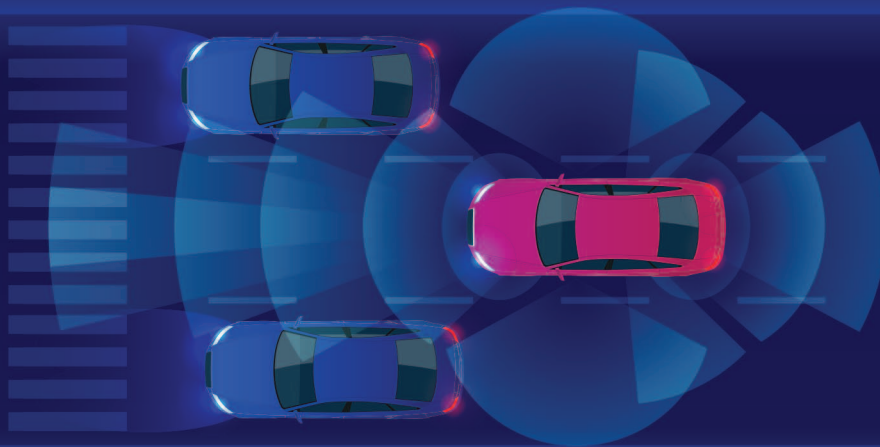


Trajectory Planning and Tracking for Autonomous Vehicle Based on State Lattice and Model Predictive Control



©ISTOCKPHOTO.COM/OSTAPENKOOLENA

Abstract—Trajectory planning and tracking control are two keys of collision avoidance for autonomous vehicles in critical traffic scenarios. It requires not only the system functionality, but also strong real-time. In this paper, we integrated trajectory planner and tracking controller for autonomous vehicle to implement trace planning and tracking for obstacle avoidance. The trajectory planner is based on the state lattice approach and the tracking controller is designed based on the model predictive control

Chaoyong Zhang, Duanfeng Chu, Shidong Liu, Zejian Deng, and Chaozhong Wu
Intelligent Transportation Systems Research Center, Wuhan University of Technology.
Correspondence: chudf@whut.edu.cn.

Xiaocong Su
Wuhan Kotei Technology Corporation.
E-mail: xiaocongs@kotei.com.cn

Digital Object Identifier 10.1109/MITS.2019.2903536
Date of publication: 18 March 2019

using the vehicle kinematics model. The simulation shows that the planner can generate smooth trajectories which could be selected as references for the controller. The maximum tracking error is less than 0.2 m when the vehicle speed is below 50 km/h. Additionally, the on field test shows that the test vehicle with this method is capable of following the reference path accurately, even at sharp corners.

I. Introduction

With the rapid development of sensor and control technologies, advanced driver assistance systems (ADAS) become more and more powerful [1], [2]. Further, the highly autonomous vehicle technology, i.e., SAE L4, and L5 attracts more attention from the academia and industry for improving the driving safety, efficiency and comfort.

However, the deficiency of the technology leads to the half-baked automated vehicle. There are still various challenges in achieving the fully autonomous vehicle for the masses of the public. As is well known, the key technologies of automated driving include environment perception, decision-making, trajectory planning, motion control, car networking, human-vehicle-interaction, etc. [3].

Among these key technologies, the vehicle's trajectory planning and motion control are two of the most significant ones. The trajectory planner generates viable trajectory based on the road environment in time, then the motion controller tracks the reference accurately. Depending on the complexity of the trajectory planner, these two modules can be considered as a combined block or independent from each other [4]. As the core module, both of them play an important role in driving safety and comfort.

Vehicle driving is a complex non-linear system considering multiple constraints. In order to simulate the real driving process, the literature has built a large amount of driver models. The main methods include various improved PID control, robust control, model predictive control (MPC), etc.

There are a large amount of literature, which shows the success in the vehicle longitudinal control by using PID control. Adaptive Cruise Control (ACC) [5] and Lane Keeping Control (LKC) can achieve good performance using PID controllers [6], [7]. However, the vehicle lateral control is a comparative complex problem, which needs more robustness to ensure safety and comfort. The parameters of PID controller are very difficult to determine because handling systems with complex nonlinearities that requires high accuracy [8]. Over the state of the art in autonomous driving technology, the method of MPC has

been shown as a promising method to achieve good control performance [9].

The MPC method utilizes the vehicle model to predict the vehicle's future motion states by combining current sampled states and target states generated by the path planner. In each period, the MPC controller generates a control action sequence by minimizing the objective function considering the control constraints. The first control action of the sequence is the input variable for the vehicle low-level controller. As time goes by, the vehicle's current motion states and the target states are both changing. Hence, the next action of the sequence are not the most satisfying for the optimization requirements. So the same progress will be repeated at subsequent time steps. These iteration steps will eventually lead to the smallest error and the best performance.

The basic function of the MPC controller is to track the expected states to make the vehicle reach the destination with guaranteeing safety and comfort [10]. It is better to improve the energy efficiency at the same time [11]. An autonomous vehicle based on MPC with the function of obstacle avoidance was presented [12]. Further, it can be equipped with other advanced functions, i.e., the planning and control strategies both reflect the driver behavior. In other words, different trajectories and control strategies can be customized for different drivers in autonomous vehicles. A MPC controller was presented to reflect different characteristics of the driver [8]. The MPC-based driver model could reflect different drivers' skills through considering the stochastic characteristics of drivers' steering features [13].

In the literature about the controller of the autonomous vehicle, there is a common assumption that the reference path tracked is known prior and completely which can be considered as a global path [14], [15]. However, the reference path without optimization is rough and makes the vehicle movement bumpy. Although there are several methods for finding optimized smooth path from the origin to the destination [16], [17]. However, the driver cannot predict the next moment trace unless the vehicle completely track the reference path. For example, if an obstacle is on the reference path, in order to avoid the obstacle, the vehicle cannot completely track the reference path. At the meantime, the driver does not predict the starting time and the direction that the vehicle moves around the obstacle. These uncontrollable factors may lead to driver's anxiety and potential unsafe issues. Besides, the traffic flow is also dynamic and the surroundings are changing all the time. Therefore, it is necessary to plan and optimize the reference trajectory in real time during the vehicle's movement [18], [19].

The purpose of trajectory planning is to generate a trajectory between current and desired points that is as smooth as possible to avoid obstructions [20]. Path

planning and trajectory planning already had a great deal of successful experience in the field of mobile robots and unmanned aerial vehicle [17], [21].

In the process of autonomous vehicle driving, the basic idea is to drive through a series of discrete points between the start and the end. Each point represents a grid area. These grids can construct a complete driving scene. The vehicle eventually reaches the goal point by passing through the series of adjacent grids, which are obtained by the heuristic search algorithm.

Dijkstra algorithm [22] and its extension, i.e., A-star algorithm (A*) are the most common path searching algorithms. In 2007 DARPA Urban Challenge (DUC), many teams used the A* algorithm to implement automated driving [23], [24]. Except for simplicity, another advantage of the algorithm is the strong expandability. The winner of DUC also used the improved algorithm based on the A* algorithm [25].

The first step in using the A* algorithm is to construct the environment map with many uniform grids. For the straight lanes, low-density or low speed, the A* algorithm and its improved algorithm are feasible. However, when it comes to complex curved road, the uniform grid cannot satisfy the demand. To deal with large environment, it needs vast memory usage to construct the map. Besides, for the real automated vehicle, it is hard to devise a cost function to reflect the energy efficiency, comfort, vehicle physics constraints, etc.

The state lattice algorithm is an improved graph search algorithm that can solve these problems. The algorithm uses a hyper-dimensional grid of states to represent the planning area [26]. The major advantage of the state lattice algorithm is that multiple target endpoints could be selected as end points. In this way, it generates multiple reference trajectories and then choose the best one according to the established cost function. In [27], by choosing the best trajectory from the multiple reference paths, it made autonomous vehicle driving performs more superior.

The cost function could be divided into static cost and dynamic cost [28]. The static cost was related to obstacle and the motion state of the vehicle, while the dynamic cost considered the velocity and acceleration. Literature pointed out that the future driving progress should completely independent to the past state, so the driving process could be seen as Markov process, which could be dealt with the Belleman's Principle to find the optimal actions [29].

The trajectory generation of the state lattice algorithm uses curvature polynomials with parameters to get the trajectory between two state points, so that the vehicle

The MPC method utilizes the vehicle model to predict the vehicle's future motion states by combining current sampled states and target states generated by the path planner. In each period, the MPC controller generates a control action sequence by minimizing the objective function taking the control constraints into account.

movement can be carried out with the constraints about the kinematic and road shape. In [30], it proposed that curvature polynomials could be arbitrary order. Using quintic polynomials could lead to a more smooth vehicle motion at high speed by comparing the cubic polynomials and quintic polynomials [28].

However, the trajectory generation method could only be applied in highway scenes [31]. Besides, in the field of trajectory planning, it is generally assumed that the tracking controller is perfect enough to track any trajectory [29]. Meanwhile in the trace tracking control area, the reference path is always a global path, which is not a real-time trajectory. The purpose of this paper is to combine the individual parts and consider the performance as a whole.

Based on these circumstances, this paper establishes a framework of integrating local trajectory planner and tracking controller for autonomous vehicle. According to the road information, the trajectory planner based on state lattice generates reference trajectory in real time for the tracking controller based on model predictive control. The tracking controller uses the reference trajectory for feedback control. The trace tracking error and computational efficiency are both better than the previous studies [10]. The trajectory planner is optimized to be used on medium speed and low speed urban roads in addition to applications on highways.

The rest of the paper is organized as follows: in Section II, the method of trajectory planning and the trace-tracking algorithm are both derived in detail. Section III presents the simulation and the on-field test for unmanned ground vehicle. It is followed by result summarizations and the future improvements.

II. Method

A. Design of the Controller

1) Vehicle Model

On urban roads, the non-holonomic vehicle kinematic model can be adopted to achieve the autonomous motion effectively [52].

This paper establishes a framework of integrating local trajectory planner and tracking controller for autonomous vehicle.

The vehicle kinematics model is given by [33]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi \\ \sin \varphi \\ \frac{\tan \delta}{l} \end{bmatrix} \times v \text{ or } \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (1)$$

Where $\mathbf{x} = [x \ y \ \varphi]^T$ represents the vehicle position states in the global inertial coordinate system; $\mathbf{u} = [v \ \delta]$ represents the controlled variables which are the outputs of the MPC controller. Vector $[x \ y]^T$ represents the coordinate whose origin located at the center of the vehicle's rear shaft; φ is the vehicle heading angle; v is the longitudinal speed; δ is the front wheel steering angle; and l is the axle distance of the vehicle.

In order to obtain the vehicle system control outputs, it needs feedback compensation by comparing the sampled vehicle state with the state of a reference vehicle, which is assumed to be driving on the reference path. The trajectory planner plans the reference vehicle state. It can be defined as the following form

$$\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{u}_r) \quad (2)$$

Calculating the error of the sampled vehicle state through the Taylor series around the reference point $(\mathbf{x}_r, \mathbf{u}_r)$, then discretizing the system into linear time invariant formulation [34]

$$\begin{cases} \tilde{\mathbf{x}}(k+1) = \mathbf{A}_{k,t} \tilde{\mathbf{x}}(k) + \mathbf{B}_{k,t} \tilde{\mathbf{u}}(k) \\ \mathbf{y}(k) = \mathbf{C}_{k,t} \mathbf{x}(k) \end{cases} \quad (3)$$

With

$$\mathbf{A}_{k,t} = \begin{bmatrix} 1 & 0 & -v_r(k) \sin \varphi_r(k) T \\ 0 & 1 & v_r(k) \cos \varphi_r(k) T \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}_{k,t} = \begin{bmatrix} \cos \varphi_r(k) T & 0 \\ \sin \varphi_r(k) T & 0 \\ \frac{\tan \delta_r(k) T}{l} & \frac{v_r(k) T}{l \cos^2 \delta_r(k)} \end{bmatrix}$$

$$\mathbf{C}_{k,t} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where T is the sampling period, t is the discrete time index, k means the step ahead from t .

2) MPC Controller

Generally, the MPC controller selects the system output variable by optimizing an objective function with constraints. Its solution can make the vehicle move safely and

fast by minimizing the objective function. The most common objective function is defined as a quadratic function of the states and control inputs like [34].

$$J(k) = \sum_{i=1}^{N_p} \tilde{\mathbf{y}}^T(k+i|k) \cdot \mathbf{Q} \cdot \tilde{\mathbf{y}}(k+i|k) + \sum_{i=1}^{N_c} \tilde{\mathbf{u}}^T(k+j-1|k) \cdot \mathbf{R} \cdot \tilde{\mathbf{u}}(k+j-1|k) \quad (4)$$

Where N_p is the prediction horizon and N_c is the control horizon. In general, $N_p > N_c$. \mathbf{Q} and \mathbf{R} are weighting matrices. For each parameters, the corresponding coefficients in the weighting matrices may be different. It is of significant importance to adjust them to obtain desired control performance with minimum error or parallel to the route.

In reality, the vehicle speed control is implemented by operating the throttle and the brake, as well as the steering wheel for tracking its driving direction. Both of them do not use $\mathbf{u} = [v \ \delta]^T$ as the input variable directly but the increment of \mathbf{u} , i.e., $\Delta \mathbf{u}$.

For that purpose, the discrete-time system should be transformed as follows.

$$\begin{cases} \tilde{\mathbf{X}}(k+1|t) = \tilde{\mathbf{A}}_{k,t} \tilde{\mathbf{X}}(k) + \tilde{\mathbf{B}}_{k,t} \Delta \tilde{\mathbf{u}}(k|t) \\ \mathbf{Y}(k) = \tilde{\mathbf{C}}_{k,t} \tilde{\mathbf{X}}(k|t) \end{cases} \quad (5)$$

Where

$$\mathbf{X}(k|t) = \begin{bmatrix} \tilde{\mathbf{X}}(k|t) \\ \tilde{\mathbf{u}}(k-1|t) \end{bmatrix}$$

$$\tilde{\mathbf{A}}_{k,t} = \begin{bmatrix} \mathbf{A}_{k,t} & \mathbf{B}_{k,t} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\tilde{\mathbf{B}}_{k,t} = \begin{bmatrix} \mathbf{B}_{k,t} \\ \mathbf{I} \end{bmatrix}$$

$$\tilde{\mathbf{C}}_{k,t} = [\mathbf{C}_{k,t} \ \mathbf{0}]$$

Referring to [35], the modified objective function of Equation(4) should be changed accordingly:

$$J(k) = \sum_{i=1}^{N_p} \tilde{\mathbf{Y}}^T(k+i|k) \cdot \mathbf{Q} \cdot \tilde{\mathbf{Y}}(k+i|k) + \sum_{i=1}^{N_c} \Delta \tilde{\mathbf{u}}^T(k+j-1|k) \cdot \mathbf{R} \cdot \Delta \tilde{\mathbf{u}}(k+j-1|k) + \rho \cdot \varepsilon^2 \quad (6)$$

Where ε is the slack variable. In this objective function, the first term indicates the capacity of tracking the reference path and reflects the driving safety. The second term emphasizes control outputs smoothness, which reflects the driving comfort. The last term is not endowed with physical meaning of reality, but it could prevent null solution of the optimization [36]. In some cases, hard constrained MPC problem lead to no feasible actions can be computed. By using the slack variables, the hard constraint MPC problem is turned into soft constraint MPC problem, then the feasibility of the vehicle system could be recovered [35].

The modified objective function incorporates the increment of the control variate directly. It also includes the following constraints related to the control outputs and their increments.

$$u_{\min} \leq u \leq u_{\max}$$

$$\Delta u_{\min} \leq \Delta u \leq \Delta u_{\max}$$

In order to achieve good tracking performance, the constraints in this paper are the most basic. In its implementation, the key is to look for the most suitable reference position in the prediction horizon.

B. Trajectory Planner

Drivers are accustomed to change lanes based on lane lines rather than the shortest distance for saving time or energy [37]. In addition, due to the diversity of the road alignment, it is very troublesome to make the road discretization by only adopting the inertial X-Y coordinate. Therefore, we adopt S-L coordinate system to take road shapes into consideration [38].

As shown in Fig. 1, we define a point (s, l) , where, s represents the arc length along the center line, which is also called station, and l represents the lateral offset of the center line, which is also called latitude. Like the X-Y coordinate system, we can define the following kinematic vector γ about arc length s and latitude l .

$$\gamma = [x(s, l) \ y(s, l) \ \theta(s, l) \ k(s, l)] \quad (7)$$

Where, $\theta(s, l)$ is the heading angle, $k(s, l)$ is the curvature of the path.

Note that the vector γ is used to describe the state of the sampling point. In highway scenarios, the vehicle is assumed to be parallel to the center line so that the vehicle's heading angle and the curvature are the same in term of the road shape. However, the purpose here is to make the

In order to achieve good tracking performance, the constraints in this paper are the most basic. In its implementation, the key is to look for the most suitable reference position in the prediction horizon.

vehicle run at any speed. Thus, the initial sampling state is depending on the vehicle's initial state. Likewise, the endpoint status depends on the desired vehicle state.

The vehicle model defined in Equation (1) can be reformed as follows in terms of the S-L coordinate system.

$$\frac{d\theta}{ds} = k(s)$$

$$\frac{dx}{ds} = \cos[\theta(s)]$$

$$\frac{dy}{ds} = \sin[\theta(s)] \quad (8)$$

The Equations (8) shows that the vehicle position state is determined by the curvature. Thus for the vehicle trajectory planner curvature can be the only controlled variable. In other words, the local trajectory can be carried out by determining the curvature.

In [27], a curve was described by mixing cubic and quartic polynomials. However, the quadratic polynomials consume more computational resources, and need more time to generate the result. Besides, high-order polynomial curve makes the motion seem exaggerated at low speed situation. Dealing with the compromise and increase computational efficiency, here the curvature is defined as a cubic polynomial function of the arc length.

$$k(s) = a + b \cdot s + c \cdot s^2 + d \cdot s^3 \quad (9)$$

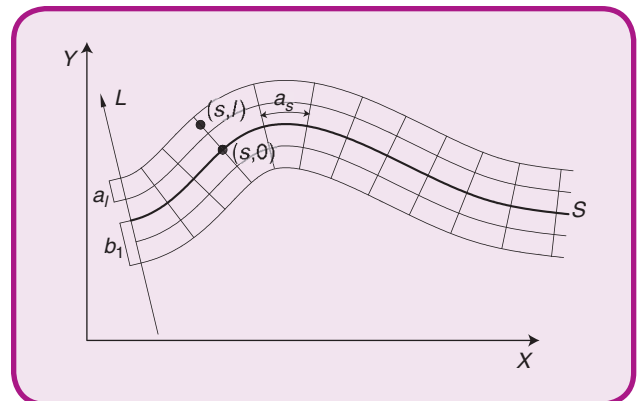


FIG 1 S-L coordinate.

A set of parameters $[a \ b \ c \ d]$ can determine a path. In order to reduce the magnitude error among the parameters [27] and make it easier to adjust the parameters, we use a parameter $\mathbf{p} = [p_0 \ p_1 \ p_2 \ p_3 \ p_4 \ p_5]$ to reformulate the function referring to [28] and [39].

$$k(s) = a(\mathbf{p}) + b(\mathbf{p}) \cdot s + c(\mathbf{p}) \cdot s^2 + d(\mathbf{p}) \cdot s^3 \quad (10)$$

Where

$$\begin{aligned} k(0) &= p_0 \\ k(s_f/3) &= p_1 \\ k(2s_f/3) &= p_2 \\ k(s_f) &= p_3 \end{aligned}$$

When the vehicle drives on the road, the initial position $k(0) = p_0$ and the desired endpoint $k(s_f) = p_3$ are given. So we only need $\mathbf{p} = [p_1 \ p_2 \ p_4]$ to determine the path state of the system.

Since the vehicle drives on the road, the local trajectory planning should start from the current vehicle position, ending to the previewed target point. Therefore, at each planning cycle, the vehicle's current position is defined as the initial sampling point. Then, the arc length of the vehicle's location should be defined as $s = 0$. To create trajectory planner in highway scenarios, it is assumed that the heading angle $\theta = 0$ [28], [31]. However, autonomous vehicle drives not only on highway scenes, but also on complex urban roads with medium or low speeds. Therefore, the heading angle θ cannot be assumed as 0. In each planning cycle, the vehicle can obtain the vehicle's current state through the positioning system. Thus, Equation (8) (10) can be reformed as follows.

$$\begin{aligned} k_p(s) &= a(\mathbf{p}) + b(\mathbf{p}) \cdot s + c(\mathbf{p}) \cdot s^2 + d(\mathbf{p}) \cdot s^3 \\ \theta_p(s) &= a(\mathbf{p}) \cdot s + b(\mathbf{p}) \cdot \frac{s^2}{2} + c(\mathbf{p}) \cdot \frac{s^3}{3} + d(\mathbf{p}) \cdot \frac{s^4}{4} + \theta_p(0) \\ x_p(s) &= \int_0^s \cos[\theta_p(s)] ds + x_p(0) \\ y_p(s) &= \int_0^s \sin[\theta_p(s)] ds + y_p(0) \end{aligned} \quad (11)$$

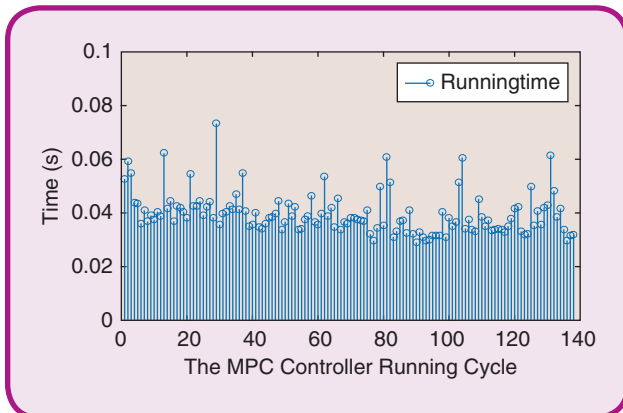


FIG 2 Iteration time.

Where, $\mathbf{x}_p(0) = [x_p(0) \ y_p(0) \ \theta_p(0) \ k_p(0)]$ is the initial sampling point state which can be obtained by the vehicle's positioning system. The endpoint of the trajectory is $\mathbf{x}_p(s_f) = [x_p(s_f) \ y_p(s_f) \ \theta_p(s_f) \ k_p(s_f)]$. We find the parameter \mathbf{p} which can make the endpoint equals or very close to the desired point $\mathbf{x}_{des}(s)$. Because the form of $\theta_p(s)$ and $k_p(s)$ are similar, in order to simplify the computation complexity, we set the endpoint status of the path $\mathbf{x}_p(s_f) = [x_p(s_f) \ y_p(s_f) \ \theta_p(s_f)]$ in the process of looking for the parameter \mathbf{p} .

The gradient decent algorithm can achieve a wonderful effect on the iterative solution problems. It is useful in rapidly minimizing the error between $\mathbf{x}_{des}(s)$ and $\mathbf{x}(s)$. The key of the method is to calculate the Jacobian of the endpoint status vector $\mathbf{x}_p(s_f) = [x_p(s_f) \ y_p(s_f) \ \theta_p(s_f)]$ with respect to the parameter $\mathbf{p} = [p_1 \ p_2 \ p_4]$.

$$\mathbf{J}_p(\mathbf{x}_p(s_f)) = \begin{bmatrix} \frac{dx_p(s_f)}{dp_1} & \frac{dy_p(s_f)}{dp_1} & \frac{d\theta_p(s_f)}{dp_1} \\ \frac{dx_p(s_f)}{dp_2} & \frac{dy_p(s_f)}{dp_2} & \frac{d\theta_p(s_f)}{dp_2} \\ \frac{dx_p(s_f)}{dp_4} & \frac{dy_p(s_f)}{dp_4} & \frac{d\theta_p(s_f)}{dp_4} \end{bmatrix}$$

Before the start of each calculate cycle, an initial value \mathbf{p}_{init} needs be assumed, then calculating the error between the desired state and the estimated state. After that, it can generate a new estimated \mathbf{p}_i with the Jacobian by using Newton's method.

$$\begin{aligned} \Delta \mathbf{x} &= \mathbf{x}_{des} - \mathbf{x}_{p_i} \\ \Delta \mathbf{p} &= \mathbf{J}_p(\mathbf{x}_p(s_f))^{-1} \cdot \Delta \mathbf{x} \\ \mathbf{p}_{i+1} &= \mathbf{p}_i + \Delta \mathbf{p} \end{aligned}$$

In order to improve the efficiency, the iteration process continues until reaching the maximum number of iteration times or Equation (12) is smaller than a value that shows the error is sufficiently small. Because of the different characteristics of each elements in the vector γ , a specific weight should be added to each element when calculating the error.

$$\mathbf{J} = \left\| x_{p_i} - x_{des} \right\|_{Q_1} + \left\| y_{p_i} - y_{des} \right\|_{Q_2} + \left\| k_{p_i} - k_{des} \right\|_{Q_3} + \left\| \theta_{p_i} - \theta_{des} \right\|_{Q_4} \quad (12)$$

The local trajectory can be finally generated as the iteration ends. In addition, such trajectory will be input into the tracking controller as a reference trajectory.

III. Experiment

In order to fully demonstrate the effectiveness of the method provided. Three different kinds of experiments were carried out including fixed trajectory tracking, trajectory generation and on-field test with an autonomous vehicle.

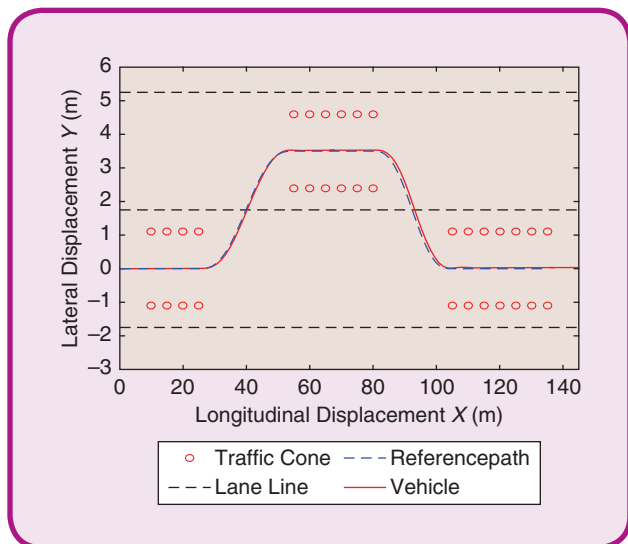


FIG 3 Trace tracking (average speed $v = 50$ km/h).

A. Fix-Path Tracking

The trace tracking simulation was studied via combining MATLAB/Simulink with CarSim on a Windows 10 PC with Intel Core2 Duo CPU.

Comparing with the results in [10], although the CPU performance was worse, the average time per iteration was only 0.040 s, which was less than that in [10]. Moreover, the speed set is higher, but the lateral tracking error is much less. In [10], the errors were out of the safety bounds (error ≤ 0.3 m) in up to eighteen percent of the time. Here, using the same standard, the vehicle is safe in the whole driving process. As shown in Fig. 3, the maximum error is 0.18 m appearing in the course of changing lanes. We do not study the average error, to protect the driver's safety, the maximum error should be in safety bounds.

Specifically, in the trace tracking simulation, the double lane change (DLC) maneuver was adopted [40], because this maneuver could fully reflect the driving performance of the vehicle.

To verify the influence of the trajectory on the controller, different trajectories are used for tracking (Fig. 4). These two reference paths are both based on the ISO 3888-1 DLC maneuver. The difference between the two paths is that when changing lines, reference path 2 uses straight lines and the other uses curves generated by the method proposed in Method B.

As shown in Fig. 4, the maximum tracking error is 0.18 m with the reference path 1, and the other is 0.58 m. The results demonstrated that the performance of the controller made according to the method mentioned in the paper satisfied the requirements, at the meantime, proved the importance of trajectory planning. When changing lanes or driving on the corners, the optimized trajectory

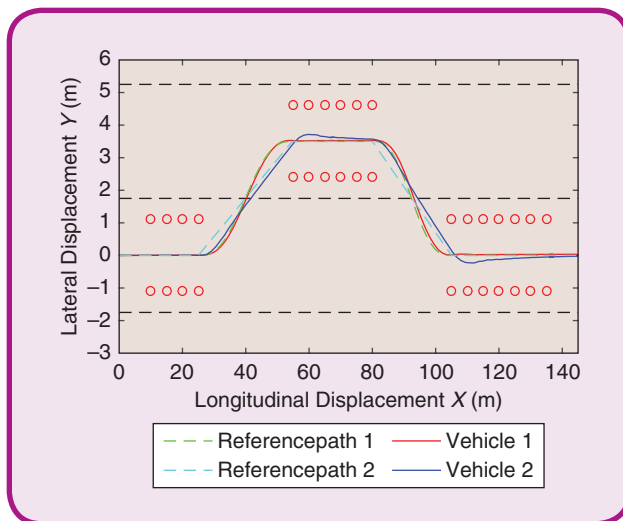


FIG 4 Comparison of tracking effects with the different reference path(average speed $v = 50$ km/h).

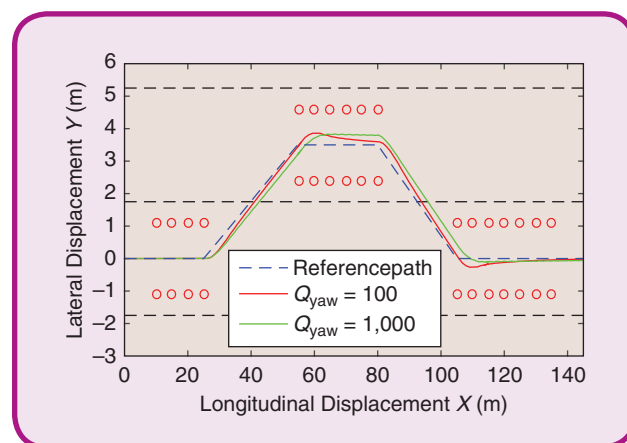


FIG 5 Different characteristics.

Table I. Maximum error at different speed.

Speed (km/h)	40	50	60
Maximum error (m)	0.05	0.18	0.45

makes the tracking controller perform better to ensure the safety of the vehicle.

There are many adjustable parameters in MPC controller. Different combinations may result in different effects, such as the weighting matrices \mathbf{Q} . Reference path 2 was adopted to fully reflect the influence of the weight coefficient of the heading angle. As shown in Fig. 5, while the heading weight coefficient Q_{yaw} is larger, the trace of the vehicle is parallel to the reference path. Another situation is that the vehicle gradually reduced the error with the reference path. Adjusting the parameters in the controller can reflect different drivers' characteristic. In [8], the

In the field of trajectory planning, there are two common assumptions, perfect perception and perfect tracker.

paths, using the smooth one as the reference path shows the influence of different vehicle velocity. As show in Fig. 6 the vehicle can follow the reference path very well at low speeds. When the vehicle drives at high speed, the maximum error occurs during the lane change process.

authors used the evaluation index of steering smoothness and path tracking accuracy reflects the drivers' characteristic. Therefore by changing the parameters of the cost function, it is easy to modify the control performance to be the same as the human driver's characteristic.

The choice of speed can also reflect driving characteristics. Based on the results of tracking different reference

Using the model predictive control method, the prediction model is a key part. As we introduce in the method, in the implementation, the key is to look for the most suitable reference position. The reference state is selected based on the predicted state. If the prediction state is not accurate, the reference state is not accurate too. Meanwhile under high-speed condition, vehicle kinematic model cannot predict the vehicle future state accurately, especially in the corners. Therefore, higher speed leads to higher error and returns to the reference path slower.

B. Trajectory Generation

In the field of trajectory planning, there are two common assumptions, perfect perception and perfect tracker [29]. Based on the assumptions, during the driving, the vehicle advanced one step, then planning once. Reference trajectory was updated every cycle.

Two kinds of trajectory planning strategies are proposed in this paper. One is the key point strategy, considering the real drivers' driving behavior, after reaching a key point such as the entrance or exit of the curves, then considering the next key point. In the DLC maneuver, the entrance and the exit of the traffic cones area are the key points (A,B,C,D). Reference trajectory 1 is generated based on this strategy. Until the vehicle reaches the nearest key point ahead, the nearest key point ahead is always the desired end point in each planning cycle.

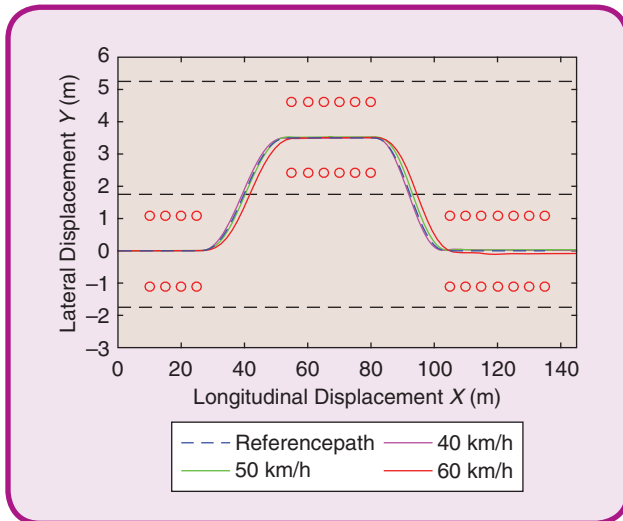


FIG 6 Trace tracking at different speed.

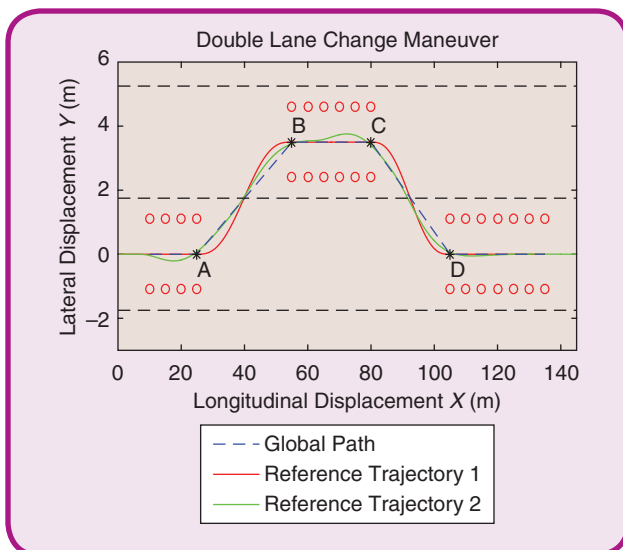


FIG 7 Different planning strategies.

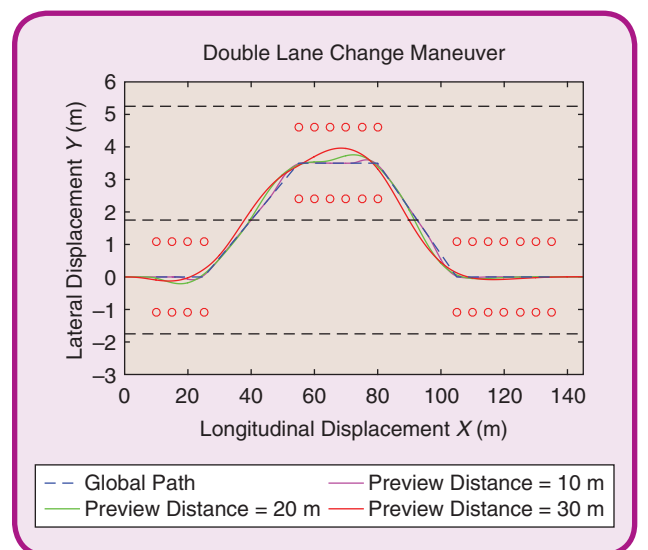


FIG 8 Different preview distance.

For example, based on the assumption that the controller is perfect, the vehicle perfectly tracks the reference trajectory. After the vehicle travels for a certain distance, the reference trajectory will be planned again. Since the desired end point is the same, the trajectory planned is the same as previous reference trajectory from the vehicle current position.

The other one is the preview distance strategy. It takes a desired point, which is fixed ahead of the vehicle as the end point of each planning cycle. Reference trajectory 2 takes the point on the global path that is 20 meters ahead of the vehicle as the desired point of each planning cycle. Thus, in each planning cycle, the desired end point is different. Therefore, the trajectory planned cannot coincide with the previous reference trajectory.

In each planning cycle, the difference in the desired end-point selection has a great influence on the result. As shown in Fig. 8, preview points are selected on the same global path, the trajectories generated are different because of the different preview distances. The preview distance and the lateral offset of the global path are the main influence factors to generate different trajectories. If the preview distance is longer, then the turn is earlier when passing the corner. The choice of the preview distance is also selected according to the driver's behavior and road conditions. Thus, according to the method, we can produce path-planning solutions that reflect different drivers' behavior, which will be studied in future publications.

C. On-Filed Test

The on-field test was carried out with a car BYD Qin provided by Wuhan Kotei Technnology Corporation. The vehicle is equipped with two IPC with i7 CPU. The model of the industrial computer is ARK-3500. The software are running in Windows 7 environment. The perceptual layer adopted multi-radar fusion system in the vehicle. On-field test includes fixed-path tracking and obstacle avoidance.

The experiment process was divided into several steps. First, a real driver drove the car around the test site for several circles to collect GPS data to get the global reference path. The fixed-path tracking experiment was carried out based on these data. Therefore, the fixed-path tracking experiment did not necessarily need the trajectory planner.

Due to site some restrictions, the maximum vehicle travel speed was set as 15 km/h. Like the simulation result, the maximum trajectory tracking occurred at the curves. This was because of the defects of using the vehicle kinematic model. In order to cross the sharp curves, the speed was set very low when entering the curves. The surroundings could influence the multi-radar fusion system and ultimately affected the experiment result. As shown in Fig. 10, the unmanned ground vehicle can finish fixed-path tracking successfully. As shown in simulation, the maximum error occurred in the corner, which is still in an acceptable range.

In the trajectory planning simulation, the tracker was able to track the trajectory accurately. However, it would not happen in real vehicle experiment, because the tracking process might generate errors. Therefore, it is necessary



FIG 9 Unmanned ground vehicle: BYD Qin.

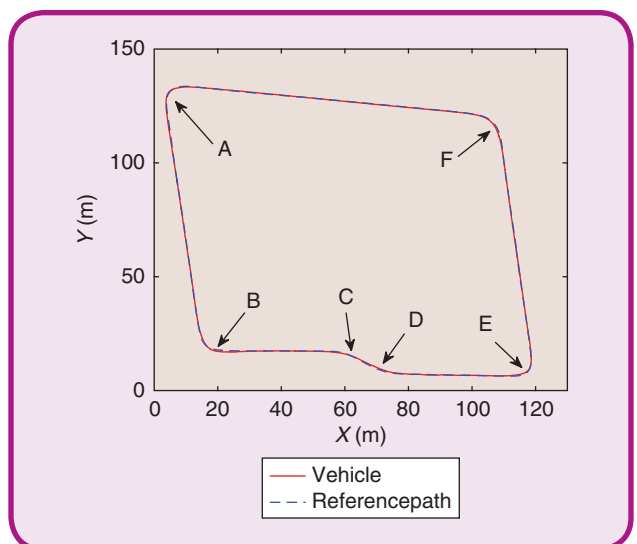


FIG 10 Unmanned ground vehicle fixed-path tracking.

The MPC controller based on the vehicle kinematic model can show good performance at low speed, but on highway roads, vehicle kinematic model cannot predict the vehicle future state accurately, especially in the corners

of vehicle tracking is less than 0.2 m when the speed is below 50 km/h. At the meantime, it proved the importance of smooth trajectory in autonomous vehicle movement. Besides, we compared the impact of different parameters both in tracking controller and trajectory planner. This paper also proposed two strategies on local planning strategies, i.e., the key point strategy

to continuously plan new trajectory based on the current vehicle position during the moving to modify the errors between the desired position and the current position.

To reflect the importance of the trajectory planner, obstacle-avoiding experiment was carried out. The traffic cones were placed on both sides of the centerline. They were separated by 8 meters in the longitudinal direction.

As shown in Fig. 12, the vehicle drove through the cones area successfully with a S-shaped trajectory. The route was not smooth due to the acquisition accuracy and sampling time.

IV. Conclusion

This paper proposed a joint framework for trajectory planning and tracking control. The planner and the tracker were proved through the simulation and the on-field test. The trajectory tracking control based on model predictive control could ensure that the maximum error

and the preview distance ahead strategy. Finally, the two algorithms were combined and experimentally proved by the on-field test.

The MPC controller based on the vehicle kinematic model can show good performance at low speed, but on highway roads, vehicle kinematic model cannot predict the vehicle future state accurately, especially in the corners. As shown in the simulation results, when the vehicle is driving at a speed higher than 50 km/h, the lateral error has exceeded 0.2 m. So that in the future, the vehicle kinematic model needs to be improved with vehicle dynamic model to test high speed scenarios. In our controller, we set slack variables in the objective function to prevent null solution problem. However, it cannot guarantee that the soft-constrained MPC solution is equal to the hard-constrained MPC solution. In order to solve this problem, the weighting of the slack variable need to be expressed by an exact penalty function.

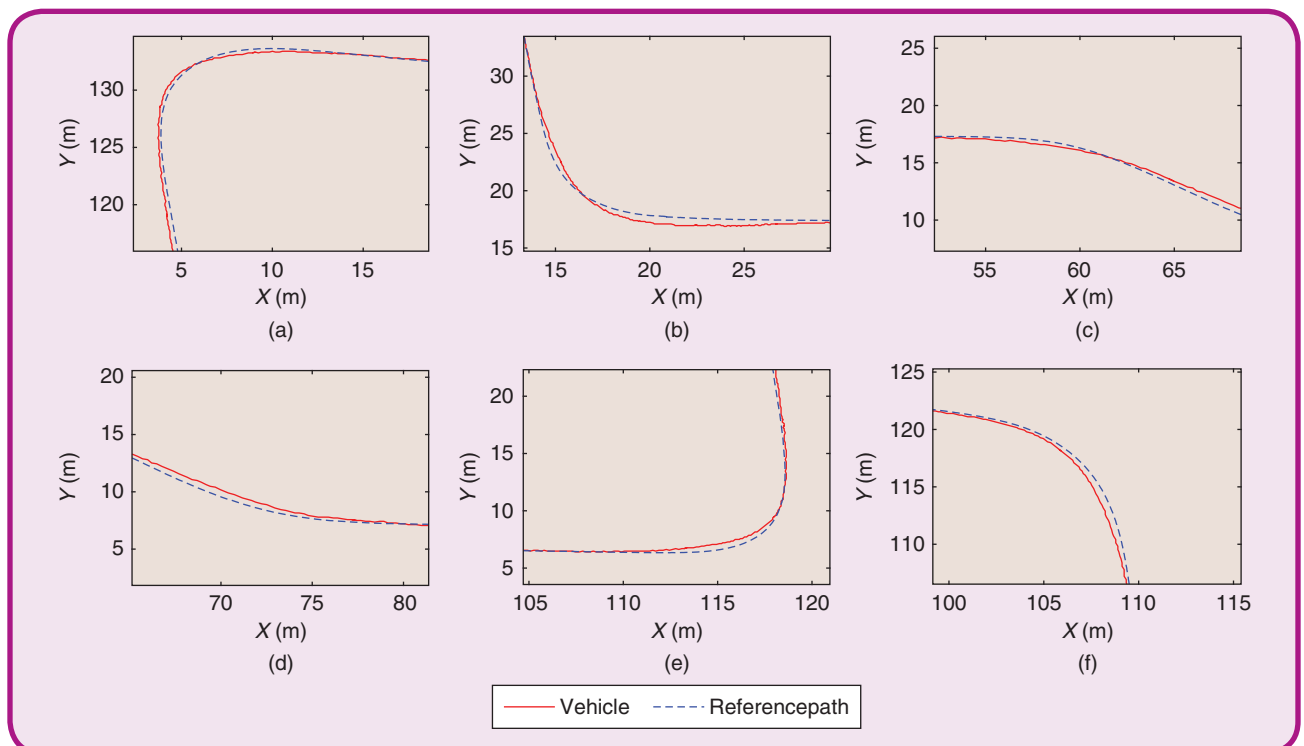


FIG 11 Running at corners. (a) Corner A, (b) corner B, (c) corner C, (d) corner D, (e) corner E, and (f) corner F.

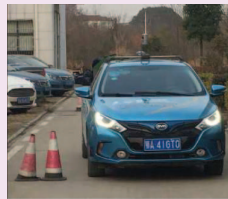


(a)

(b)



(c)



(d)

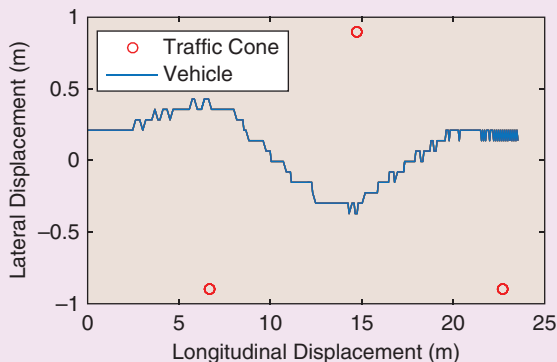
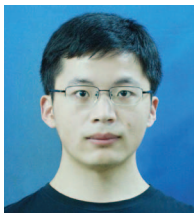


FIG 12 Obstacle avoiding test experiment (a) before starting, (b) the first obstacle, (c) the second obstacle, (d) the third obstacle.

Acknowledgment

This work is supported by the National Key R&D Program of China (2018YFB0105004), the National Natural Science Foundation of China (51675390, U1764262), the Major Project of Technological Innovation in Hubei Province (2016AAA007), and the Science Fund of State Key Laboratory of Automotive Safety and Energy (KF1807). The authors would like to thank Wuhan Kotei Technology Corporation for providing the autonomous vehicle testbed.

About the Authors



Chaoyong Zhang received B.E. degree in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China in 2014. He is currently working hard toward the M.E. degree with Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, China. His research interests include automated control, path planning, intelligent transportation system, etc.



Duanfeng Chu received the B.E. degree and the Ph.D. degree, both in mechanical engineering from Wuhan University of Technology, Wuhan, China, in 2005 and 2010, respectively.

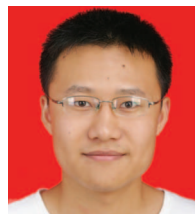
He is currently an Associate Professor of Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, China, focusing on the research of automated and connected vehicle, intelligent transportation systems, etc. He has visited University of California, Berkeley, USA, and the Ohio State University, USA, in 2009 and 2017, respectively.

Dr. Chu has been a reviewer for several international journals and conferences in the field of automated vehicle.



Shidong Liu received B.E. in vehicle engineering from Wuhan University of Technology, Wuhan, China in 2015. He is currently working hard toward the M.E. degree with Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan.

His research interests include automated control, path planning, intelligent transportation system, etc.



Zejian Deng received the B.S. degree in automation and M.S. degree in transportation engineering, both from Wuhan University of Technology, Wuhan, China, in 2015 and 2018, respectively. He is currently working towards the Ph.D. degree in

mechanical and mechatronics engineering with University of Waterloo, Waterloo, Canada. His research interests include driver model and decision-making in automated driving.



Chaozhong Wu received the B.E. degree in marine engineering, the M.S. degree in management science and engineering, and the Ph.D. degree in transportation engineering, all from Wuhan University of Technology, Wuhan, China in 1996, 1999 and 2002, respectively.

He is currently a Professor and the Director of Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, China. He is an expert in traffic safety, connected vehicle and intelligent transportation systems, etc. He has published over 60 journal articles and holds 11 patents. He was a visiting scholar of University of Regina, Canada, and Northeastern University, USA, in 2006 and 2015, respectively.

Dr. Wu was a recipient of the New Century Excellent Talent of Ministry of Education of China in 2010, the Young Talent of Ministry of Transport of China in 2012, the Distinguished Young Scholar of Hubei Province of China in 2012, and the Young Leading Talent of Ministry of Transport of China in 2014. He serves as an Associate Editor for some journals.



Xiaocong Su obtained his master of Electronics and Communication Engineering at the Huazhong University of Science and Technology in China in 2011.

He is currently working as a senior software engineer at Wuhan Kotei Technology Corporation. In 2013 he joined the team charged with the development of autonomous vehicle at Wuhan Kotei Technology Corporation in Wuhan.

References

- [1] A. Tigadi, R. Gujanatti, A. Gonchi, and B. Klemsscet, "Advanced driver assistance systems," *Int. J. Eng. Res. Gen. Sci.*, vol. 4, no. 3, pp. 151–158, 2016.
- [2] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, Winter 2014.
- [3] S. Thrun, "Toward robotic cars," *Commun. ACM*, vol. 53, no. 4, pp. 99–106, 2010.
- [4] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *Proc. American Control Conf. (ACC)*, IEEE, 2017, pp. 4025–4028.
- [5] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle Syst. Dyn.*, vol. 48, no. 10, pp. 1167–1192, 2010.
- [6] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Eng. Pract.*, vol. 19, no. 12, pp. 1459–1467, 2011.
- [7] J. Zhou and H. Peng, "Range policy of adaptive cruise control vehicles for improved flow stability and string stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 229–257, 2005.
- [8] A. Koga, H. Okuda, Y. Tazaki, T. Suzuki, K. Haraguchi, and Z. Kang, "Realization of different driving characteristics for autonomous vehicle by using model predictive control," in *Proc. Intelligent Vehicles Symp. (IV)*, IEEE, 2016, pp. 722–728.
- [9] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *Int. J. Vehicle Auton. Syst.*, vol. 3, no. 2–4, pp. 265–291, 2005.
- [10] X. Du, K. K. K. Htet, and K. K. Tan, "Development of a genetic-algorithm-based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6970–6977, 2016.
- [11] F. Oldewurtel et al., "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy Build.*, vol. 45, pp. 15–27, 2012.
- [12] M. A. Abbas, R. Milman, and J. M. Eklund, "Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles," *Can. J. Electr. Comput. Eng.*, vol. 40, no. 1, pp. 12–22, 2017.
- [13] C. Wang, X. Zhang, K. Guo, F. Ma, and D. Chen, "Application of stochastic model predictive control to modeling driver steering skills," *SAE Int. J. Passeng. Cars-Mech. Syst.*, vol. 9, no. 2016-01-0462, pp. 116–123, 2016.
- [14] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 5, pp. 566–580, 2007.
- [15] X. Li, Z. Sun, D. Liu, Q. Zhu, and Z. Huang, "Combining local trajectory planning and tracking control for autonomous ground vehicles navigating along a reference path," in *Proc. Intelligent Transportation Systems (ITSC)*, IEEE 17th Int. Conf., IEEE, 2014, pp. 725–751.
- [16] S. M. LaValle, *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Ames: Iowa State Univ., 1998.
- [17] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [18] S. Zhang, W. Deng, Q. Zhao, H. Sun, and B. Litkouhi, "Dynamic trajectory planning for vehicle autonomous driving," in *Proc. Systems, Man, and Cybernetics (SMC)*, IEEE Int. Conf., IEEE, 2013, pp. 4161–4166.
- [19] K. Chu, J. Kim, K. Jo, and M. Sunwoo, "Real-time path planning of autonomous vehicles for unstructured road navigation," *Int. J. Automot. Technol.*, vol. 16, no. 4, pp. 655–668, 2015.
- [20] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, "Path planning and obstacle avoidance for autonomous mobile robots: a review," in *Proc. Int. Conf. Knowledge-Based and Intelligent Information and Engineering Systems*. New York: Springer, 2006, pp. 537–544.
- [21] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *Digital Avionics Systems Conf. Proc. 21st IEEE*, 2002, vol. 2, pp. 8D2–8D2.
- [22] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [23] A. Bacha et al., "Odin: team VictorTango's entry in the DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 467–492, 2008.
- [24] J. Bohren et al., "Little ben: the ben franklin racing team's entry in the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 598–614, 2008.
- [25] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *J. Field Robot.*, vol. 25, no. 11–12, pp. 959–960, 2008.
- [26] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *Proc. Int. Symp. Artificial Intelligence, Robotics, Automation Space*, 2005, pp. 1–7.
- [27] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. Robotics and Automation (ICRA)*, IEEE Int. Conf., IEEE, 2012, pp. 2061–2067.
- [28] M. McNaughton, *Parallel Algorithms for Real-Time Motion Planning*. Pittsburgh, PA: Carnegie Mellon Univ., 2011.
- [29] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *Proc. Int. Conf. Intelligent Robotics and Applications*. New York: Springer, 2012, pp. 588–597.
- [30] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *Int. J. Robot. Res.*, vol. 22, no. 7–8, pp. 585–601, 2005.
- [31] M. McNaughton, C. Urmsion, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Proc. Robotics and Automation (ICRA)*, IEEE Int. Conf., IEEE, 2011, pp. 4889–4895.
- [32] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proc. Intelligent Vehicles Symp. (IV)*, IEEE, 2015, pp. 1094–1099.
- [33] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, 1996.
- [34] F. Kuhne, W. F. Lages, and J. G. da Silva Jr., "Model predictive control of a mobile robot using linearization," in *Proc. Mechatronics and Robotics*, 2004, pp. 525–550.
- [35] E. C. Kerrigan and J. M. Maciejowski, *Soft Constraints and Exact Penalty Functions in Model Predictive Control*. Cambridge, U.K.: Univ. of Cambridge, 2000.
- [36] J. M. Maciejowski, *Predictive Control with Constraints*. London: Pearson Education, 2002.
- [37] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 346–359, 2012.
- [38] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Proc. Intelligent Robots and Systems, IROS IEEE/RSJ Int. Conf.* IEEE, 2009, pp. 1879–1884.
- [39] T. M. Howard, *Adaptive Model-Predictive Motion Planning for Navigation in Complex Environments*. Pittsburgh, PA: Carnegie Mellon Univ., 2009.
- [40] S. Schnelle, J. Wang, H. Su, and R. Jagacinski, "A driver steering model with personalized desired path generation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 111–120, 2017.