of the surrounding environment with 20 cm by 20 cm grid resolution. The planning cycle is 100 ms and the control cycle is 50 ms.

Several typical scenarios are chosen to illustrate capabilities of the proposed trajectory generation approach to run in real time. As shown in Fig. 18, the top image is collected from the camera. The middle image shows the corresponding local planning map. The occupant grid map is used to represent the environment. The grid with red color are static obstacles. The purple curve indicates the reference path obtained from the global planner. A rich set of generated dynamically-feasible trajectory candidates are aligned with the reference path, while avoiding obstacles. The maximal path length is limited to be 60 m for low-speed driving purpose. The color of the trajectories represents the cost value evaluated by the objective function (18). The green trajectory is the optimal trajectory, which is tracked by the low-level tracking controller. The bottom image illustrates the velocity planning profile, which is generated via an enhanced trapezoidal profile. The maximal speed is constrained by the path length, maximal allowed longitudinal and lateral acceleration, as well as commands from behavioral planner. The final speed is set to be zero to always ensure that the vehicle is able to stop safely.

Two snapshots of the local trajectory planner are shown in Figs. 19 and 20 during the autonomous driving along the reference path. It can be seen from Fig. 19, the local path planner is able to generate collision-free trajectory for the vehicle negotiating a tight turn, while safely interacting with the bicyclist and the car in front. As Fig. 20 illustrates, the reference path obtained from the global planner deviates from the road due to GPS positioning error, the local planner remains to produce safe and drivable trajectories to follow the direction of the reference path.

We have tested the proposed ILPTC algorithms on our autonomous vehicle platform for hundreds of kilometers with various road and weather conditions. Fig. 21 depicts one of the experimental results in a closed-loop test field. It can be seen from Fig. 21(b), the lateral offset is less than the 0.5 m during the whole tracking course. The maximal allowed speed is limited to be 25 km/h. As illustrated in Fig. 21(c), the vehicle decelerates when it approaches a curvy turn and accelerates again during leaving the turn.

5. Conclusions and future work

This study has developed and evaluated a new integrated local trajectory planning and tracking control (ILTPC) strategy for autonomous ground vehicles driving along a reference path with obstacles avoidance. The efficient trajectory generation scheme provides AGVs abilities to smoothly follow the reference path and promptly react to the unexpected obstacles as well. The desired speed and yaw-rate profiles can be easily derived from the generated trajectory, then they can be employed to significantly reduce control efforts. The presented combined feedback and feedback tracking controller is capable of tracking the desired commands while rejecting the model uncertainties and external disturbances.

Although the paper has focused on handling static obstacles avoidance in the experiments conducted, the proposed planning strategy could also be extended to handle dynamic obstacles in urban environments. The structured road information, the motion predictions of other dynamic objects (such as vehicles, cyclists and pedestrians) as well as spatio-temporal collision tests could be incorporated into the trajectory generation and evaluation framework. Therefore, the extension of the proposed motion planning strategies will be continuously studied in the future. To further enhance motion prediction accuracy and improve control performance, the time-varying parameters identification of the vehicle dynamic model will be carefully incorporated into both trajectory planning and tracking control procedures. In addition, the motion planning approach will also be extended to handle more complex and realistic traffic scenarios.

References

- [1] R. Wang, C. Hu, Z. Wang, F. Yan, N. Chen, Integrated optimal dynamics control of 4WD4WS electric ground vehicle with tire-road frictional coefficient estimation, *Mech. Syst. Signal Process.* 60–61 (2015) 727–741.
- [2] Y. Luo, T. Chen, K. Li, Multi-objective decoupling algorithm for active distance control of intelligent hybrid electric vehicle, *Mech. Syst. Signal Process.* 64–65 (2015) 29–45.
- [3] X. Zhu, H. Zhang, Z. Fang, Speed synchronization control for integrated automotive motor–transmission powertrain system with random delays, *Mech. Syst. Signal Process.* 64–65 (2015) 46–57.
- [4] A. Pazooki, S. Rakheja, D. Cao, Modeling and validation of off-road vehicle ride dynamics, *Mech. Syst. Signal Process.* 28 (2012) 679–695.
- [5] C. Urmson, J. Anhalt, D. Bagnell, et al., Autonomous driving in urban environments: boss and the urban challenge, *J. Field Robot.* 25 (8) (2008) 425–466.
- [6] M. Montemerlo, J. Becker, S. Bhat, et al., Junior: the stanford entry in the urban challenge, *J. Field Robot.* 25 (9) (2008) 569–597.
- [7] T.M. Howard, A. Kelly, Optimal rough terrain trajectory generation for wheeled mobile robots, *Int. J. Robot. Res.* 26 (2) (2007) 141–166.
- [8] J. Wang, M.F. Hsieh, Vehicle yaw inertia and mass independent adaptive control for stability and trajectory tracking enhancements, in: Proceedings of American Control Conference (ACC), 2009, pp. 689–694.
- [9] J. Wang, J. Steiber, B. Surampudi, Autonomous ground vehicle control system for high-speed and safe operation, *Int. J. Veh. Auton. Syst.* 7 (1) (2009) 18–35.
- [10] T. Gu, J. Snider, J.M. Dolan, J.-W. Lee, Focused Trajectory Planning for autonomous on-road driving, in: Proceedings of IEEE Intelligent Vehicles Symposium (IV), 2013, pp. 547–552.
- [11] T. Gu, J.M. Dolan, On-road motion planning for autonomous vehicles, *Intelligent Robotics and Applications*, Springer, Berlin, Heidelberg, 2012, 588–597.
- [12] H. Zhang, X. Zhang, J. Wang, Robust gain-scheduling energy-to-peak control of vehicle lateral dynamics stabilisation, *Veh. Syst. Dyn.* 52 (3) (2014) 309–340.
- [13] R. Wang, H. Zhang, J. Wang, Linear parameter-varying controller design for four-wheel independently actuated electric ground vehicles with active steering systems, *IEEE Trans. Control Syst. Technol.* 22 (4) (2014) 1281–1296.
- [14] X. Huang, H. Zhang, G. Zhang, J. Wang, Robust weighted gain-scheduling vehicle lateral motion control with considerations of steering system backlash-type hysteresis, *IEEE Trans. Control Syst. Technol.* 22 (5) (2014) 1740–1753.
- [15] T.M. Howard, C.J. Green, A. Kelly, D. Ferguson, State space sampling of feasible motions for high-performance mobile robot navigation in complex environments, *J. Field Robot.* 25 (6–7) (2008) 325–345.
- [16] M. McNaughton, C. Urmson, J.M. Dolan, J.-W. Lee, Motion planning for autonomous driving with a conformal spatiotemporal lattice, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 4889–4895.
- [17] M. Pivtoraiko, R.A. Knepper, A. Kelly, Differentially constrained mobile robot motion planning in state lattices, *J. Field Robot.* 26 (3) (2009) 308–333.
- [18] A. Kelly, B. Nagy, Reactive nonholonomic trajectory generation via parametric optimal control, *Int. J. Robot. Res.* 22 (7–8) (2003) 583–601.
- [19] T. Gu, J.M. Dolan, Toward Human-like Motion Planning in Urban Environment, in: Proceedings of IEEE Intelligent Vehicles Symposium (IV), 2014, pp. 350–355.
- [20] M. Werling, S. Kammel, J. Ziegler, L. Gröll, Optimal trajectories for time-critical street scenarios using discretized terminal manifolds, *Int. J. Robot. Res.* 31 (3) (2012) 346–359.
- [21] L. Hu, Z. Gu, J. Huang, Y. Yang, X. Song, Research and realization of optimum route planning in vehicle navigation systems based on a hybrid genetic algorithm, *Proc. Inst. Mech. Eng. Part D: J. Automob. Eng.* 222 (5) (2008) 757–763.
- [22] J. Ryu, J.C. Gerdes, Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control, *J. Dyn. Syst. Meas. Control* 126 (2) (2004) 243–254.
- [23] G. Bevan, H. Gollee, J. O'Reilly, Trajectory generation for road vehicle obstacle avoidance using convex optimization, *Proc. Inst. Mech. Eng. Part D: J. Automob. Eng.* 224 (4) (2010) 455–473.
- [24] E. Kayacan, E. Kayacan, H. Ramon, et al., Distributed nonlinear model predictive control of an autonomous tractor–trailer system, *Mechatronics* 24 (8) (2014) 926–933.
- [25] A. Gray, M. Ali, Y. Gao, J.K. Hedrick, F. Borrelli, A unified approach to threat assessment and control for automotive active safety, *IEEE Trans. Intell. Transp. Syst.* 14 (3) (2013) 1490–1499.
- [26] T. Shim, G. Adiredy, H. Yuan, Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control, *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* 226 (6) (2012) 767–778.
- [27] C.E. Beal, J. Gerdes, Model predictive control for vehicle stabilization at the limits of handling, *IEEE Trans. Control Syst. Technol.* 21 (4) (2013) 1258–1269.
- [28] K. Macek, R. Philippsen, R.Y. Siegwart, Path following for autonomous vehicle navigation based on kinodynamic control, *J. Comput. Inf. Technol.* 17 (1) (2009) 17–26.
- [29] U. Schweißinger, M. Rufli, P. Furgale, et al., A sampling-based partial motion planning framework for system-compliant navigation along a reference path, in: Proceedings of IEEE Intelligent Vehicles Symposium(IV), 2013, pp. 391–396.
- [30] K. Yang, S. Sukkarieh, An analytical continuous-curvature path-smoothing algorithm, *IEEE Trans. Robot.* 26 (3) (2010) 561–568.

- [31] T. Gu, J.M. Dolan, J.-W. Lee, On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability, in: Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), 2014.
- [32] J. Ziegler, P. Bender, T. Dang, et al., Trajectory Planning for Bertha – a local, continuous method, in: Proceedings of IEEE Intelligent Vehicle Symposium (IV), 2014, pp. 450–457.
- [33] R. Lenain, B. Thuilot, C. Cariou, P. Martinet, Mixed kinematic and dynamic sideslip angle observer for accurate control of fast off-road mobile robots, *J. Field Robot.* 27 (2) (2010) 181–196.
- [34] J. Wang, Z. Sun, X. Xu, et al., Adaptive speed tracking control for autonomous land vehicles in all-terrain navigation: an experimental study, *J. Field Robot.* 30 (1) (2013) 102–128.
- [35] Q. Chen, Z. Sun, D. Liu, et al., Ribbon model based path tracking method for autonomous ground vehicles, *J. Central South Univ.* 21 (5) (2014) 1816–1826.
- [36] R. Rajamani, *Vehicle Dynamics and Control*, Springer, US, 2011.
- [37] H. Zhang, Y. Shi, J. Wang, On energy-to-peak filtering for nonuniformly sampled nonlinear systems: a markovian jump system approach, *IEEE Trans. Fuzzy Syst.* 22 (1) (2014) 212–222.