IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 June 2023, accepted 4 July 2023, date of publication 12 July 2023, date of current version 18 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294713

RESEARCH ARTICLE

Trajectory Planning in Frenet Frame via Multi-Objective Optimization

JIANYU HUANG[1], ZUGUANG HE[2], YUTAKA ARAKAWA[1], (Member, IEEE), AND BILLY DAWTON[1]
[1]ISEE, Kyushu University, Fukuoka 819-0395, Japan
[2]WUSIE, South China University of Technology, Guangzhou 511442, China
Corresponding author: Jianyu Huang (huang.jianyu.757@s.kyushu-u.ac.jp)

ABSTRACT Autonomous vehicles are an essential tool for promoting the development of intelligent transportation systems (ITS) and can effectively reduce traffic accidents caused by human errors. As an important part of the automatic driving software system, path planning is responsible for generating the motion trajectory of the vehicle, which is the primary factor determining driving quality. However, solution space construction and optimization problem formulation remain challenging research areas in the field of path planning. In this paper, we propose a multi-objective optimization algorithm for static obstacle avoidance to improve the comfort, safety and anti-deviation of the planned trajectory. We decouple the lateral and longitudinal motion of the vehicle using the Frenet frame and discretize the driving state space to generate target states of the vehicle. Based on the initial and target states, we generate a set of lateral and longitudinal motion trajectories using quintic and quartic polynomials, respectively. In addition, we design a cost function that comprehensively considers the comfort, safety, and deviation distance of the road center line by combining an acceleration check, curvature check, and collision check. As part of the cost function, we propose a novel method to quantify the safety of candidate trajectories considering the size of obstacles. The experimental results show that the proposed algorithm can quantize the safety of candidate paths and improve comfort 13.47%, 32.19%, 59.36% and 18.60% on a straight road, curvy road, intersection and U-shaped road, respectively. Furthermore, the algorithm can improve anti-deviation by 63.72%, 13.86%, 44.36%, and 45.56% on a straight road, curvy road, intersection and U-shaped road, respectively.

INDEX TERMS Autonomous driving, intelligent transportation systems (ITS), trajectory planning, Frenet frame, convex optimization, cost function.

I. INTRODUCTION

Autonomous driving is an important development direction in the field of intelligent transportation technology. It can reduce traffic accidents caused by improper operations such as drunk driving, drowsy driving and speeding, while also relieving traffic congestion and improving the overall performance of a transportation system. The framework of an autonomous driving system is a hierarchical structure [1], as demonstrated in Figure 1. This hierarchical framework typically encompasses four key modules: perception, decision, path planning, and vehicle control [2]. The perception module serves as the front end of the autonomous driving system, utilizing sensors to gather data about the surrounding environment. This data is utilized to perform essential tasks such as localization, object detection, path prediction and object tracking. These tasks enable the system to identify and comprehend diverse elements and obstacles, including roads, vehicles, traffic signs, precise vehicle locations and predicted trajectories. The decision module receives information from the perception module, and then analyzes and reasons based on this information to make appropriate driving decisions. It considers factors such as traffic rules and passenger requirements and generates a behavioural strategy to respond to the current driving scenario. The planning module takes the behavioural strategy provided by the decision module and combines it with high-definition maps and real-time perception data to generate a planned path. This path guides the vehicle's driving

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

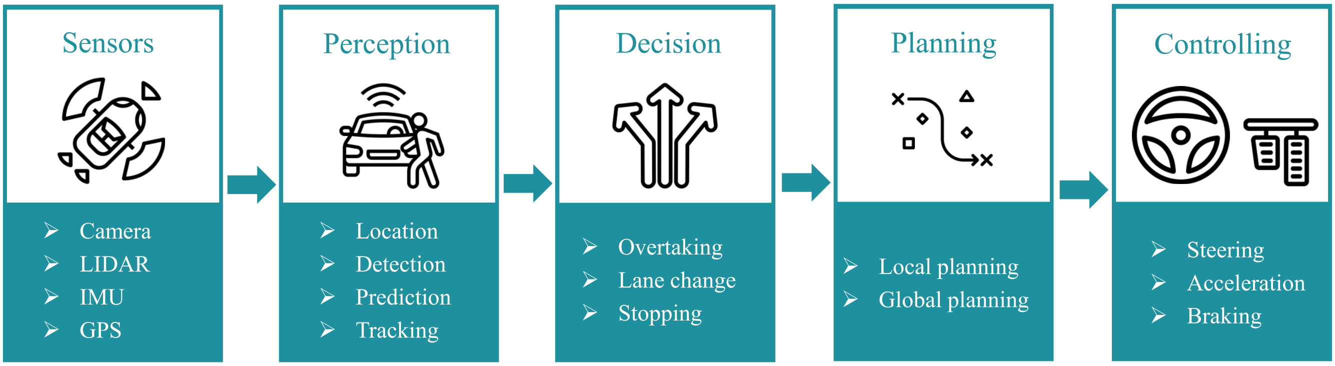


FIGURE 1. An illustration of function modules of the autonomous driving system.

direction, lane changes, turns, and ensures the vehicle reaches its destination safely and efficiently. The control module is responsible for precisely and smoothly tracking the planned path through actual vehicle operations, including acceleration, braking, steering, etc [3].

The path planning module plays a pivotal role in the hierarchical framework, serving as a vital bridge between the perception module, decision-making module and vehicle control module. Its essence lies in solving optimization problems with constraints in a complex convex space. Additionally, it is also involved in multi-agent clustering, obstacle avoidance and target tracking control, and is a key basic common problem. Therefore, the path planning algorithm is the core of autonomous driving technology. The performance of the algorithm directly determines the success or failure of autonomous driving [4], [5]. Path planning technology is broadly classified into two categories: the first category is global path planning, which aims to find the optimal or suboptimal path from the starting point to the destination point. The second category is local path planning, which involves obtaining environmental information through sensors in unknown or partially unknown environments, allowing autonomous driving vehicles to obtain a collision-free executable optimal planned path. It focuses on considering the current local environmental information on vehicles. A path with a timestamp is called a trajectory and includes both spatial information (location, curvature and heading) and temporal information (speed and acceleration).

Despite the numerous local planning algorithms [6], [7], [8] have been proposed since the DARPA Grand Challenge (2004, 2005) and the Urban Challenge (2007), they all present a common limitation: the utilization of the Cartesian frame cannot sufficiently describe the relationship between the position of the vehicle and the road. Therefore, in 2010, Werling et al. [9] proposed the use of the Frenet frame as an alternative solution. This coordinate system employs the road center line as the horizontal axis and represents the vehicle's offset from the road center line as the vertical axis. By adopting the Frenet frame, the relationship between the vehicle and the road can be intuitively described. Based on the Frenet frame, a complex three-dimensional motion can be decoupled into two independent two-dimensional motions, which effectively reduces the complexity of the path planning problem [10]. Consequently, this paper focuses on studying the local path planning problem under the hierarchical framework.

A. RELATED WORK

The essence of path planning is the search for an optimum in a non-convex space. However, it is well established that there is no optimal solution to the non-convex problem. Thus, researchers have been focusing on transforming non-convex problems into convex ones, formulating path planning problems to find optimal solutions. Currently, there are two main independent research orientations for this including the learning-based orientation and algorithm-based orientation. Table 1 shows a summary of the main methods of these two orientations.

The learning-based methodology simplifies the hierarchical structure. It straightforwardly inputs raw sensor data, such as camera-captured frames and radar point cloud data, into a multi-layer neural network for training. This process enables the network to output path points or even directly generate control commands [11], [12]. Typical approaches within this methodology include imitation learning methods utilizing various network structures, as well as deep reinforcement learning methods employing diverse reward strategies [13], [14], [15], [16]. The learning-based method streamlines the path planning process by directly mapping sensor inputs to control outputs without the need for explicit algorithms or intermediate steps. Although it can perform well in specific training scenarios, it may struggle to generalize to new or unseen situations, potentially resulting in poor performance or safety issues. Moreover, this approach demands substantial computational resources to find a feasible plan due to extensive trial and error in the training [17].

The algorithm-based methodology depends on the hierarchical structure and takes into account the behavioural strategy provided by the decision module, high-definition maps and real-time perception data from the perception module to generate a planned path. In contrast to the learning-based method, a notable strength of the algorithm-based

TABLE 1. Taxonomy of motion planning techniques applied in autonomous driving scenarios.

Algorithm Type	Technique Description	Advantages and Disadvantages	Existing Implementations
Learning-Based	realise the mapping from raw data sensor to path information or control commands after trail and error	*Advantages*: Simplify the mapping process from raw sensor data to control commands *Disadvantages*: lack of interpretability; weak generalization ability; cost substantial computational resources	[13], [14], [15], [16]
Artificial Potential Field	Abstract vehicle motion as a type of artificial gravitational field; Target point produces "gravity" on the vehicle and the obstacle produces "repulsive force" on the vehicle	*Advantages*: Simple mathematical description; good real-time performance *Disadvantages*: Trapped in local optimal solution; unable to reach destination; collision occurance	[20], [21]
Graph Search-Based	Known nodes/cells search space with associated weights; Grid and node/cells weight computation according to the environment	*Advantages*: High robustness (structured and unstructured roads) *Disadvantages*: Cannot meet the non-integrity constraints of the vehicle; large computation and time consuming	[22], [23], [24]
Random Sample-Based	Random-tree based algorithm; randomly samples in state space	*Advantages*: Suitable for multi-dimensional space; good real-time performance; simple principles; strong applicability *Disadvantages*: high randomness; hard to converge; low quality path; processing required	[25], [26], [27], [28], [29]
Discrete Optimization-Based	Discretize space; Formulate path optimization problem	*Advantages*: low computation; excellent real-time performance *Disadvantages*: Discretization resolution affects the optimization results	[30], [31], [32]

method is its interpretability, enabling the identification of defective modules when malfunctions or unexpected system behaviour occur. The main methods employed for this include artificial potential field-based methods, graph search-based methods, random sample-based methods, and discrete optimization-based methods [18], [19].

The artificial potential field method is a virtual force method proposed in [20]. It abstracts the vehicle motion as a type of artificial gravitational field. The target point produces "gravity" and the obstacle produces "repulsive force" on the vehicle. The algorithm of this method is concise in mathematical description and has good real-time performance, but easily falls into the local optimal solution problem [21]. Among the methods based on graph search, A* and D* algorithms are commonly used in [22], [23], and [24]. Although these algorithms are widely used in the field of robotics, their planned paths fail to satisfy the non-integrity constraints of vehicles and are computationally intensive and time-consuming. The best-known random sample-based method is Rapidly Exploring Random Tree (RRT) [25] which can efficiently solve high-DOF robot motion planning with differential constraints. It is difficult, however, to utilize the domain knowledge from the structured environment for quick convergence, and the computed trajectory is generally low quality and thus cannot be used directly without a post-processing step [26], [27], [28]. The algorithm proposed in [29] can generate high-performance trajectories given enough planning time. The long convergence to optimal trajectory computation time makes it unsuitable in rapidly changing environments, however. The discrete optimization-based method discretizes the state space and utilizes mathematical processes such as numerical integration and differentiation to generate the solution space for a limited number of candidate paths, before solving the optimal path by establishing an optimization expression. This method has been widely used in recent years because of its low computation and excellent real-time performance [30], [31], [32].

The work in [30] performs local path planning based on the discrete optimization-based method. The major drawback of this method is that, while it considers path safety, it ignores the effect of obstacle size on evaluating the safety of candidate paths. In certain scenarios, the optimal solution may not be unique for ensuring path safety, and the same obstacle avoidance action is taken regardless of the obstacle's size. Furthermore, the deviation from the road center line is not taken into account, which means that the vehicle may not actively return to the center line after avoiding obstacles. This can trigger the lane departure warning system frequently and increase the risk of collision due to lane departure.

To sum up, learning-based methods, artificial potential field methods, graph search-based methods, and random sample-based methods are prone to problems such as lacking interpretability, falling into a local optimum, requiring a large amount of calculation, being time-consuming, and having difficulty describing the relationship between vehicles and roads. However, discrete optimization-based methods have better real-time performance, which is why we use this kind of method in our proposed work rather than other mentioned methods. Furthermore, in this paper, our study is path planning for static obstacle avoidance based on discrete optimization methods which belong to algorithm-based methodology. Our research purpose is not to compare the learning-based methodology and the algorithm-based methodology to find their strength and weakness.

B. CONTRIBUTIONS

Considering the strengths and weaknesses of the above methods, in order to solve the trajectory planning problem in real time for autonomous driving, we propose a new multi-objective path planning algorithm for static obstacle avoidance based on discrete optimization methods. Our main contributions can be summarized as follows:

- We propose an efficient trajectory planning algorithm framework. To describe the complicated three-dimensional motion of the vehicle, our approach first uses the Frenet frame to decouple the vehicle's movement into two orthogonal movements: one longitudinal movement along the driving guideline and one lateral movement perpendicular to the road center line. Then, the state space is discretized to generate starting and ending position points for the path. Trajectory solution spaces are generated for both movements by connecting the sampled end conditions to the initial condition using quintic or quartic polynomials. The longitudinal and lateral trajectories are then combined to generate the trajectory sets. A set of driveable candidate trajectories are screened out through vehicle kinematics collision-avoidance constraints. Finally, the optimal trajectory is selected by minimizing a predefined cost function formulated for optimal path planning, taking into account comfort, safety, and road center line deviation.
- To evaluate the safety of candidate trajectories, we include obstacle size as a novel feature for consideration and calculate the safety loss of each candidate trajectory using the variation of the Gauss-Laplace operator.
- We demonstrate the performance of our proposed algorithm in four traffic scenarios. The experimental results show that the algorithm can generate optimal paths for a straight road, curvy road, intersection and U-shaped road, allowing autonomous vehicles to safely and comfortably avoid obstacles and complete the path planning from start to endpoint. On the straight road, the mean value of J_{jerk} decreases by 13.47% and the mean value of J_{offset} decreases by 63.72 %. On the curvy road, the mean value of J_{jerk} decreases by 32.19% and the mean value of J_{offset} decreases by 13.86%. On the intersection scenario, the mean value of J_{jerk} decreases by 59.36% and the mean value of J_{offset} decreases by 44.36%. On the U-shaped scenario, the mean value of J_{jerk} decreases by 18.60% and the mean value of J_{offset} decreases by 45.56%.

The remainder of this paper is structured as follows: we introduce the Frenet and Cartesian frames in Section II, present and illustrate the proposed algorithm in Section III, show the algorithm evaluation results in Section IV, before finally, concluding the paper and discussing potential avenues for future work in Section V.

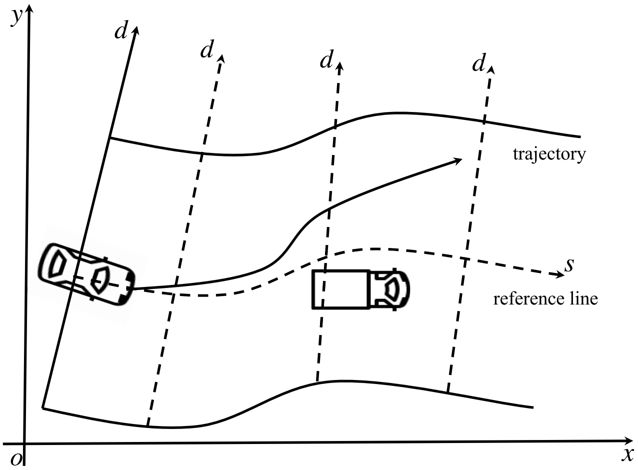


FIGURE 2. Vehicle motion in the Frenet and cartesian frames.

II. FRENET AND CARTESIAN FRAMES

A. FRENET FRAME FOR VEHICLE MOTION

As shown in Figure 2, in the Frenet frame, the center line of the road is generally selected as the reference line. The direction along the reference line is the s-axis, and the direction perpendicular to the reference line is the d-axis [33], [34], [35], [36].

Since the s-axis and d-axis are perpendicular to each other and the s-axis is parallel to the lane line, it is easier to describe the relationship between vehicle motion and road. Compared to the Cartesian frame, the Frenet frame simplifies the calculation of the offset distance of the vehicle from the road center line and the driving distance along the lane line without requiring consideration of road curvature. This coordinate system is widely used in path planning research for autonomous vehicles, robots, and unmanned aerial vehicles (UAVs) [37], [38], [39], [40], [41].

B. FROM FRENET FRAME TO CARTESIAN FRAME

As shown in Figure 3, the ego vehicle often needs to adjust its driving trajectory due to the presence of other vehicles and obstacles, instead of strictly following the reference line (i.e., the road center line). When in the Cartesian frame, we can describe the current state of the ego vehicle as $\left[\vec{x}, \theta_x, \kappa_x, v_x, a_x\right]$, where $\vec{x}$ represents the vehicle's position at $Q(x, y)$, and $\vec{n}_x$ and $\vec{t}_x$ are the unit normal and tangent vectors of the vehicle's motion trajectory at Q. Additionally, θ_x denotes the angle between $\vec{x}$ and the x-axis, κ_x represents the curvature at Q, v_x is the velocity of the ego vehicle, and a_x is the vehicle's acceleration. On the other hand, in a Frenet frame, the point P represents the projection of the vehicle's position point Q onto the reference line. The angle between $\vec{r}$ and the x-axis is denoted by θ_r, while $\vec{n}_r$ and $\vec{t}_r$ denote the

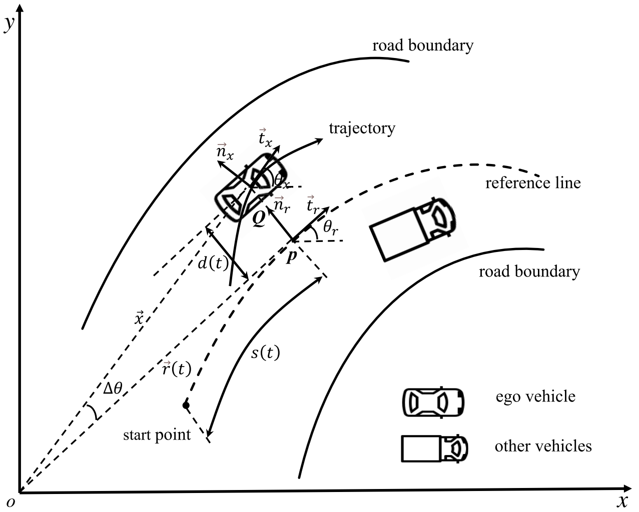


FIGURE 3. **Transformation from Frenet frame to cartesian frame.**

unit normal and tangent vectors of the reference line at point P. The vehicle's status can typically be described using the vector $\left[s, \dot{s}, \ddot{s}, d, \dot{d}, \ddot{d}, d', d''\right]$, where the distance between P and Q corresponds to the transverse displacement d, and the curve distance from the start point of the reference line to P represents the longitudinal displacement s. Additionally, d' denotes the first derivative of d with respect to s (i.e., $d' = \frac{dd}{ds}$), while d'' denotes the second derivative of d with respect to s (i.e., $d'' = \frac{dd'}{ds}$).

The equation of vehicle status in the Cartesian frame can be written as follows:

$$\begin{cases} x_x = x_r - d\sin\theta_r \\ y_x = y_r + d\cos\theta_r \\ \theta_x = \arctan\left(\frac{d'}{1-k_r d}\right) + \theta_r \in [-\pi, \pi] \\ v_x = \sqrt{[\dot{s}\,(1-k_r d)]^2 + (\dot{s}d')^2} \\ a_x = \ddot{s}\frac{1-k_r d}{\cos\Delta\theta} + \frac{\dot{s}^2}{\cos\Delta\theta}\cdot \\ \left[d'\left(k_x\frac{1-k_r d}{\cos\Delta\theta)} - k_r\right) - \left(k_r' d + k_r d'\right)\right] \\ k_x = \left(\left(d'' + \left(k_r' d + k_r d'\right)\tan\Delta\theta\right)\frac{\cos^2\Delta\theta}{1-k_r d} + k_r\right) \end{cases} \tag{1}$$

where $\Delta\theta = \theta_x - \theta_r$; κ_r is the curvature at point P.

III. PROPOSED ALGORITHM

A. ALGORITHM OVERVIEW

In a Cartesian frame, since the lateral and longitudinal motions of the vehicle are coupled, the calculation of vehicle motion is complex and large. Therefore, the Frenet frame is utilized to decouple the lateral and longitudinal motion of the vehicle. Furthermore, since the vehicle driving environment is a non-convex problem, and an optimal solution cannot be obtained, the non-convex problem is converted into a convex problem by discretizing the motion space. Finally, the optimal path is calculated according to the cost function and related constraints.

The framework of the proposed algorithm can be seen in Figure 4 (a). The algorithm consists of two main stages: trajectory generation in the Frenet frame and optimal trajectory selection.

In the stage consisting of trajectory generation in the Frenet frame, the main steps are as follows:

1) SPATIAL DISCRETIZATION

The road width is parameterized at Δd intervals along the d-axis, the upper boundary of the road is $d_{\text{roadwidth}}$, and the lower boundary is $-d_{\text{roadwidth}}$.

2) START/END STATUS INITIALIZATION

The statuses of the start and end trajectory points affect the generation of the path. Since the Frenet frame decouples the vehicle motion into lateral and longitudinal motion, the start and end statuses along the d-axis are denoted as $\left[d_s, \dot{d}_s, \ddot{d}_s, d_i, \dot{d}_i, \ddot{d}_i\right]$, where d_s is the coordinate of the start trajectory point along the d-axis, and d_i denotes the coordinates of the end trajectory point along the d-axis. In this paper, we assume the autonomous vehicle is under adaptive cruise control, and the desired speed of the vehicle is maintained in the forward direction, so the start and end statuses along the s-axis are $[s_s, \dot{s}_s, \ddot{s}_s, \dot{s}_i, \ddot{s}_i]$, where s_s is the coordinate of the start trajectory point along the s-axis; s_i is the coordinate of the end trajectory point along the s-axis.

3) TRAJECTORY GENERATION

Given the initial and final states of the vehicle along the d-axis as $\left[d_s, \dot{d}_s, \ddot{d}_s, d_i, \dot{d}_i, \ddot{d}_i\right]$, a set of six equations can be formulated to solve the polynomial coefficients, enabling the representation of the trajectory in the d direction using a quintic polynomial. Similarly, for the s-axis, with its start and end states denoted as $[s_s, \dot{s}_s, \ddot{s}_s, \dot{s}_i, \ddot{s}_i]$, the trajectory can be formulated using a quartic polynomial. Then, the planning cycle T is discretized by Δt, and the obtained time series $[0, \Delta t, \ldots, T]$ is brought into the d-axial and s-axial trajectory equations, and the d-axial and s-axial coordinates of the trajectory points in each trajectory are calculated. Finally, the corresponding d-axial and s-axial coordinates are combined to obtain the planned trajectories. The further details are demonstrated in Section III-B and Section III-C.

In the optimal trajectory selection stage, the main steps are as follows:

4) TRAJECTORY POINT COORDINATE TRANSFORMATION

According to Equation 1, the generated trajectories in the Frenet frame are remapped to the global Cartesian frame.

5) TRAJECTORY CHECK

A trajectory essentially consists of several trajectory points containing spatial-temporal information, i.e., location, curvature, heading, speed and acceleration, etc. In this paper, the trajectory check only examines the collision, acceleration and curvature of each trajectory point on the path. Through Section III-A2 and Section III-A3, we obtained the trajectory equations and the trajectory points. In this step, a virtual ego vehicle is mapped at each trajectory point, and whether there is a collision between the boundary of the virtual ego car and the boundary of the obstacle is checked, as depicted in Figure 4 (b). Considering the limitations of the vehicle kinematics and dynamics, the acceleration and curvature of each trajectory point on the trajectory are checked using hard constraints. The trajectories are filtered out if there is a trajectory point in the trajectory that cannot satisfy the collision-free, acceleration and steering constraints. Detailed operations and derivation processes are discussed within Section III-D. Through this step, the candidate trajectories are generated and collision-free.

6) OPTIMAL TRAJECTORY SELECTION

The total loss of each trajectory is calculated, and the trajectory with the minimum total loss is chosen as the optimal trajectory. The cost function is composed of three indicators: comfort, trajectory safety, and trajectory anti-deviation. These three indicators are defined in detail in Section III-E.

7) THE START POINT STATUS UPDATE

The planning frequency is denoted as $1/\Delta t$. As the ego vehicle moves forward, its position changes continuously. Thus, in our trajectory planning approach, the current position of the vehicle is used as a reference point for generating the next trajectory. We designate the current position of the vehicle as (x_0, y_{m0}), and calculate the trajectory point (x_1, y_{m1}) which is Δt seconds after the current time. This point serves as the new starting point for the subsequent round of trajectory planning.

B. LATERAL MOTION TRAJECTORY PLANNING

Lateral motion planning involves the tasks of obstacle avoidance and lane change. As Equation 2 shows, the lateral motion trajectory $d(t)$ can be modelled based on a quintic polynomial:

$$d(t) = c_{d0} + c_{d1}t + c_{d2}t^2 + c_{d3}t^3 + c_{d4}t^4 + c_{d5}t^5 \tag{2}$$

where c_{d0}, c_{d1}, c_{d2}, c_{d3}, c_{d4} and c_{d5} are the coefficients of a quintic polynomial.

Where $\dot{d}(t)$ is the derivative of Equation 2:

$$\dot{d}(t) = c_{d1} + 2c_{d2}t + 3c_{d3}t^2 + 4c_{d4}t^3 + 5c_{d5}t^4 \tag{3}$$

and $\ddot{d}(t)$ is the derivative of Equation 3:

$$\ddot{d}(t) = 2c_{d2} + 6c_{d3}t + 12c_{d4}t^2 + 20c_{d5}t^3 \tag{4}$$

The known statuses of the start point and end point at the time of t_s and t_e are introduced into Equation 2 to Equation 4:

$$\begin{pmatrix} d\,(t_s) \\ \dot{d}\,(t_s) \\ \ddot{d}\,(t_s) \\ d\,(t_e) \\ \dot{d}\,(t_e) \\ \ddot{d}\,(t_e) \end{pmatrix} = \begin{pmatrix} 1 & t_s & {t_s}^2 & {t_s}^3 & {t_s}^4 & {t_s}^5 \\ 0 & 1 & 2\,t_s & 3\,{t_s}^2 & 4\,{t_s}^3 & 5\,{t_s}^4 \\ 0 & 0 & 2 & 6\,t_s & 12\,{t_s}^2 & 20\,{t_s}^3 \\ 1 & t_e & {t_e}^2 & {t_e}^3 & {t_e}^4 & {t_e}^5 \\ 0 & 1 & 2\,t_e & 3\,{t_e}^2 & 4\,{t_e}^3 & 5\,{t_e}^4 \\ 0 & 0 & 2 & 6\,t_e & 12\,{t_e}^2 & 20\,{t_e}^3 \end{pmatrix} \begin{pmatrix} c_{d0} \\ c_{d1} \\ c_{d2} \\ c_{d3} \\ c_{d4} \\ c_{d5} \end{pmatrix} \tag{5}$$

The following Equation 6 can be obtained by modifying Equation 5, and the matrix of coefficients can be calculated as:

$$\begin{pmatrix} c_{d0} \\ c_{d1} \\ c_{d2} \\ c_{d3} \\ c_{d4} \\ c_{d5} \end{pmatrix} = \begin{pmatrix} 1 & t_s & {t_s}^2 & {t_s}^3 & {t_s}^4 & {t_s}^5 \\ 0 & 1 & 2\,t_s & 3\,{t_s}^2 & 4\,{t_s}^3 & 5\,{t_s}^4 \\ 0 & 0 & 2 & 6\,t_s & 12\,{t_s}^2 & 20\,{t_s}^3 \\ 1 & t_e & {t_e}^2 & {t_e}^3 & {t_e}^4 & {t_e}^5 \\ 0 & 1 & 2\,t_e & 3\,{t_e}^2 & 4\,{t_e}^3 & 5\,{t_e}^4 \\ 0 & 0 & 2 & 6\,t_e & 12\,{t_e}^2 & 20\,{t_e}^3 \end{pmatrix}^{-1} \begin{pmatrix} d\,(t_s) \\ \dot{d}\,(t_s) \\ \ddot{d}\,(t_s) \\ d\,(t_e) \\ \dot{d}\,(t_e) \\ \ddot{d}\,(t_e) \end{pmatrix} \tag{6}$$

C. LONGITUDINAL MOTION TRAJECTORY PLANNING

In this paper, when planning the longitudinal motion trajectory, we assume the vehicle's speed remains constant along the reference line direction. Therefore, the position configuration of the endpoint can be disregarded for now, and the trajectory can be represented by a quartic polynomial equation, as shown in Equation 7.

$$s(t) = c_{s0} + c_{s1}t + c_{s2}t^2 + c_{s3}t^3 + c_{s4}t^4 \tag{7}$$

where c_{s0}, c_{s1}, c_{s2}, c_{s3} and c_{d4} are the coefficients of a quartic polynomial, where $\dot{s}(t)$ is the derivative of Equation 7:

$$\dot{s}(t) = c_{s1} + 2c_{s2}t + 3c_{s3}t^2 + 4c_{s4}t^3 \tag{8}$$

and where $\ddot{s}(t)$ is the derivative of Equation 8:

$$\ddot{s}(t) = 2c_{s2} + 6c_{s3}t + 12c_{s4}t^2 \tag{9}$$

Equation 10 can be obtained from Equations 7-9, with the known configuration of the start point and the end point at the time of t_s and t_e:

$$\begin{pmatrix} s\,(t_s) \\ \dot{s}\,(t_s) \\ \ddot{s}\,(t_s) \\ \dot{s}\,(t_e) \\ \ddot{s}\,(t_e) \end{pmatrix} = \begin{pmatrix} 1 & t_s & {t_s}^2 & {t_s}^3 & {t_s}^4 \\ 0 & 1 & 2\,t_s & 3\,{t_s}^2 & 4\,{t_s}^3 \\ 0 & 0 & 2 & 6\,t_s & 12\,{t_s}^2 \\ 1 & t_e & {t_e}^2 & {t_e}^3 & {t_e}^4 \\ 0 & 1 & 2\,t_e & 3\,{t_e}^2 & 4\,{t_e}^3 \\ 0 & 0 & 2 & 6\,t_e & 12\,{t_e}^2 \end{pmatrix} \begin{pmatrix} c_{s0} \\ c_{s1} \\ c_{s2} \\ c_{s3} \\ c_{s4} \end{pmatrix} \tag{10}$$

Finally, the coefficients of the quartic polynomial can be obtained by transforming Equation 10:

$$\begin{pmatrix} c_{d0} \\ c_{d1} \\ c_{d2} \\ c_{d3} \\ c_{d4} \end{pmatrix} = \begin{pmatrix} 1 & t_s & {t_s}^2 & {t_s}^3 & {t_s}^4 \\ 0 & 1 & 2\,t_s & 3\,{t_s}^2 & 4\,{t_s}^3 \\ 0 & 0 & 2 & 6\,t_s & 12\,{t_s}^2 \\ 0 & 1 & 2\,t_e & 3\,{t_e}^2 & 4\,{t_e}^3 \\ 0 & 0 & 2 & 6\,t_e & 12\,{t_e}^2 \end{pmatrix}^{-1} \begin{pmatrix} s\,(t_s) \\ \dot{s}\,(t_s) \\ \ddot{s}\,(t_s) \\ \dot{s}\,(t_e) \\ \ddot{s}\,(t_e) \end{pmatrix} \tag{11}$$

D. CANDIDATE PATH GENERATION

Figure 4 shows that collisions can occur on the generated trajectories, and that there are limitations on the vehicle's motion and dynamic characteristics. To enhance the system's response time, trajectories that fail to meet the constraints are eliminated through trajectory checking. The remaining trajectories are then presented as candidate paths for the subsequent module to choose the best path. The primary components of the check involve collision, curvature, and acceleration checks.

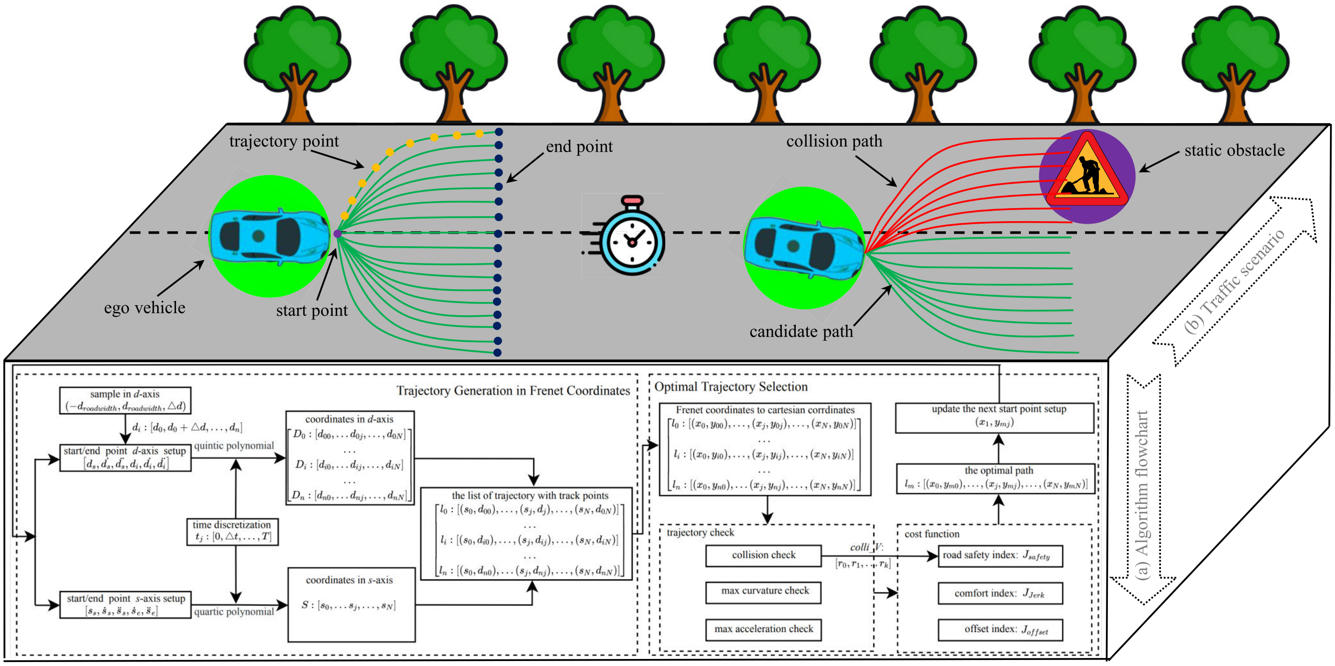

trajectory point
end point
collision path
static obstacle
ego vehicle
start point
candidate path
(b) Traffic scenario
(a) Algorithm flowchart
Trajectory Generation in Frenet Coordinates
sample in d-axis $(-d_{roadwidth}, d_{roadwidth}, \triangle d)$
$d_i : [d_0, d_0 + \triangle d, \dots, d_n]$
start/end point d-axis setup $[d_s, \dot{d_s}, \ddot{d_s}, d_i, \dot{d_i}, \ddot{d_i}]$
quintic polynomial
coordinates in d-axis
$D_0 : [d_{00}, \dots d_{0j}, \dots, d_{0N}]$ … $D_i : [d_{i0}, \dots d_{ij}, \dots, d_{iN}]$ … $D_n : [d_{n0}, \dots d_{nj}, \dots, d_{nN}]$
time discretization $t_j : [0, \triangle t, \dots, T]$
start/end point s-axis setup $[s_s, \dot{s}_s, \ddot{s}_s, \dot{s}_e, \ddot{s}_e]$
quartic polynomial
coordinates in s-axis
$S : [s_0, \dots s_j, \dots, s_N]$
the list of trajectory with track points
$l_0 : [(s_0, d_{00}), \dots, (s_j, d_j), \dots, (s_N, d_{0N})]$ … $l_i : [(s_0, d_{i0}), \dots, (s_j, d_{ij}), \dots, (s_N, d_{iN})]$ … $l_n : [(s_0, d_{n0}), \dots (s_j, d_{nj}), \dots, (s_N, d_{nN})]$
Optimal Trajectory Selection
Frenet coordinates to cartesian corrdinates
$l_0 : [(x_0, y_{00}), \dots, (x_j, y_{0j}), \dots, (x_N, y_{0N})]$ … $l_i : [(x_0, y_{i0}), \dots, (x_j, y_{ij}), \dots, (x_N, y_{iN})]$ … $l_n : [(x_0, y_{n0}), \dots (x_j, y_{nj}), \dots, (x_N, y_{nN})]$
update the next start point setup (x_1, y_{mj})
the optimal path
$l_m : [(x_0, y_{m0}), \dots, (x_j, y_{mj}), \dots, (x_N, y_{mN})]$
trajectory check
collision check
max curvature check
max acceleration check
$colli\_V$: $[r_0, r_1, \dots, r_k]$
cost function
road safety index: J_{safety}
comfort index: J_{Jerk}
offset index: J_{offset}

FIGURE 4. Proposed algorithm framework. (a) the flowchart of the algorithm; (b) the illustration of the proposed algorithm in a real traffic scenario.

1) COLLISION CHECK

Circles are used as bounding boxes for collision check. The conditions for passing the collision check are as follows:

$$(x_{ij} - x_{ob})^2 + (y_{ij} - y_{ob})^2 < (r_{car} + r_{ob})^2 \tag{12}$$

where (x_{ob}, y_{ob}) is the coordinates of a given obstacle; (x_{ij}, y_{ij}) are the coordinates of the trajectory point j on trajectory i; r_{car} is the radius of the circle bounding box of the ego vehicle; r_{ob} is the radius of the circle bounding box of the ego vehicle.

2) CURVATURE CHECK

The curvature of each point in the generated trajectories can be calculated using Equation 1.

$$\kappa_x[ij] \leqslant \kappa_{max}, 0 \leqslant j \leqslant N \tag{13}$$

where κ_{max} is the maximum allowed curvature, and N is the number of trajectory points.

3) ACCELERATION CHECK

The acceleration of each point in the generated trajectories can also be calculated using Equation 1.

$$a_x[ij] \leqslant a_{max}, 0 \leqslant j \leqslant N \tag{14}$$

where a_{max} is the maximum allowed acceleration.

E. COST FUNCTION

After the trajectory check, a set of candidate trajectories is generated. However, the number of candidates remains large, and we must choose a single trajectory to follow. To do so, we develop a cost function that assesses each candidate trajectory and selects the one with the lowest cost as the most optimal. This paper's cost function is designed based on three criteria: the trajectory's comfort, safety, and deviation from the road's center line. This cost function $J[i]$ is defined in Equation 15:

$$J[i] = w_0 J_{jerk}[i] + w_1 J_{safety}[i] + w_2 J_{offset}[i] \tag{15}$$

where J_{jerk}, J_{safety} and J_{offset} are the cost function of jerk, trajectory safety, and offset from the road center line respectively, and w_0, w_1, w_2 are the weights of the three cost functions, which determine the driving style of the vehicle.

1) JERK COST FUNCTION

Jerk is one of the most significant indicators for measuring the quality of planned trajectories. In general, the derivative of acceleration is used to assess jerk. Since the Frenet frame decouples transverse and longitudinal motion, the cost function of jerk can be constructed by summing the derivatives of the transverse and longitudinal accelerations.

$$J_{jerk}[i](t) = \int_{t_0}^{t_1} \dddot{d}^2(t)\mathrm{d}t + \int_{t_0}^{t_1} \dddot{s}^2(t)\mathrm{d}t \tag{16}$$

2) TRAJECTORY SAFETY COST FUNCTION

While all candidate paths are free of collisions, their safety levels may still differ due to their proximity to obstacles. It is evident that trajectories that are further from obstacles are safer. The [30] proposes a method to assess the safety of candidate trajectories based on their distance from obstacles, but this approach fails to account for the impact of obstacle size on trajectory safety. When the distance between the

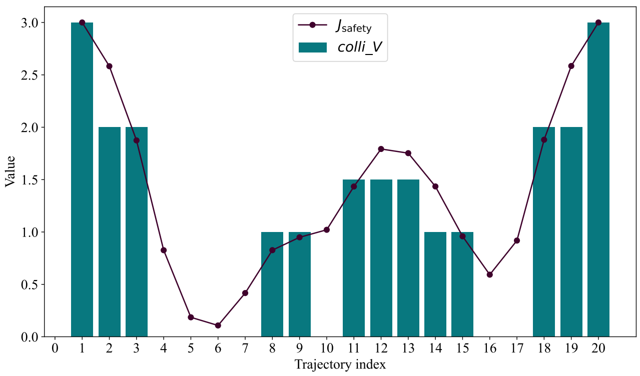


FIGURE 5. Safety cost of each candidate trajectory.

candidate path and the center of the obstacle is the same, the actual distance from the candidate trajectory to the obstacle's edge varies. Therefore, the safety of the candidate trajectory should be higher when faced with smaller obstacles, and the cost of safety should be lower.

In this paper, we draw inspiration from the Gauss-Laplace operator and the concept of convolution in the computer vision field to construct a convolution kernel, denoted as $f(x)$, and perform a convolution operation on the collision value, *colli_V*, to obtain the safety cost of each trajectory.

$$J_{\text{safety}}[i] = f * colli\_V[i] \tag{17}$$

The definition of the convolution kernel $f(x)$ is as follows:

$$f(x) = g(x) + \min(|g(x)|) \tag{18}$$

where $g(x)$ is the variation of Gauss Laplace operator:

$$g(x) = \frac{1}{2\pi\sigma} \cdot e^{-\frac{(x-u)^2}{2\sigma^2}} \cdot \frac{\sigma^2 - x^2}{\sigma^4} \tag{19}$$

and σ and u are the standard deviation and mean value, respectively.

Since any plane geometry can be surrounded by its circumscribed circle, in this paper, we use the circumscribed circle as the mathematical model to describe obstacles. The collision value *colli_V* is represented by the size of the largest obstacle on the trajectory:

$$colli\_V[i] = \begin{cases} r_{ob}[i], & \text{collision happen} \\ , & \text{collision free} \end{cases} \tag{20}$$

where r_{ob} is the radius of the obstacle-circumscribed circle.

As shown in Figure 5, the collision value reflects whether a collision occurs on the generated trajectories for different obstacle sizes. The safety cost of each candidate trajectory can be effectively calculated according to the size and distribution of the obstacles. The closer to a large obstacle, the higher the safety cost of the trajectory and the lower the safety of the trajectory. The closer to the dense obstacle pile, the higher the safety cost and the lower the trajectory safety.

TABLE 2. Experiment parameters.

	Parameter Name	Value
r_{car} (m)	Vehicle collision radius	1
$d_{\text{roadwidth}}$ (m)	Road width	10
w_0	Weight of J_{jerk}	0.4
w_1	Weight of J_{safety}	0.3
w_2	Weight of J_{offset}	0.3
Δt (s)	Sampling time	0.1
Δd (m)	Transverse sampling distance	0.5
$k_{\max}$ (m^{-1})	Maximum curvature	0.5
$a_{\max}$ ($\text{m} \cdot \text{s}^{-2}$)	Maximum acceleration	3
$T_{\max}$ (s)	Maximum planning period	5
$T_{\min}$ (s)	Minimum planning period	4
V_{target} ($\text{km} \cdot \text{h}^{-1}$)	Target speed	50

3) TRAJECTORY ANTI-DEVIATION COST FUNCTION

While driving, deviating from the center line of the road can result in insufficient space to maneuver around obstacles, requiring a large steering angle to avoid them. This driving behaviour can cause inconvenience to other drivers and increase the risk of misjudgments, leading to traffic accidents. Additionally, it may frequently trigger lane departure alarms, leading to functional conflicts between modules in the driving system. Therefore, when selecting the optimal trajectory, it is crucial to ensure that the trajectory follows the road's center line as closely as possible. In this paper, we design a cost function that calculates the integral of the square of the transverse displacement, d, of the candidate trajectory to satisfy this requirement.

$$J_{\text{offset}}\,[i] = \int_{s_0}^{s_1} d^2(s)\mathrm{d}s \Big/ \int_{t_0}^{t_1} s^2(t)\mathrm{d}t \tag{21}$$

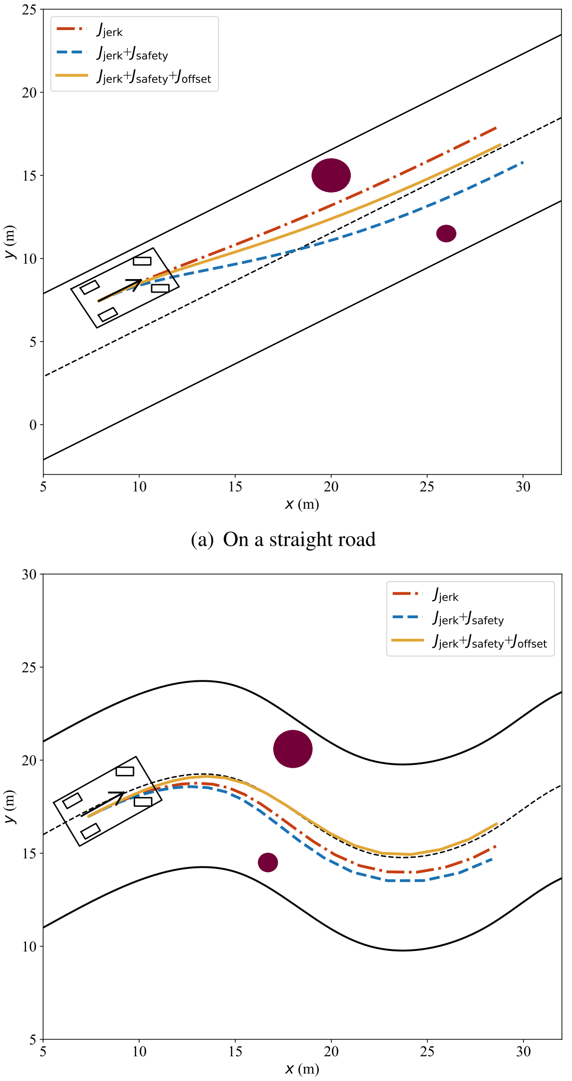
Since the planning time has been discretized, the Equation 21 also can be expressed as follows:

$$J_{\text{offset}}[i] = \sum_{j=0}^{N} d_{ij}^2 \Big/ \sum_{j=0}^{N} s_{ij}^2 \tag{22}$$

where N is the number of trajectory points in trajectory i.

IV. SIMULATION AND ANALYSIS

In the research on path planning, it is commonly assumed that map, localization and detection information have been obtained, without delving into the discussion of the implementation of front-end module functionalities in autonomous driving systems. Referring to the traffic scenarios in current public datasets of perception field [42], [43], [44], [45] and the simulation environment setting in [46], [47], and [48]. In the simulation, a straight road, a curvy road, an intersection scenario and a "U" shaped road are built in a Python environment, and several static obstacles of different sizes are set up on the roads. The experiments in this paper are divided into two parts: the first part analyzes the impact of different cost functions on trajectory generation, and the second part compares the advantages and disadvantages of the methods proposed in this paper and that proposed in [30]. The simulation parameters are shown in Table 2.



(a) On a straight road

(b) On a curvy road

FIGURE 6. **Influence of cost function on trajectory selection.**

A. INFLUENCE OF COST FUNCTION ON TRAJECTORY SELECTION

As shown in Figure 6 (a) and (b), different cost functions can result in different optimal trajectories on both straight and curvy roads. This paper analyzes the impact of three different cost functions on trajectory selection. When the cost function only takes into account jerk, the vehicle will attempt to drive in its current direction without colliding. While this can provide a comfortable ride, it may not be safe enough as it can bring the vehicle too close to obstacles. The second cost function considers both J_{jerk} and J_{safety} indicators. Since the safety indicator in this paper takes into account the size of the obstacle, it is clear that under the influence of the safety indicator, the vehicle will choose the corresponding avoidance distance based on the size of the obstacle. However, the resulting trajectory may still deviate from the road center line after obstacle avoidance. The third cost function considered in this paper, which takes into account jerk, safety, and offset from the road center line, produces a smoother trajectory that maintains a safe distance from obstacles and returns to the center line of the road after obstacle avoidance.

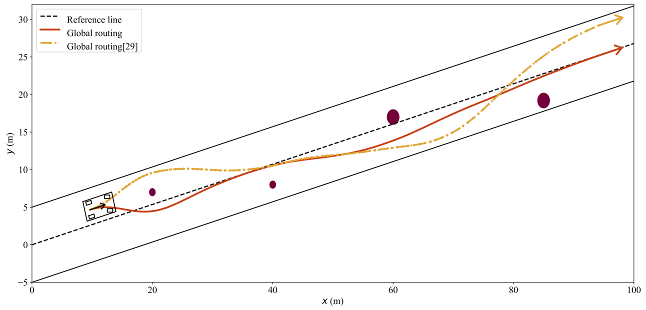


FIGURE 7. **Comparison of global routing on a straight road.**

B. PERFORMANCE OF PROPOSED ALGORITHM

To gauge the feasibility and effectiveness of the proposed local path planning algorithm, we simulated both the algorithm proposed in this paper and the algorithm proposed in reference [30], and analyzed their performance on the following four traffic scenarios. The results of comparisons are summarized in Table 3.

1) STRAIGHT ROAD SCENARIO

In order to validate the ability of the algorithm to continuously avoid obstacles, this paper builds a straight road scene with static obstacles. Figure 7 illustrates that both the method proposed in this paper and the one in reference [30] can successfully avoid obstacles and reach the destination safely. However, the global routing of the method proposed in reference [30] changes greatly, almost forming an "s" curve, and the trajectory cannot return to the center of the road after avoiding obstacles. This is because the safety cost function of the trajectory in reference [30] is insensitive to the size of the obstacle, and it measures the safety of candidate trajectories mechanically by the distance from the obstacle. The method tends to choose trajectories far away from the obstacle as the optimal path when avoiding obstacles, and it lacks the constraint of the center line keeping, which results in the lateral avoidance distance often being too large. In contrast, the method proposed in this paper takes into account the three indicators of J_{jerk}, J_{safety}, and J_{offset}, which enables it to avoid obstacles safely and return to the road center line after avoiding obstacles. The global route calculated in this paper is relatively smoother.

Figure 8 more clearly illustrates the deviation of the vehicle from the center line of the road. The proposed method in this paper outperforms the method in reference [30] in terms of returning the vehicle to the road center line after avoiding obstacles, thus reducing the risks and danger caused by making a large detour to avoid obstacles. The mean value of J_{offset} decreases by 63.72%. Figure 9 shows that on the straight road, during the first 20 seconds, the algorithm proposed in this paper requires the consideration of the road center line

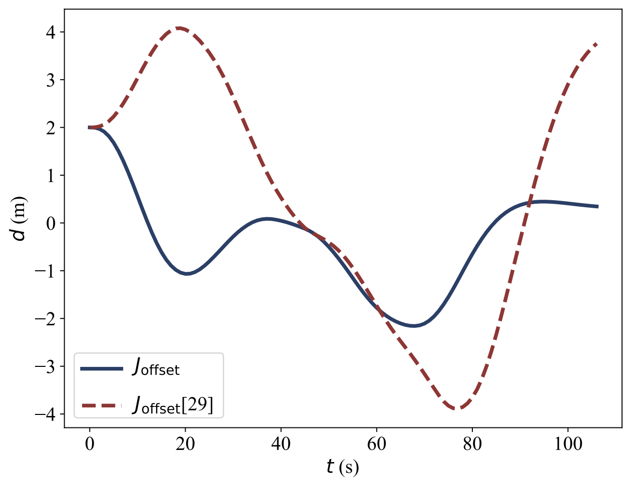


FIGURE 8. Comparison of road center line offset on a direct road.

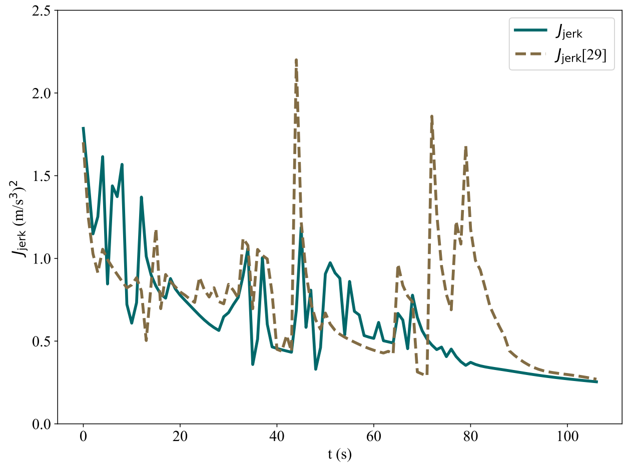


FIGURE 9. Comparison of comfort on a direct road.

offset, causing J_{jerk} to be slightly higher than the algorithm proposed in reference [30]. However, in the subsequent obstacle avoidance process, J_{jerk} significantly improves, resulting in a 13.47% reduction in the mean value of J_{jerk} throughout the process.

2) CURVY ROAD SCENARIO

To verify the effectiveness of the algorithm in continuous curvy road scenes, an S-shaped road scene was constructed.

Figure 10 demonstrates the complete driving trajectory of the vehicle from the starting point to the ending point on the ''S'' shaped road. The vehicle on the curvy road uses the cost function proposed in this paper to start obstacle avoidance at approximately 22 m and successfully returns to the road center line after avoiding the obstacle due to the influence of J_{offset}. On the other hand, the method proposed in [30] starts obstacle avoidance at approximately 25 m. Although the longitudinal avoidance distance is shorter than that of the method proposed in this paper, the absence of J_{offset} as a constraint means that the vehicle does not return to the road center line until 60 m after avoiding the obstacle, despite constantly approaching the road center line.

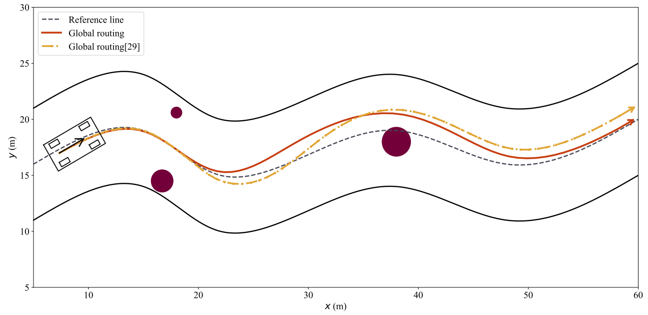


FIGURE 10. Comparison of global routing on a curvy road.

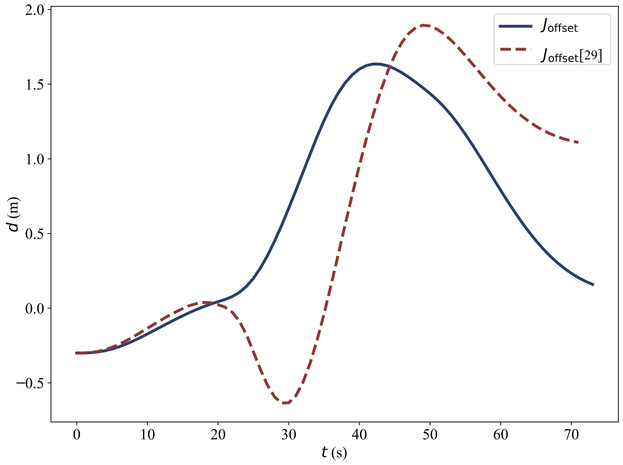


FIGURE 11. Comparison of road center line offset on a curvy road.

As demonstrated in Figure 11, from approximately 20 s to 40 s, the J_{offset} in this paper is larger, that is because our method starts to avoid the static obstacle at approximately 22 m, earlier than the method proposed in [30] and returns to road center line in time. In terms of the entire process, the mean value of J_{offset} decreases 13.86% on the curvy road.

Figure 12 shows a comparison of J_{jerk}. On the curvy road, the J_{jerk} in the entire process is smaller, and the mean value of J_{jerk} decreases by 32.19%. It indicates the algorithm proposed in this paper can provide better driving comfort.

3) INTERSECTION SCENARIO

Intersections are one of the most common traffic scenarios. To evaluate the performance of the algorithms, an intersection traffic scenario is constructed.

Figure 13 shows the complete driving trajectory of the vehicle during the obstacle avoidance process at the intersection. Both the algorithms proposed in this paper and in reference [30] are effective in obstacle avoidance. However, in this paper, the cost function considers the impact of obstacle size on road safety, enabling effective handling of distances maintained with obstacles of varying sizes. Additionally, considering the deviation from the road center line, the path actively returns to the road center line after evading obstacles. On the other hand, reference [30] only relies on the distance between obstacles to evaluate the safety value of

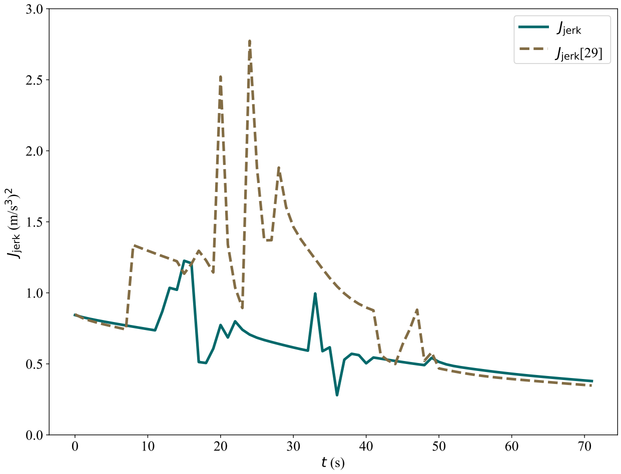


FIGURE 12. Comparison of comfort on a curvy road.

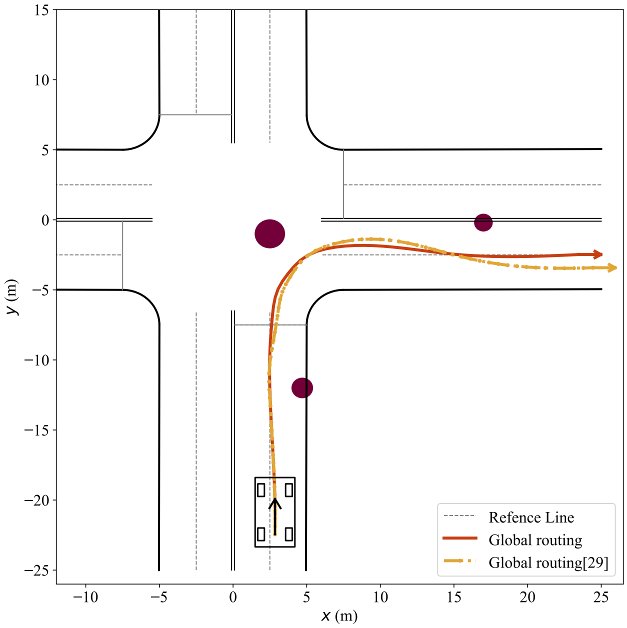


FIGURE 13. Comparison of global routing in intersection scenario.

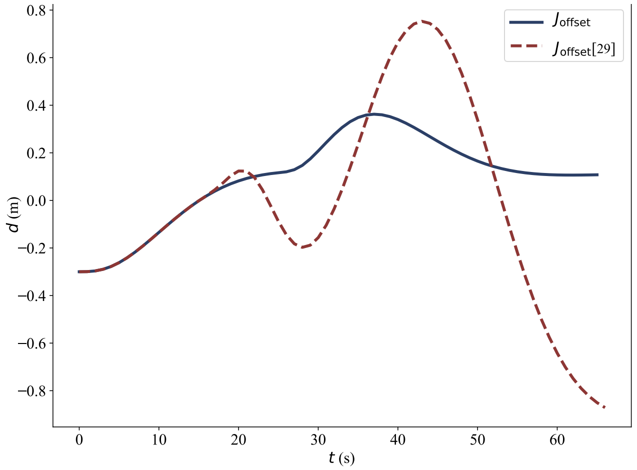


FIGURE 14. Comparison of road center line offset in intersection scenario.

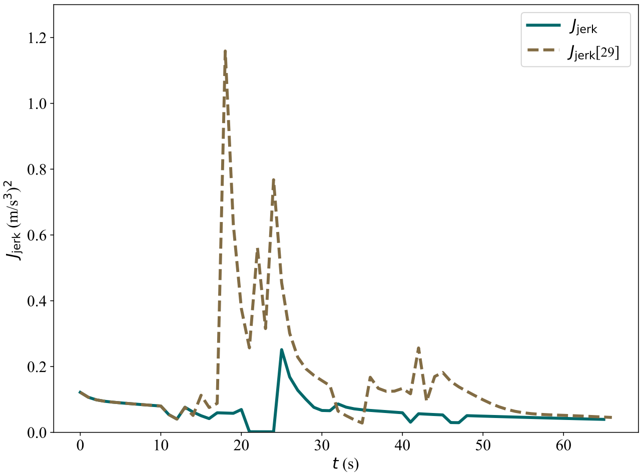


FIGURE 15. Comparison of comfort in intersection scenario.

candidate paths. Consequently, it tends to keep a maximum distance from obstacles during obstacle avoidance, resulting in less smooth and more variable trajectories. Furthermore, due to the lack of consideration for maintaining the lane center line, it often fails to return to the road center line after obstacle avoidance and continues in the current direction under the influence of J_{jerk}.

Figure 14 illustrates the deviation from the road center line throughout the entire trajectory. From approximately 0 s to 18 s, both algorithms exhibit similar deviations from the road center line. However, between 18 s and 35 s, the algorithm proposed in this paper, influenced by J_{jerk} and J_{offset}, aims to maintain the current direction of travel while ensuring a safe distance from the second encountered obstacle. On the other hand, the algorithm in reference [30] tends to keep a maximum distance from obstacles, leading to a deviation towards the opposite side of the lane center line. Due to the absence of constraints on road center line deviation, the effect of the J_{jerk} results in a slightly larger turning radius. Additionally, similar to other traffic scenarios, it is challenging for the algorithm to return to the road center line even on straight road segments after obstacle avoidance. In contrast, the trajectory planned by the algorithm in this paper appropriately selects a safe distance from obstacles while ensuring a smooth return to the road center line after obstacle avoidance. The mean value of J_{offset} decreases 44.36%.

Figure 15 shows a comparison of J_{jerk} in the intersection scenario. It is evident that there is a significant variation in jerk values during obstacle avoidance. In this traffic scenario, the algorithm proposed in this paper maintains a good level of driving comfort and decreases J_{jerk} by 59.36% compared to the algorithm proposed in the reference [30].

4) U-SHAPED ROAD SCENARIO

U-shaped roads are also a common and challenging traffic scene. Therefore, in this study, a U-shaped road is constructed, and static obstacles of different sizes are placed at the turning points of the U-shaped road.

TABLE 3. Performance comparison.

Scenario	Algorithm	J_{offset}	Decline Rate(%)	J_{jerk}	Decline Rate(%)
Straight Road	proposed	0.83	63.72%	0.63	13.47%
	[30]	2.29		0.73	
Cruvy Road	proposed	0.69	13.86%	0.6	32.19%
	[30]	0.8		0.9	
Intersection	proposed	0.18	44.36%	0.06	59.36%
	[30]	0.33		0.16	
U-shaped Road	proposed	0.42	45.56%	0.07	18.60%
	[30]	0.77		0.08	

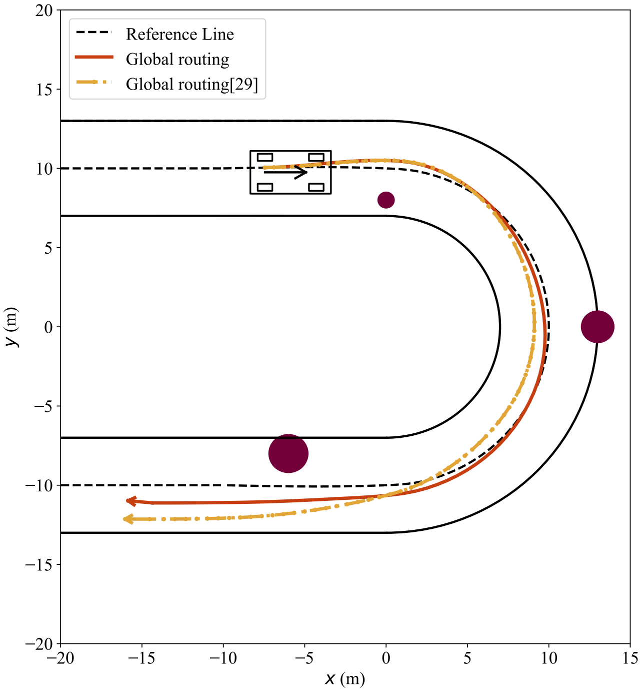


FIGURE 16. Comparison of global routing on a U-shaped road.

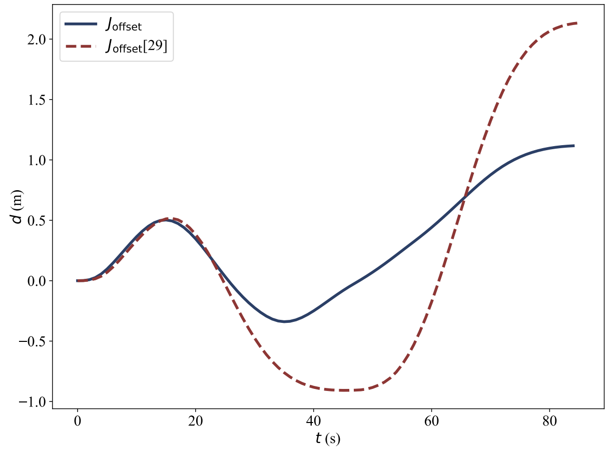


FIGURE 17. Comparison of road center line offset on a U-shaped road.

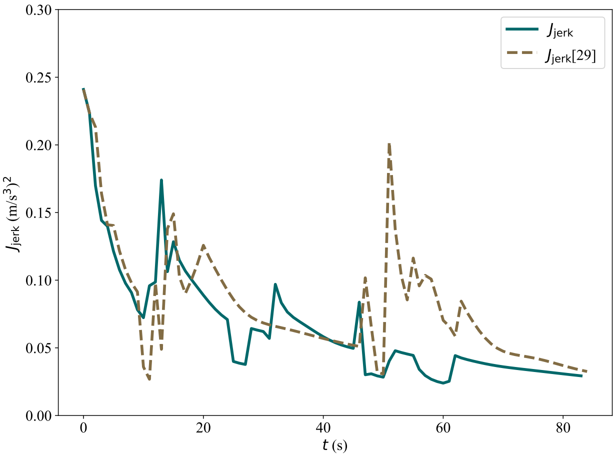


FIGURE 18. Comparison of comfort on a U-shaped road.

Figure 16 depicts the entire driving routing of continuously avoiding static obstacles in the U-shaped road. Both algorithms can effectively avoid obstacles. However, similar to other simulated traffic scenarios, in reference [30] the evaluation of the safety values of the candidate paths is based solely on the distance from the obstacles. As a result, regardless of the size of the obstacles, the algorithm tends to maintain a safe distance by deviating as far as possible from the obstacles while ensuring comfort. This leads to a significant deviation from the road center line in the planned path, especially at the turning points of the U-shaped road. In contrast, this study incorporates the size of the obstacles into the evaluation of the safety values of the candidate paths. It selects an appropriate deviation distance based on the size of the obstacles and actively seeks to return to the road center line after obstacle avoidance, within the constraints of the offset.

Figure 17 shows the comparison of road center line deviation on the U-shaped road. It can be observed that the proposed method in this paper is able to maintain driving near the road center line under the dual challenges of obstacle avoidance and turning. However, the method proposed in reference [30] deviates more from the center line due to the lack of the constraint of J_{offset}. The mean value of deviation decreases by 45.56%. Figure 18 is the comparison of driving comfort. From 0 s to 50 s, there is little difference in jerk values between the two algorithms. However, from 50 s to 80 s, the proposed method in this paper shows a significant improvement in jerk compared to the method proposed in reference [30]. This is because, when facing the last obstacle, the algorithm in this paper chooses a path that aligns roughly with the original direction after the turn. In contrast, the method in the reference always tends to stay as far away from the obstacle as possible, resulting in increased J_{jerk} due to excessive manoeuvres. The mean value of J_{jerk} decreases by 18.60%.

V. CONCLUSION

In this paper, we propose a local path planning algorithm for autonomous driving based on the Frenet frame. The algorithm first decouples the transverse and longitudinal motion using the Frenet frame and generates a cluster of candidate trajectories in the $s-d$ coordinate system based on the start and end point state information. The algorithm then selects the optimal trajectory based on a new cost function that considers the comfort, safety, and offset from the road center line. In the candidate trajectory selection stage, we propose a new cost function to select the optimal trajectory. This cost function is designed to comprehensively consider comfort, safety, and offset from the road center line. We also consider the size of obstacles in the cost function of trajectory safety, which further improves performance. By minimizing the cost function, the algorithm selects the optimal trajectory that satisfies the constraints and provides the best driving experience. Experimental results show that different cost functions will lead to different final optimal trajectories.

Compared with the reference method proposed in [30], the trajectories planned on the straight road, curvy road, intersection and U-shaped road based on the method proposed in this paper can safely avoid obstacles and actively return to the road center line. In addition, the mean value of J_{jerk} respectively decreased by 13.47%, 32.19%, 59.36% and 18.60% on the straight road, curvy road, intersection and U-shaped road indicating that the trajectory planned by our method is more comfortable. The mean value of J_{offset} decreased by 63.72%, 13.86%, 44.36%, and 45.56% on the straight road, curvy road, intersection and U-shaped road, respectively, which shows that our algorithm is capable of swiftly guiding a vehicle back to the center line of the road after evading obstacles.

The algorithm proposed in this paper only addresses the static obstacle avoidance problem in path planning and doesn't account for dynamic obstacles. As a result, our future research will concentrate on tackling the challenge of avoiding dynamic obstacles.

REFERENCES

[1] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458–488, 2022.

[2] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, and L. Chen, "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Trans. Intell. Vehicles*, early access, May 9, 2023, doi: 10.1109/TIV.2023.3274536.

[3] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, Feb. 2017, doi: 10.3390/machines5010006.

[4] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[5] S. Li, Z. Li, Z. Yu, B. Zhang, and N. Zhang, "Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control," *IEEE Access*, vol. 7, pp. 132074–132086, 2019.

[6] V. Hoang, D. C. Hernández, J. Hariyono, and K.-H. Jo, "Global path planning for unmanned ground vehicle based on road map images," in *Proc. 7th Int. Conf. Hum. Syst. Interact. (HSI)*, Jun. 2014, pp. 82–87.

[7] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[8] X. Cao, X. Zou, C. Jia, M. Chen, and Z. Zeng, "RRT-based path planning for an intelligent litchi-picking manipulator," *Comput. Electron. Agricult.*, vol. 156, pp. 105–118, Jan. 2019.

[9] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 987–993.

[10] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on Frenet frame: A Cartesian-based trajectory planning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15729–15741, Sep. 2022.

[11] F. Ye, S. Zhang, P. Wang, and C. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 1073–1080.

[12] H. Zhou, J. Laval, A. Zhou, Y. Wang, W. Wu, Z. Qing, and S. Peeta, "Review of learning-based longitudinal motion planning for autonomous vehicles: Research gaps between self-driving and traffic congestion," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2676, no. 1, pp. 324–341, Jan. 2022.

[13] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 673–683, Jan. 2023, doi: 10.1109/TIV.2022.3225340.

[14] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14128–14147, Sep. 2022, doi: 10.1109/TITS.2022.3144867.

[15] Y. Du, J. Chen, C. Zhao, C. Liu, F. Liao, and C.-Y. Chan, "Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103489. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004757

[16] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022, doi: 10.1109/TITS.2020.3024655.

[17] Z. Ajanović, E. Regolin, B. Shyrokau, H. Ćatić, M. Horn, and A. Ferrara, "Search-based task and motion planning for hybrid systems: Agile autonomous vehicles," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105893, doi: 10.1016/j.engappai.2023.105893.

[18] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[19] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1376–1386, Feb. 2020.

[20] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 1398–1404.

[21] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–19, Dec. 2019.

[22] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[23] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1994, pp. 3310–3317.

[24] A. Candra, M. A. Budiman, and K. Hartanto, ‘‘Dijkstra's and A-Star in finding the shortest path: A tutorial,’’ in *Proc. Int. Conf. Data Sci., Artif. Intell., Bus. Anal. (DATABIA)*, Medan, Indonesia, Jul. 2020, pp. 28–32, doi: 10.1109/DATABIA50434.2020.9190342.

[25] J. J. Kuffner and S. M. LaValle, ‘‘RRT-connect: An efficient approach to single-query path planning,’’ in *Proc. ICRA Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, Apr. 2000, pp. 995–1001.

[26] S. M. LaValle and J. J. Kuffner, ‘‘Randomized kinodynamic planning,’’ *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[27] R. Chen, J. Hu, and W. Xu, ‘‘An RRT-Dijkstra-based path planning strategy for autonomous vehicles,’’ *Appl. Sci.*, vol. 12, no. 23, Nov. 2022, Art. no. 11982.

[28] Y. Zhang, H. Sun, J. Zhou, J. Hu, and J. Miao, ‘‘Optimal trajectory generation for autonomous vehicles under centripetal acceleration constraints for in-lane driving scenarios,’’ in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3619–3626.

[29] J. h. Jeon, S. Karaman, and E. Frazzoli, ‘‘Anytime computation of time-optimal off-road vehicle maneuvers using the RRT,’’ in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, Dec. 2011, pp. 3276–3282.

[30] M. WEI, D. TENG, and S. WU, ‘‘Trajectory planning and optimization algorithm for automated driving based on Frenet coordinate system,’’ *Control Decis.*, vol. 36, no. 4, pp. 815–824, 2021.

[31] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, ‘‘Baidu Apollo EM motion planner,’’ 2018, *arXiv:1807.08048*.

[32] Y. Zhang, H. Sun, J. Zhou, J. Pan, J. Hu, and J. Miao, ‘‘Optimal vehicle path planning using quadratic optimization for Baidu Apollo open platform,’’ in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 978–984.

[33] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang, and K. Liu, ‘‘Hybrid trajectory planning for autonomous driving in highly constrained environments,’’ *IEEE Access*, vol. 6, pp. 32800–32819, 2018.

[34] W. Lim, S. Lee, M. Sunwoo, and K. Jo, ‘‘Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 341–355, Jan. 2021.

[35] S. Manzinger, C. Pek, and M. Althoff, ‘‘Using reachable sets for trajectory planning of automated vehicles,’’ *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 232–248, Jun. 2021.

[36] F. Milano, G. Tzounas, I. Dassios, and T. Kërçi, ‘‘Applications of the Frenet frame to electric circuits,’’ *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 4, pp. 1668–1680, Apr. 2022.

[37] Y. Li, ‘‘Motion planning for dynamic scenario vehicles in autonomous-driving simulations,’’ *IEEE Access*, vol. 11, pp. 2035–2047, 2023.

[38] J. Chen, R. Zhang, W. Han, W. Jiang, J. Hu, X. Lu, X. Liu, and P. Zhao, ‘‘Path planning for autonomous vehicle based on a two-layered planning model in complex environment,’’ *J. Adv. Transp.*, vol. 2020, pp. 1–14, Nov. 2020.

[39] Y. Sun, D. Ren, S. Lian, S. Fu, X. Teng, and M. Fan, ‘‘Robust path planner for autonomous vehicles on roads with large curvature,’’ *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2503–2510, Apr. 2022.

[40] M. Moghadam and G. H. Elkaim, ‘‘An autonomous driving framework for long-term decision-making and short-term trajectory planning on Frenet space,’’ in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 1745–1750.

[41] B. Li and Y. Zhang, ‘‘Fast trajectory planning in Cartesian rather than Frenet frame: A precise solution for autonomous driving in complex urban scenarios,’’ *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17065–17070, 2020.

[42] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, ‘‘The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections,’’ in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1929–1934.

[43] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, ‘‘INTERACTION dataset: An INTERnational, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps,’’ 2019, *arXiv:1910.03088*.

[44] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, ‘‘Argoverse: 3D tracking and forecasting with rich maps,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8740–8749.

[45] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, ‘‘NuScenes: A multimodal dataset for autonomous driving,’’ 2019, *arXiv:1903.11027*.

[46] T. Chen, Y. Cai, L. Chen, and X. Xu, ‘‘Trajectory and velocity planning method of emergency rescue vehicle based on segmented three-dimensional quartic Bezier curve,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3461–3475, Mar. 2023.

[47] C. Ma, C. Yu, and X. Yang, ‘‘Trajectory planning for connected and automated vehicles at isolated signalized intersections under mixed traffic environment,’’ *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103309, doi: 10.1016/j.trc.2021.103309.

[48] W. Xu, J. Pan, J. Wei, and J. M. Dolan, ‘‘Motion planning under uncertainty for on-road autonomous driving,’’ in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2507–2512.

JIANYU HUANG was born in Jiangsu, China, in 1992. He received the B.S. degree in automotive engineering and the M.Sc. degree in traffic and transportation engineering from Jiangsu University, Zhenjiang, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan. From 2018 to 2022, he was a senior engineer with automobile industry. His current research interests include vehicle path planning, vehicle dynamics control, and intelligent vehicle.

ZUGUANG HE was born in Guangdong, China, in 2001. He is currently pursuing the bachelor's degree in robotics engineering with Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, China. His research interests include intelligent robot and robot trajectory planning and control.

YUTAKA ARAKAWA (Member, IEEE) received the B.E., M.Sc., and Ph.D. degrees from Keio University, Japan, in 2001, 2003, and 2006, respectively. He is currently a Professor with the Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University. His current research interests include human activity recognition, behavior change support systems, and location-based information systems. He is a member of ACM, IPSJ, and IEICE.

BILLY DAWTON received the B.E. degree in electrical and electronic engineering from the University of Sussex, Brighton, U.K., in 2014, the M.Sc. degree in wireless and optical communications from University College London, London, U.K., in 2015, and the Ph.D. degree in information science and electrical engineering from Kyushu University, Japan, in 2022. He is currently a Specially Appointed Assistant Professor with the Department of Information Science and Electrical Engineering, Kyushu University. His current research interests include low-cost and low-complexity classification frameworks and NLP-based tourist information recommender systems.