

RESEARCH ARTICLE

Intelligent Vehicle Path Based on Discretized Sampling Points and Improved Cost Function: A Quadratic Programming Approach

CHENGTAO ZHANG^{1,2} AND WEIHANG XU¹¹School of Mechanical and Automotive Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China²Engineering Research Center for Advanced Design and Manufacturing of Heavy Vehicle Components, Ministry of Education, Liuzhou 545006, China

Corresponding author: Chengtao Zhang (zhchtzct@gxust.edu.cn)

This work was supported in part by the Central Guidance for Local Scientific and Technological Development Funds under Grant Guike ZY23055014 and in part by the National Natural Science Foundation of China under Grant 52202491.

ABSTRACT The paper introduces a quadratic programming algorithm for real-time local path planning of autonomous vehicles. The algorithm relies on discretized sampling points and an enhanced cost function. Initially, we formulate the cost function to optimize the reference trajectory and establish the Frenet coordinate system. The drivable region undergoes discretization to generate sampling points in the Frenet coordinate system. We apply the principles of convex spatial obstacle avoidance to define the vehicle's drivable area, taking into account the vehicle's kinematics and establishing barrier boundary conditions. Subsequently, quadratic programming is employed to determine an optimal path within the vehicle's drivable area. Concurrently, two cost functions are devised, the first evaluates the distance between the vehicle and obstacles, while the second assesses ride comfort, these cost functions are employed to evaluate sampling points and speed profiles, facilitating the planning of an optimal speed profile on the selected path. Finally, the algorithm undergoes validation through co-simulation using Matlab/Simulink, PreScan, and CarSim software. Various road scenarios, including straight and S-curve roads with both dynamic and static obstacles, are created to assess the method's feasibility. The test results demonstrate the algorithm's efficacy in avoiding moving and stationary obstacles and generating an ideal path compliant with driving conditions.


INDEX TERMS Autonomous driving, path planning, quadratic programming, cost function.

I. INTRODUCTION

The forefront of vehicle technology is currently occupied by autonomous driving [1]. Path planning, a fundamental component of autonomous vehicles [2], initially drew its theoretical foundations from the field of robotics. Autonomous driving has subsequently optimized and adapted these methods to meet the specific demands of vehicles and address the complexities of diverse traffic environments [3]. In the context of autonomous driving, path planning can be categorized into two primary domains [4]: Global path planning, relying on high-precision maps, and local path planning, designed to respond to changes in the vehicle's immediate surroundings. Local path planning involves creating short-term vehicle movement paths within a specific environment [5], [6]. Its

primary objective is to facilitate safe and efficient movement in the vicinity of the current position, aligning with the objectives set by global path planning [7]. Local path planning frequently complements global path planning, with the latter determining the comprehensive route from the starting point to the endpoint. Meanwhile, local path planning is responsible for managing a specific segment of this path, adapting to constraints, and navigating obstacles in response to environmental changes [8].

Conventional path planning algorithms include the artificial potential field algorithm and Dijkstra's algorithm [9]. The artificial potential field algorithm is a prevalent technique in robot navigation and path planning [8]. Drawing inspiration from the concept of potential fields in physics, this algorithm models the path of a robot or a moving entity by simulating particle motion within a potential field [10]. At its core, this approach guides the robot along a feasible path by

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

creating a potential field that attracts the target to the robot while repelling the robot from obstacles [11]. In light of this, Luo et al. introduced a path-planning algorithm that enhances the artificial potential field by adjusting the radius of the virtual potential field detection circle, effectively mitigating the issue of local minima [12]. Dijkstra's algorithm serves as a solution to the shortest path problem, facilitating the determination of the shortest paths from a starting node to all other nodes within a directed graph [13]. In the inaugural Driverless Challenge of 2004, sponsored by the Defense Advanced Research Projects Agency (DARPA), vehicles were tasked with autonomously navigating desert terrain [14]. Many teams integrated Dijkstra's algorithm into their vehicle systems. While conventional path-planning methods are both widespread and efficient, they may not always yield the globally optimal solution when confronted with complex problems. Consequently, traditional path-planning algorithms should be supplemented with other methods when dealing with intricate driving environments [15].

Machine learning, a subfield of artificial intelligence, has numerous applications in the realm of path planning [16], [17]. It empowers vehicles to autonomously acquire knowledge, enhancing the adaptability of path planning to environmental fluctuations and mission specifications [18]. Yin et al. introduced an N-step prioritized dual DQN path planning algorithm, specifically designed to resolve the issue of robot obstacle avoidance through the incorporation of a reward screening mechanism [19]. Han et al. presented an enhanced dual-DQN path planning method that optimizes the reward function by employing a two-branch network structure. This refinement expedites model convergence, leading to the generation of stable obstacle avoidance paths [17]. While machine learning can adapt to local environments, its models require a substantial volume of training data, exhibit diminished reliability and security, and grapple with handling uncertainties in path planning. Consequently, machine learning-based approaches have not witnessed widespread implementation in real-world driving scenarios [20], [21].

Dynamic programming provides a viable approach to addressing the shortest path planning problem [22]. At its core, this method dissects the path planning challenge into manageable sub-problems, leveraging the optimal solutions from these sub-problems to construct an overarching optimal path [23]. The Baidu Apollo autonomous vehicle employs dynamic programming, rooted in EM planning and lattice planning, both well-established within the literature [24]. Song et al. introduced an innovative dynamic programming approach for computing optimal paths. This approach harnesses Dijkstra's algorithm and expands the number of optimal path error correction regions [25]. Ren et al. presented a path-planning methodology grounded in dynamic programming principles. It aims to identify an exact globally optimal solution by discretizing the continuous state space into a grid [26]. The adaptability of dynamic programming to complex constraints within path planning, coupled with its

high scalability, positions it as a suitable choice for intricate path-planning tasks [27].

This paper introduces a novel quadratic path planning methodology, which involves the discretization of sampling points and the enhancement of cost functions to tackle real-time and stability challenges in the context of local path planning for autonomous vehicles. The contributions of this study are outlined as follows:

1) This article introduces an innovative method for calculating an optimized reference line, formulating evaluation criteria, and imposing constraint conditions to achieve optimized reference lines. Additionally, a path planning algorithm is developed based on these optimized reference lines, aiming to enhance the overall smoothness of the path.

2) The paper introduces a quadratic programming algorithm designed to handle nonlinear constraint conditions and optimize objective functions for path planning. This algorithm integrates the principles of convex obstacle avoidance with a discretized drivable area, resulting in enhanced efficiency and security within the path planning process.

3) To mitigate the influence of speed on path planning, a metric for assessing speed quality is introduced, and the speed undergoes two optimization phases. The findings indicate that rational speed curves can be strategically planned across diverse driving scenarios, thereby enhancing the resilience of the path.

The subsequent sections of this paper are structured as follows: Section II furnishes comprehensive information regarding the implementation of coordinate transformation between the Frenet and Cartesian coordinate systems, placing a particular emphasis on optimized reference lines. Section III delineates the formulation of the cost function and the establishment of boundary conditions necessary for generating the optimal path using quadratic programming applied to the sampling points. Section IV concentrates on the derivation of optimal velocities tailored to the planned paths. Section V offers an exposition of the simulation results. Section VI serves as the conclusion of this paper.

II. FRENET-CARTESIAN COORDINATE TRANSFORMATION

The Frenet coordinate system is widely applied in path planning, enabling the distinct consideration of a vehicle's lateral and longitudinal movements. This separation allows for distinct lateral and longitudinal control, leading to increased adaptability to road geometries and the generation of smooth, secure trajectories across diverse road conditions [3].

A. REFERENCE LINE OPTIMIZATION

Considering the challenges posed by excessively long paths for coordinate transformation between Cartesian and Frenet coordinate systems, and the associated difficulties in locating the vehicle's projection point on the reference line, it becomes imperative to address these issues to ensure path smoothness. To achieve this, enhancements to the

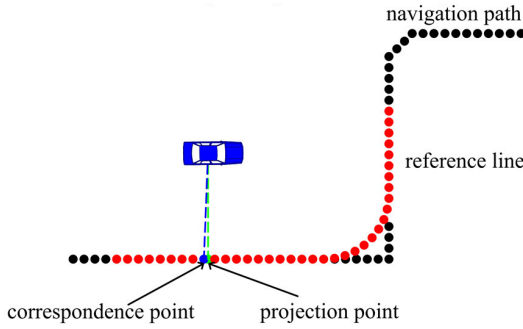


FIGURE 1. Discretization and smoothing of the reference line.

reference line are deemed necessary. In each planning cycle, the vehicle’s projection point on the global path is initially determined. Subsequently, the nearest discrete point to this projection point is designated as the coordinate origin (x, y) . A 120-meter span is considered forward, and a 30-meter span is considered backward, serving as the foundation for the reference line. The design cost function is then applied to refine and optimize this foundational reference line, resulting in a trajectory formed by the processed point set, which serves as the reference line. Figure 1 provides a visual representation of this process: the black point represents the global path, the green point signifies the vehicle’s projection point on the reference line, the blue point indicates the nearest sampling point to the same location, and the red point symbolizes the optimized point.

This paper introduces three distinct cost functions designed to assess the quality of the reference line. In Figure 2(a), we observe $P_0, P_1,$ and P_2 as the three consecutive path points, with $|P_1P_3|$ serving as a measure of the reference line’s smoothness. Smaller $|P_1P_3|$ values correspond to a smoother reference line. Ensuring a minimal gap between the optimized reference line and the navigation path is crucial, as illustrated in Figure 2(b). The blue straight line represents the optimized path, and $|P_1P_{1r}| + |P_2P_{2r}| + |P_3P_{3r}|$ serves as a metric for assessing geometrical similarity to the original path. Reduced PIP2 values indicate a closer resemblance to the original path. Furthermore, achieving uniform spacing between two path points on the optimized reference line is essential, as depicted in Figure 2(c) by the dark straight line. $|P_1P_2|^2 + |P_2P_3|^2$ is employed as a metric for evaluating the uniformity of the reference line’s path. Smaller $|P_1P_2|^2 + |P_2P_3|^2$ values signify a more uniform path.

In summary, we formulate the reference line cost function, where $[(x_{1r}, y_{1r}), (x_{2r}, y_{2r}), \dots, (x_{nr}, y_{nr})]$ represents the original navigation path points $(P_{1r}, P_{2r}, \dots, P_{nr})$, and $[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$ signifies the reference line path points (P_1, P_2, \dots, P_n) for the desired solution. The reference line cost function is thus defined and computed as follows:

$$Cost_{Function} = \omega_{smoothCost} \left\{ \sum_{i=1}^{n-2} [(x_i + x_{i+2} - 2x_{i+1})^2 + (y_i + y_{i+2} - 2y_{i+1})^2] \right\}$$

$$+ \omega_{simCost} \left\{ \sum_{i=1}^n [(x_i - x_{ir})^2 + (y_i - y_{ir})^2] \right\} + \omega_{compCost} \left\{ \sum_{i=1}^{n-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2] \right\} \quad (1)$$

these are denoted as $\omega_{smoothCost}, \omega_{simCost},$ and $\omega_{compCost},$ representing the cost weights for their respective terms. Let’s formulate this into a quadratic programming optimal solution problem, which is typically represented as follows [30]:

$$J = \frac{1}{2}x^T Hx + f^T x \text{ s.t. } Ax \leq b, lb \leq x \leq ub \quad (2)$$

The path smoothing cost is calculated as follows:

$$Cost_{smooth} = \sum_{i=1}^{n-2} [(x_i + x_{i+2} - 2x_{i+1})^2 + (y_i + y_{i+2} - 2y_{i+1})^2] = (x_1, y_1, \dots, x_n, y_n) \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ -2 & 0 & 1 & 0 & \dots \\ 0 & -2 & 0 & 1 & \dots \\ 1 & 0 & -2 & 0 & \dots \\ 0 & 1 & 0 & -2 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}_{2n \times 2n-4} \times \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ -2 & 0 & 1 & 0 & \dots \\ 0 & -2 & 0 & 1 & \dots \\ 1 & 0 & -2 & 0 & \dots \\ 0 & 1 & 0 & -2 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}_{2n \times 2n-4} (x_1, y_1, \dots, x_n, y_n)^T \quad (3)$$

remember that the $2n \times 2n - 4$ matrix in Eq. is $A_1,$ so the smoothing cost is $\omega_{smoothCost} x^T A_1^T A_1 x,$ where $x = (x_1, y_1, \dots, x_n, y_n)^T.$

The path similarity cost is calculated as follows:

$$Cost_{sim} = \sum_{i=1}^n [(x_i - x_{ir})^2 + (y_i - y_{ir})^2] = (x_1, y_1, \dots, x_n, y_n) \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}_{2n \times 2n} \times (x_1, y_1, \dots, x_n, y_n)^T - 2(x_{1r}, y_{1r}, \dots, x_{nr}, y_{nr})(x_1, y_1, \dots, x_n, y_n)^T \quad (4)$$

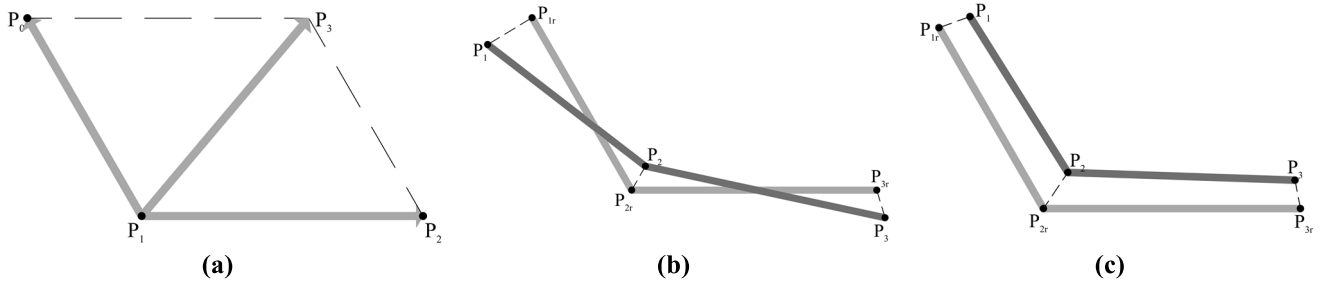


FIGURE 2. Measurement of the merits of the reference line. (a) The standard for measuring the smoothness of the reference line, (b) Criteria for measuring the similarity of reference lines, (c) Standard for measuring the uniformity of reference line.

where the $2n \times 2n$ matrix is the unit matrix, denoted as A_2 , and hence the similarity cost is $\omega_{simCost} x^T A_2^T A_2 x + h^T x \left(\frac{\pi}{2} - \theta\right)$, where $h = (-2x_{1r}, -2y_{1r}, \dots, -2x_{nr}, -2y_{nr})^T$.

The uniform cost of the path is calculated as follows:

$$\begin{aligned}
 & Cost_{comp} \\
 &= \sum_{i=1}^{n-1} \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right] \\
 &= (x_1, y_1, \dots, x_n, y_n) \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ -1 & 0 & 1 & 0 & \dots \\ 0 & -1 & 0 & 1 & \dots \\ 0 & 0 & -1 & 0 & \dots \\ 0 & 0 & 0 & -1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}_{2n \times 2n-2} \\
 &\quad \times \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ -1 & 0 & 1 & 0 & \dots \\ 0 & -1 & 0 & 1 & \dots \\ 0 & 0 & -1 & 0 & \dots \\ 0 & 0 & 0 & -1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}^T (x_1, y_1, \dots, x_n, y_n)^T
 \end{aligned} \tag{5}$$

remember that the $2n \times 2n - 2$ matrix in Eq. is A_3 , so the smoothing cost is $\omega_{compCost} x^T A_3^T A_3 x$.

In summary, the reference line cost is:

$$\begin{aligned}
 & cost_{Function} \\
 &= x^T \left(\omega_{smoothCost} A_1^T A_1 + \omega_{simCost} A_2^T A_2 + \omega_{compCost} A_3^T A_3 \right) x \\
 &\quad + \omega_{simCost} h^T x
 \end{aligned} \tag{6}$$

the weight matrix for the corresponding quadratic programming is $H = 2(\omega_{smoothCost} A_1^T A_1 + \omega_{simCost} A_2^T A_2 + \omega_{compCost} A_3^T A_3)$, $f^T = \omega_{simCost} h^T$, noting $x = (x_1, y_1, \dots, x_n, y_n)^T$ and $x_{sim} = (x_{1r}, y_{1r}, \dots, x_{nr}, y_{nr})^T$, and the constraints are $x - x_{sim} = K$, where K is a constant.

B. COORDINATE CONVERSION

In Cartesian coordinates, the vehicle's state of motion is conventionally represented as $[\vec{r}_v, \vec{v}_v, \vec{a}_v, \theta_v, k_v]$ [31], where \vec{r}_v signifies the current position vector, \vec{v}_v the current speed,

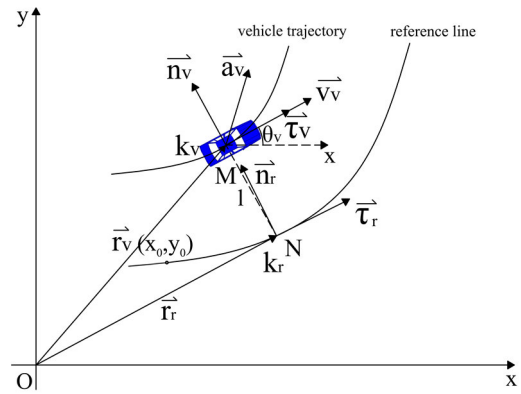


FIGURE 3. The Frenet-Cartesian transformation relation.

\vec{a}_v the current acceleration, θ_v the azimuth angle, and k_v the curvature of the vehicle's position vector along the trajectory. As illustrated in Figure 3, the vehicle's position is denoted by point M. When projecting the vehicle onto the reference line, it results in the projection point N. In contrast, in the Frenet coordinate system, the vehicle's state of motion is typically described as $[s, \dot{s}, \ddot{s}, l, \dot{l}, \ddot{l}, l', l'']$ [32]. Here, s represents the longitudinal displacement, \dot{s} the velocity in the direction of the tangent to the reference line, \ddot{s} the acceleration in the direction of the tangent, l the transverse displacement, \dot{l} the velocity in the direction of the normal to the reference line, \ddot{l} the acceleration in the direction of the normal, l' the first-order derivative of transverse displacement with respect to longitudinal displacement, and l'' the second-order derivative of transverse displacement concerning longitudinal displacement.

Utilizing (x_0, y_0) as the origin for the reference line and uniformly discretizing the reference line, the longitudinal displacement, represented by s , is approximated through a series of straight-line segments when the discretization points are adequately dense.

$$s = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{7}$$

Referring to Figure 3, the transformation equation governing the vehicle's motion state in both Cartesian and Frenet

coordinate systems can be derived [31].

$$\begin{cases} l = (\vec{r}_v - \vec{r}_r) \cdot \vec{n}_r \\ \dot{s} = \frac{\vec{v}_v \cdot \vec{\tau}_r}{1 - k_r l} \\ \dot{l} = \vec{v}_v \cdot \vec{n}_r \\ \ddot{s} = \frac{\vec{a}_v \cdot \vec{\tau}_r + \dot{s}^2 k_r l' + (k_r' l + k_r l') \cdot \dot{s}^2}{1 - k_r l} \\ \ddot{l} = \vec{a}_v \cdot \vec{n}_r - \dot{s}^2 k_r (1 - k_r l) \\ l' = \frac{\vec{v}_v \cdot \vec{n}_r}{1 - k_r l} \\ l'' = \frac{\vec{v}_v \cdot \vec{\tau}_r}{\dot{s}^2} \end{cases} \quad (8)$$

In this context, \vec{r}_r represents the position vector of the vehicle at the projection point, \vec{n}_r denotes the normal unit vector of the position vector for the vehicle's projection on the reference line, $\vec{\tau}_r$ signifies the tangential unit vector of the position vector for the vehicle's projection on the reference line, and k_r represents the curvature of the position vector for the vehicle's projection on the reference line.

III. PATH PLANNING BASED ON QUADRATIC PROGRAMMING OF SAMPLING POINTS

Vehicles must adhere to traffic regulations while traveling safely and smoothly along their planned route [33]. To achieve this, the process begins by discretizing all drivable positions into a set of sampling points. Subsequently, a heuristic search is conducted among these sampling points to identify a preliminary obstacle avoidance path. This path serves as the foundation for establishing a convex feasible space, thereby transforming the path planning problem into a convex problem. Quadratic programming is then applied within this convex feasible space to define the cost function and constraints, ensuring the path's smoothness and stability. The method presented in this paper generates sampling points based on different driving conditions, as illustrated in Figure 4, depicting the transition pattern between various scenarios.

A. GENERATE SAMPLING POINTS

The sampling points result from discretizing all the drivable locations of the vehicle. Two distinct methods for generating these sampling points are established, depending on the environmental conditions in which the vehicle is operating. The coordinates for generating these sampling points in the designated area must satisfy the following conditions:

$$\begin{cases} s_c = s_o + v_0 t + \frac{1}{2} a_0 t^2 \\ l_c = l_o \end{cases} \quad (9)$$

where s_o and l_o are the vehicle start coordinates, s_c and l_c are coordinates of the sampling point, v_0 is the vehicle's starting velocity, and a_0 is the vehicle's starting acceleration.

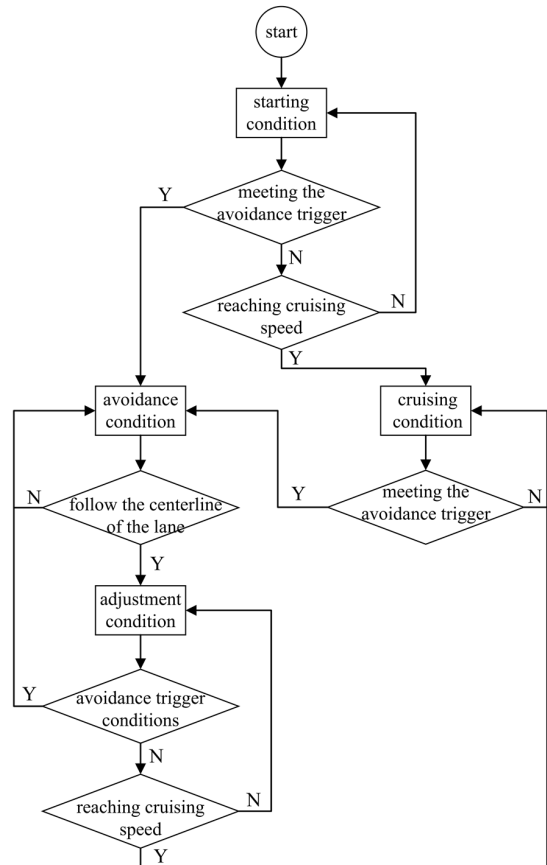


FIGURE 4. Conversion modes for various scenarios.

Generating sampling points should satisfy the following equation:

$$\begin{cases} s_{ci} = s_{ulim} + 0.1i |s_{ulim}| \leq s_{ci} \leq |s_{rlim}|, & i = 1, 2, \dots, m \\ l_{cj} = l_{ulim} + 0.1j |l_{ulim}| \leq l_{cj} \leq |l_{rlim}|, & j = 1, 2, \dots, n \end{cases} \quad (10)$$

where s_{ci} and l_{cj} are the coordinates of the sampling point, l_{rlim} and l_{ulim} are the topmost and bottom of the region, s_{rlim} and s_{ulim} are the leftmost and rightmost of the region, respectively, and s_{rlim} and s_{ulim} can be calculated by the following equation:

$$\begin{cases} s_{rlim} = s_o + v_0 t + \frac{1}{2} a_t t^2 \\ s_{ulim} = s_o + v_0 t + \frac{1}{2} a_u t^2 \end{cases} \quad (11)$$

where a_t is the maximum limited acceleration, and a_u is the minimum limited acceleration. In situations with no obstacles, only a single sampling point is required. Therefore, optimization is focused solely on selecting the optimal sampling point for obstacle avoidance conditions, as indicated in Figure 5.

B. CONVEX OPTIMIZATION

During vehicle travel, multiple options for overtaking a moving obstacle vehicle may lead to non-convex optimization

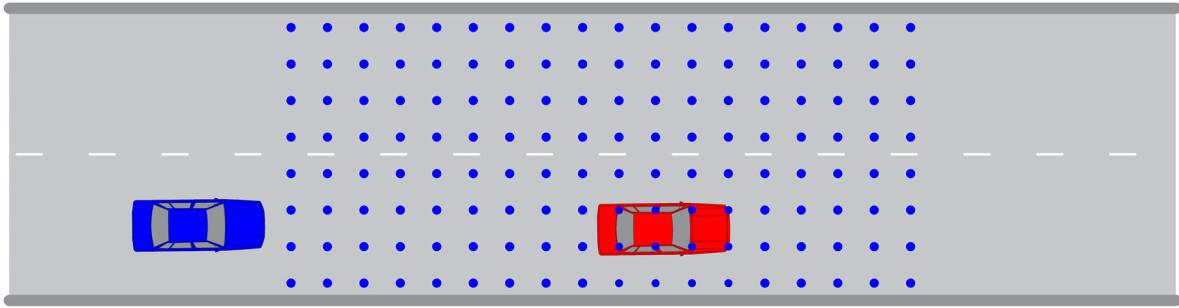


FIGURE 5. Generate candidate sampling points in the driving area ahead.

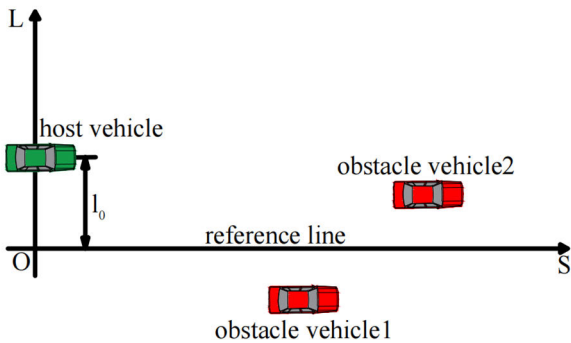


FIGURE 6. Vehicle projection on S-L diagram.

challenges in trajectory planning. To address this issue and make it manageable, we employ convexification. This process transforms a control problem into a convex optimization problem, determines the appropriate lane-changing side, and identifies the optimal path for overtaking obstacles [34], [35]. In this paper, our approach to road convex optimization focuses on dynamic path planning in higher layers, rendering previously non-convex driving areas amenable to efficient problem-solving. Initially, we establish the Frenet coordinate system, using the vehicle's projection point on the reference line as the coordinate origin. Subsequently, we create the S-L diagram by projecting the vehicle and obstacles within the Frenet coordinate system, as demonstrated in Figure 6.

C. DETERMINING THE STARTING POINT FOR PLANNING

Upon the conclusion of path planning, it is not always feasible for the vehicle's control to precisely align with the planned path. As depicted in Figure 7, deviations from the ideal position are evident after the vehicle's trajectory tracking. Therefore, the vehicle's localization point is not used as the starting point for subsequent path planning.

The process begins by projecting the vehicle's position onto the trajectory planned during the previous cycle, yielding the point P. Point P is then projected onto the reference line, providing coordinates (s_0, l_0) . These coordinates (s_0, l_0) are subsequently used as the starting point for path planning, thus contributing to the continuity and smoothness of the planned trajectory.

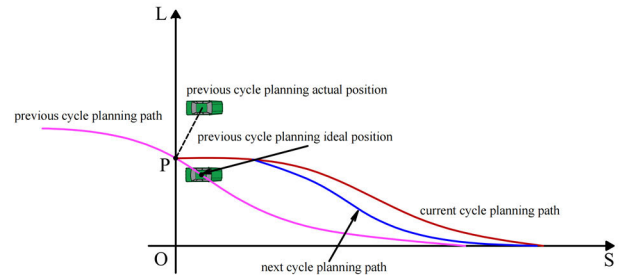


FIGURE 7. Define the starting point for path planning.

D. VEHICLE TRAJECTORY CONSTRAINTS

To connect the sampling points effectively, a quintic spline curve is chosen. This quintic spline curve plays a vital role in ensuring that the path passes through the desired sampling point while adhering to specific constraints at both the starting and ending points. This approach results in a smooth trajectory, mitigating abrupt changes in vehicle acceleration and steering. Consequently, the trajectory closely aligns with the requirements of the road's curvature [36], [37]. In the Frenet coordinate system, the vehicle's trajectory information can be expressed as $l = f(s)$, where s denotes longitudinal displacement, and l signifies lateral displacement. By establishing the vehicle's initial position coordinates as (s_0, l_0) , the constraint at the starting point can be defined as: $f(s_0) = l_0, f'(s_0) = \tan \theta, f''(s_0) = 0$, where θ represents the vehicle's heading angle in the Cartesian coordinate system. As for the other sampling points, they are constrained to $f(s_{ij}) = l_{ij}, f'(s_{ij}) = 0, f''(s_{ij}) = 0$. Since a quintic spline curve is chosen, this introduces six unknown term coefficients, resulting in their general form [38].

$$f(s) = a_0 + a_1s + a_2s^2 + a_3s^3 + a_4s^4 + a_5s^5 \quad (12)$$

therefore, for a specific target sampling point $\Gamma(s_{ij}, l_{ij})$, the coefficients can be determined based on the vehicle's state at the previous sampling point $\mathbb{R}_{i-1,j-1} = [f(s_{i-1,j-1}), f'(s_{i-1,j-1}), f''(s_{i-1,j-1})]$ and the state at the target sampling point $\mathbb{R}_{i,j} = [f(s_{i,j}), f'(s_{i,j}), f''(s_{i,j})]$.

E. DESIGNING DYNAMIC PATH COST FUNCTIONS

Following the generation of a set of trajectories, the subsequent crucial step involves identifying the optimal trajectory

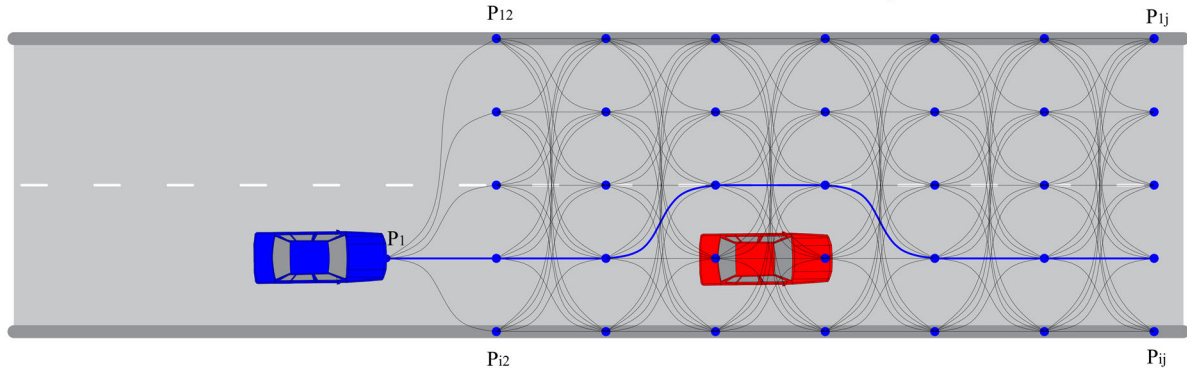


FIGURE 8. Obstacle avoidance least-cost path selection.

path planning procedure integrates the vehicle's dimensions, as depicted in Figure 10, to ensure efficient obstacle avoidance.

In the provided equation, several key parameters are defined. $P_1, P_2, P_3,$ and P_4 correspond to the four vertices of the vehicle. Additionally, we introduce d_1 , signifying the distance from the center of mass to the vehicle's front, and d_2 , representing the distance from the center of mass to the vehicle's rear. The vehicle's width is denoted as w , and θ indicates the angle between the vehicle's travel direction and the reference line. Lastly, l represents the lateral distance between the center of mass and the reference line. With these parameters established, the calculation of the lateral coordinates of the vehicle's four vertices is as follows:

$$\begin{cases} l_{P_1} = l + d_1 \sin \theta + \frac{w}{2} \cos \theta \\ l_{P_2} = l + d_1 \sin \theta - \frac{w}{2} \cos \theta \\ l_{P_3} = l - d_2 \sin \theta + \frac{w}{2} \cos \theta \\ l_{P_4} = l - d_2 \sin \theta - \frac{w}{2} \cos \theta \end{cases} \quad (17)$$

Considering that trigonometric functions are nonlinear, the above equation is simplified by expressing $\sin \theta \approx \tan \theta \approx l'$ and $\cos \theta \approx 1$. The objective is to find the minimum value of l_{max_i} in the interval $(s_i - d_2, s_i + d_1)$, denoted as ub_i , such that $l_{P_1}, l_{P_2}, l_{P_3}, l_{P_4} \leq ub_i$, and to find the maximum value of l_{mini} in the same interval, denoted as lb_i , such that $l_{P_1}, l_{P_2}, l_{P_3}, l_{P_4} \geq lb_i$. Thus, the obtained constraints are converted to matrix form as follows:

$$\begin{pmatrix} 1 & d_1 & 0 \\ 1 & d_1 & 0 \\ 1 & -d_2 & 0 \\ 1 & -d_2 & 0 \\ -1 & -d_1 & 0 \\ -1 & -d_1 & 0 \\ -1 & d_2 & 0 \\ -1 & d_2 & 0 \end{pmatrix} \begin{pmatrix} l_i \\ l'_i \\ l''_i \end{pmatrix} \leq \begin{pmatrix} ub_i - \frac{w}{2} \\ ub_i + \frac{w}{2} \\ ub_i - \frac{w}{2} \\ ub_i + \frac{w}{2} \\ -lb_i + \frac{w}{2} \\ -lb_i - \frac{w}{2} \\ -lb_i + \frac{w}{2} \\ -lb_i - \frac{w}{2} \end{pmatrix} \quad (18)$$

denoted as $A_{sub} \begin{pmatrix} l_i \\ l'_i \\ l''_i \end{pmatrix} \leq b_{sub_i}$. So the collision constraint is:

$$\begin{pmatrix} 0 & 0 & \cdots & 0 \\ A_{sub} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & A_{sub} \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix}_{8n \times 3n} \begin{pmatrix} l_1 \\ l'_1 \\ l''_1 \\ l_2 \\ l'_2 \\ l''_2 \\ \vdots \\ l_n \\ l'_n \\ l''_n \end{pmatrix}_{3n \times 1} \leq \begin{pmatrix} b_{sub1} \\ b_{sub2} \\ \vdots \\ b_{subn} \end{pmatrix} \quad (19)$$

To ensure the stability of the vehicle during path planning, the imposition of jerk constraints becomes crucial to guarantee the continuity of the curvature, ultimately resulting in a smoother and safer trajectory. Denote $l_i = f(s_i)$, $l'_i = f'(s_i)$, $l''_i = f''(s_i)$, and assume the third derivative of the curve $f(s)$ connecting l_i and l_{i+1} is constant $\frac{l''_{i+1} - l''_i}{\Delta s}$. Additionally, set the fourth derivative and all higher-order derivatives beyond the fourth to zero. The Taylor series expansion of l_{i+1} is carried out with $l_{i+1} = f(s_i + \Delta s)$, $l_i = f(s_i)$, leading to the collapsed expansion results:

$$l_{i+1} = l_i + l'_i \Delta s + \frac{1}{2} l''_i \Delta s^2 + \frac{1}{6} \frac{l''_{i+1} - l''_i}{\Delta s} \Delta s^3 \quad (20)$$

the derivation is written in matrix form as:

$$\begin{pmatrix} 1 & \Delta s & \frac{1}{2} \Delta s^2 & -1 & 0 & \frac{1}{6} \Delta s^2 \\ 0 & 1 & \frac{1}{2} \Delta s & 0 & -1 & \frac{1}{2} \Delta s \end{pmatrix} \begin{pmatrix} l_i \\ l'_i \\ l''_i \\ l_{i+1} \\ l'_{i+1} \\ l''_{i+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (21)$$

noting $A_{eq-sub} = \begin{pmatrix} 1 & \Delta s & \frac{1}{2} \Delta s^2 & -1 & 0 & \frac{1}{6} \Delta s^2 \\ 0 & 1 & \frac{1}{2} \Delta s & 0 & -1 & \frac{1}{2} \Delta s \end{pmatrix}$, and having $l_1, l'_1, l''_1, \dots, l_n, l'_n, l''_n$ to be optimized, the acceleration

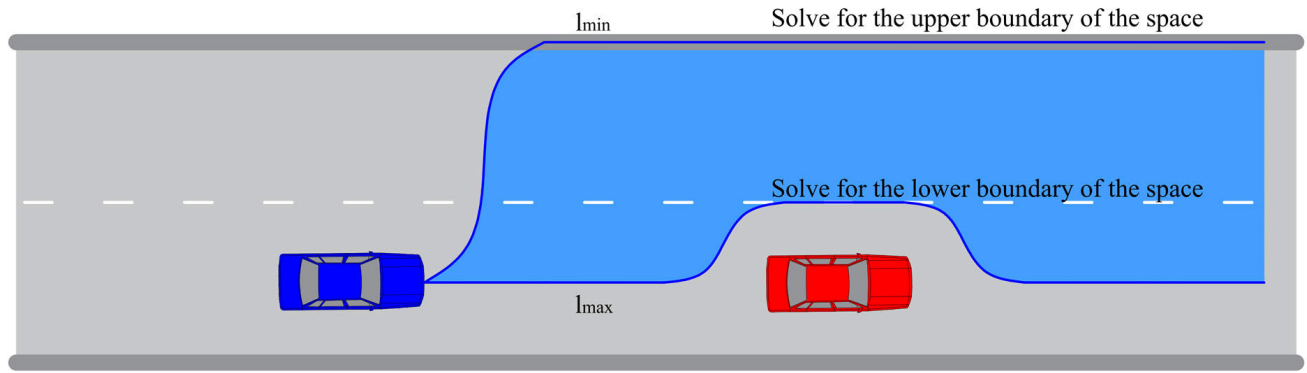


FIGURE 9. Generating Convex Spaces for Path Quadratic Programming.

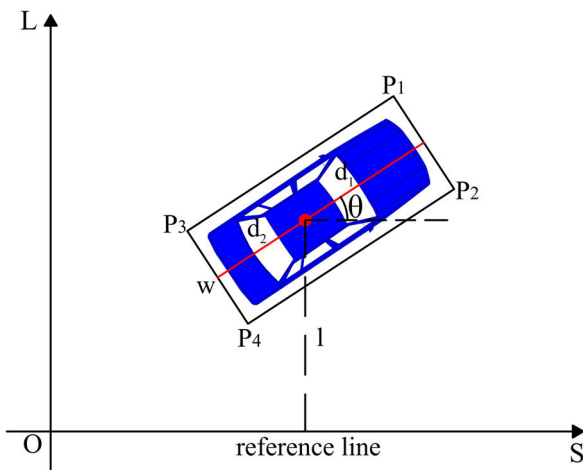


FIGURE 10. Volume expansion of vehicles.

constraint is added as:

$$\begin{pmatrix} A_{eq-sub} & 0 & \dots & \dots & 0 \\ 0 & A_{eq-sub} & 0 & \dots & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & \dots & 0 & A_{eq-sub} & 0 \\ 0 & \dots & \dots & 0 & A_{eq-sub} \end{pmatrix}_{2n-2 \times 3n} \begin{pmatrix} l_1 \\ l'_1 \\ l''_1 \\ l_2 \\ l'_2 \\ l''_2 \\ \vdots \\ l_n \\ l'_n \\ l''_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}_{2n-2 \times 1} \quad (22)$$

Denoted as $A_{eq}x = b_{eq}$, and in consideration of the ride comfort, we formulate the path smoothing cost function,

calculated as follows:

$$\begin{aligned} cost_{conf} = & \omega_{dl} \sum_{i=1}^n l'^2 + \omega_{ddl} \sum_{i=1}^n l''^2 \\ & + \omega_{ddd} \sum_{i=1}^n \left(\frac{l'''_{i+1} - l'''_i}{ds} \right)^2 \end{aligned} \quad (23)$$

where l' , l'' , and $\frac{l'''_{i+1} - l'''_i}{ds}$ represent the first derivative, second derivative, and third derivative of the trajectory's transverse displacement concerning the longitudinal displacement, respectively. The larger their sum, the more extensive the transverse displacements under the same longitudinal displacements, which is less favorable for ride comfort. ω_{dl} , ω_{ddl} , and ω_{ddd} denote the respective weights assigned to these costs.

Vehicles are advised to maintain proximity to the centerline to minimize the impact of vehicles in adjacent lanes. The cost associated with centerline distance in establishing a convex space is calculated as follows:

$$cost_{mid} = \omega_{mid} \sum_{i=1}^n \left(l_i - \frac{l_{maxi} + l_{mini}}{2} \right)^2 \quad (24)$$

where n denotes the number of discrete points; l_i represents the lateral displacement of the vehicle in the Frenet coordinate system; l_{maxi} and l_{mini} denote the lateral coordinates outlining the boundaries of the convex space; and ω_{mid} signifies the weight coefficient for the offset convex space centerline cost. The total cost function is defined as the sum of two distinct cost functions.

$$costFunction = cost_{conf} + cost_{mid} \quad (25)$$

When $l_i = \frac{l_{maxi} + l_{mini}}{2}$, the vehicle travels along the centerline of the road, and at this point $costFunction$ is 0. Under the constraint of $A_{eq}x = b_{eq}, Ax \leq b, lb \leq x \leq ub, x = (l_1, l'_1, l''_1, \dots, l_n, l'_n, l''_n)^T$, solve $l_1, l'_1, l''_1, \dots, l_n, l'_n, l''_n$ and s_1, s_2, \dots, s_n to obtain the optimal path under quadratic programming, as shown in Figure 10. The green trajectory is the path after quadratic programming.

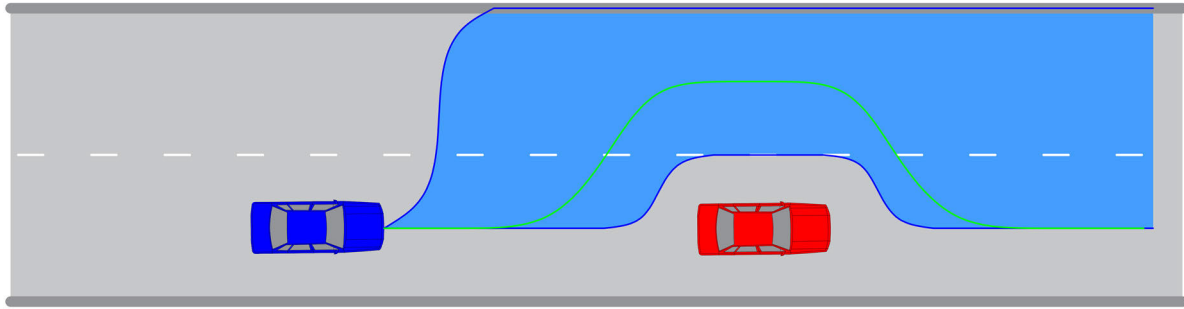


FIGURE 11. Path quadratic programming in convex spaces.

IV. CONSTRAINT-BASED SPEED PLANNING

When formulating a speed plan for a vehicle, it is crucial to consider both its smoothness and feasibility. This section introduces an S-T diagram to characterize the spatial state of obstacles and devise an optimal speed trajectory. The process initiates with predicting future trajectories for surrounding obstacles within a predefined time interval, denoted as t . Subsequently, a search graph is constructed within the S-T coordinate system, and the future positional data of these obstacles is projected onto the S-T graph. Following this, potential speed profiles within the search graph are examined, a convex space for speed optimization is established based on these profiles, and, ultimately, quadratic programming is employed to derive the optimal speed curve.

A. GENERATE S-T DIAGRAM

The process of speed planning involves determining the optimal speed for a vehicle as it traverses a predetermined path, a pivotal factor in meeting diverse criteria such as stability and efficiency. This section employs speed constraints derived from S-T plots to formulate the intended speed profile. The S-axis and T-axis correspond to prediction time and location, respectively. As depicted in Figure 12, the S-T diagram identifies obstacle objects within the blue region, with the obstacle avoidance speed profile crafted in the adjacent white area.

B. VEHICLE SPEED OPTIMIZATION

In the S-T plot, the velocity curve is modeled as a mapping function relative to both time and trajectory along the lane, denoted as $v_i = f(t_i, s_i)$, where t_i and s_i represent time and arc length along the path, respectively. To generate a feasible speed curve considering various constraint conditions, we propose a speed planning algorithm based on conditional constraints. The algorithm aims to minimize the objective functions related to the distance between the vehicle and obstacles, the reference speed of the vehicle, the reference acceleration, and the cost of the reference jerk. This is mathematically expressed by the following equation.

$$\text{costFunction} = \text{cost}_{obs} + \text{cost}_{ref-speed} + \text{cost}_{accel} + \text{cost}_{jerk} \quad (26)$$

the initial item denotes the distance cost between the vehicle and obstacles, the second item signifies the recommended speed cost of the vehicle, the third item indicates the acceleration cost of the vehicle, and the fourth item represents the jerk cost of the vehicle.

In the computation of the distance cost between vehicles and obstacles, it is imperative to sample the line connecting each point on the vehicle to the obstacle. Subsequently, the distance between each point and the obstacle is calculated, and the cumulative result of all these calculations yields the distance cost between the vehicle and the obstacle. Illustrated in Figure 13, P denotes the point under calculation, and the distance from the point to the line segment can only be one of the values: d_1, d_2 , or d_3 , where $d_1 = |\vec{v}_1|$, $d_2 = |\vec{v}_1 + \vec{v}_2|$, and $d_3 = \frac{|\vec{v}_1 \times \vec{v}_2|}{|\vec{v}_2|}$. It can be deduced that $d_{min} = \min(d_1, d_2)$ when $\vec{v}_1 \cdot \vec{v}_2$ and $\vec{v}_3 \cdot \vec{v}_2$ share the same sign; otherwise, $d_{min} = d_3$. Therefore, the obstacle distance cost is computed as follows:

$$\text{cost}_{obs} = \omega_{obs} \min(d_1, d_2, d_3) \quad (27)$$

the weighting factor for the obstacle distance cost, denoted as ω_{obs} , is assigned a value of 10^7 . Illustrated in Figure 14, the starting point is defined as $s = 0, T = 0, \dot{s} = |\vec{v}|, \ddot{s} = |\vec{a}|$. For the first column of discrete points $\dot{s} = \frac{s_1 - 0}{t_1 - 0}, \ddot{s} = \frac{\dot{s}_1 - |\vec{v}|}{t_1 - 0}, \text{jerk}_1 = \frac{\ddot{s}_1 - |\vec{a}|}{t_1 - 0}$ and the third column of discrete points $\dot{s} = \frac{s_3 - s_2}{t_3 - t_2}, \ddot{s} = \frac{\dot{s}_3 - \dot{s}_2}{t_3 - t_2}, \text{jerk}_3 = \frac{\ddot{s}_3 - \ddot{s}_2}{t_3 - t_2}$, the speed, acceleration, and jerk with the smallest cost obtained from the last planning calculation are retained as the corresponding parameters for the current point. Subsequently, the reference speed cost, acceleration cost, and added jerk cost for connecting the two points are calculated as follows:

$$\begin{cases} \text{cost}_{ref-speed} = \omega_{ref-speed} (\dot{s}_{ij} - v_r)^2 \\ \text{cost}_{accel} = \omega_{accel} \ddot{s}_{ij}^2 \\ \text{cost}_{jerk} = \omega_{jerk} s_{ij}^2 \end{cases} \quad (28)$$

among these parameters, v_r denotes the anticipated speed of the vehicle, while $\omega_{ref-speed}, \omega_{accel}$, and ω_{jerk} represent the weighted coefficients associated with the cost of reference speed, vehicle acceleration, and vehicle jerk for the aforementioned vehicles. To modulate the degree of influence of different variables on speed, the assigned values are 4000, 100, and 10, respectively.

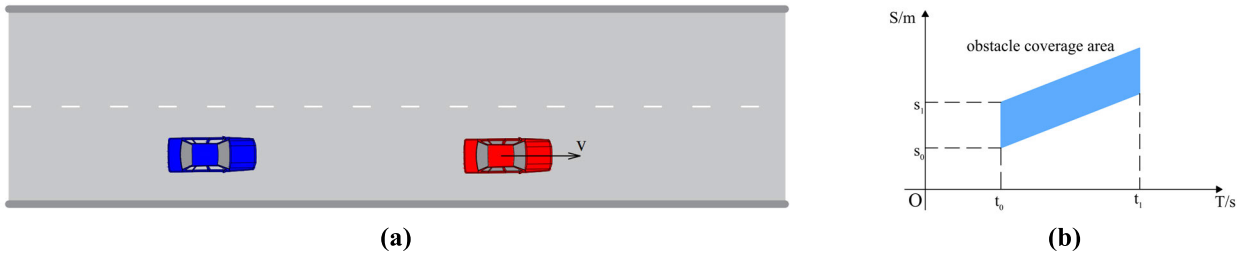


FIGURE 12. Projecting obstacles onto the S-T diagram.

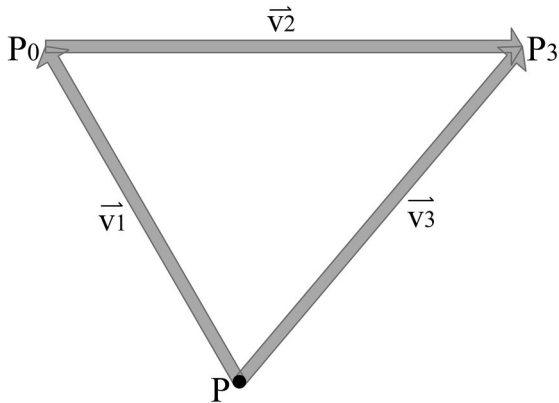


FIGURE 13. Methodology for calculating collision cost.

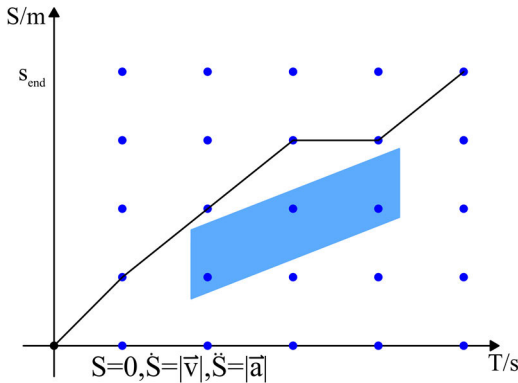


FIGURE 14. Dynamic planning of obstacle avoidance speed.

The utilization of dynamic programming to tackle velocity-related aspects streamlines the solution space of the problem. Illustrated in Figure 15 is the updated S-T diagram, an outcome derived from the dynamic programming outcomes. The subsequent quadratic programming convex space is formed utilizing these dynamic programming results, and the resultant blue curve on the diagram represents the velocity curve attained post-quadratic programming. In order to adhere to kinematic criteria, constraints are defined for speed, acceleration, and jerk at the nodes, as outlined below:

$$\begin{cases} s_i \leq s_{i+1} \\ 0 \leq \dot{s}_i \leq v_{lim} \\ a_{min} \leq \ddot{s}_i \leq a_{max} \\ s_i \leq jerk_{max} \end{cases} \quad (29)$$

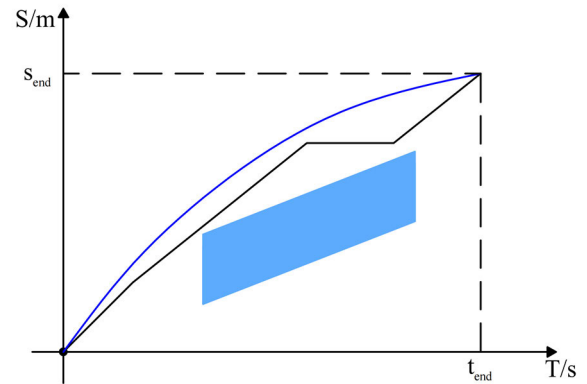


FIGURE 15. Quadratic programming to obtain the optimal speed profile.

This necessitates that the longitudinal displacement, denoted as s at each moment should not surpass that of the subsequent moment. Here, $jerk_{max}$ represents the maximum longitudinal jerk, v_{lim} stands for the velocity limit, while a_{min} and a_{max} correspond to the minimum and maximum travel accelerations, respectively.

$$\begin{cases} v_{lim} = \min(v_{kappa}, v_{road}) \\ v_{kappa} = \sqrt{\frac{a_{ymax}}{kappa}} \end{cases} \quad (30)$$

where v_{kappa} and v_{road} represent the speed limits imposed by road curvature and road regulations, respectively. The variable $kappa$ signifies the road curvature, and a_{ymax} denotes the maximum transverse acceleration. By incorporating these nonlinear constraints, the formulation is expressed as:

$$min_g(\zeta) \text{ s.t. } h_i(\zeta) \geq 0, i = 1, 2, \dots, n \quad (31)$$

where ζ represents the variable to be optimized, $h_i(\zeta)$ is the conditional constraint, and $g(\zeta)$ is the objective function. The optimization of the variable $s = [s_1, s_2, \dots, s_n]^T$ is computed as follows:

$$\begin{cases} s_{i+1} = s_i + \dot{s}_i dt + \frac{1}{2} \ddot{s}_i dt^2 + \frac{1}{6} \frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} dt^3 \\ \dot{s}_{i+1} = \dot{s}_i + \ddot{s}_i dt + \frac{1}{2} \frac{\ddot{s}_{i+1} - \ddot{s}_i}{dt} dt^2 \end{cases} \quad (32)$$

The outcome of solving the cost function, which adheres to all constraints, yields a continuous optimal velocity profile.

TABLE 1. Vehicle parameters and algorithm parameters.

| Parameter | Value (units) | Description |
|--------------|-----------------------|--|
| a_t | 2.5 m/s ² | Maximum limited acceleration |
| a_u | -3.0 m/s ² | Maximum limited deceleration |
| l_{ulim} | -3.5 m | The lower boundary of the sampling area |
| l_{tim} | 3.5 m | The upper boundary of the sampling area |
| m_1 | 10 m | Distance between vehicle and obstacle |
| m_2 | 0 m | Distance between vehicle and obstacle |
| k | -1000 | Rate of change of vehicle-obstacle collision cost |
| b | 10000 | The vehicle-obstacle collision cost constant |
| l_{ref} | 0 | Horizontal coordinates of the reference line |
| d_1 | 3.0 m | Expansion distance from the center of mass to the front of the vehicle |
| d_2 | 2.5 m | Expansion distance from the center of mass to the rear of the vehicle |
| w | 2.0 m | Width of the vehicle |
| v_r | 11.11 m/s | Desired speed of vehicle travel |
| v_{lim} | 16.67 m/s | Limited speed of vehicle travel |
| a_{min} | -4.5 m/s ² | Maximum longitudinal deceleration |
| a_{max} | 3.0 m/s ² | Maximum longitudinal acceleration |
| $jerk_{max}$ | 2.0 m/s ³ | Maximum longitudinal jerk |

The cost function is as follows:

$$\cos tFunction = \sum_{i=1}^n \omega_v (\dot{s}_i - v_r)^2 + \sum_{i=1}^n \omega_a \dot{s}_i^2 + \sum_{i=1}^n \omega_j s_i^2 \quad (33)$$

within this context, v_r is determined by the anticipated speed, while ω_v , ω_a , and ω_j represent the weight coefficients associated with the vehicle's driving speed, acceleration, and jerk, respectively. To modulate the degree of influence of different variables on the optimal speed, the assigned values are 10, 50 and 500, respectively. When $\dot{s}_i = v_r$, the vehicle speed attains the optimal value, and the speed cost $\cos tFunction$ becomes 0.

V. RESULTS

A. ANALOG PARAMETERS SETTING

To validate the viability of the method proposed in this paper, we established a simulation platform using Matlab/Simulink-Prescan-Carsim. The Prescan platform provided the scenario information, Matlab/Simulink housed the algorithms, and Carsim accommodated the imported vehicle models for joint simulation. The algorithm implementation occurred on a PC equipped with an Intel Core i5-10400 CPU, operating at 2.90 GHz, and 16 GB RAM. The simulation framework is depicted in Figure 16. Utilizing the described platform, diverse road scenes were designed for validation purposes. The tracking of the planned path involved employing an LQR controller for the horizontal direction, a dual PID controller for the vertical direction, and a bicycle model to represent the vehicle's dynamic response across various scenarios. Table 1 outlines the relevant parameters.

In Figures 17-20, the planning path is denoted by the blue solid curve, while the lane boundary line and lane centerline are represented by the black solid and dashed lines, respectively. For a clearer assessment of the planning method's reliability, the positions of the self-vehicle and obstacle

vehicles are marked at intervals of 4.5 seconds. The straightaway scenario, as shown in Figures 17-19, features a blue rectangle indicating the self-vehicle and the remaining area designating the obstacle vehicle. S1, F1, and L1 represent the initial positions of the self-vehicle and the obstacle vehicle. The curved scene, displayed in Figure 20, exhibits a blue circle for the self-vehicle and the remaining area for the obstacle vehicle. S, F, and L denote the starting locations of the self-vehicle and the obstacle vehicle, respectively. Additionally, we conduct a comparison of curvature and velocity with the dynamic programming-based method [24] (Method I) in the scenarios presented in Figures 17-20, aiming to further elucidate the advantages of the method proposed in this paper.

B. STRAIGHTAWAY SCENES

In Figure 17, there is an obstacle vehicle in the traveling lanes and the adjacent lane, vehicle F1 in front of it is moving forward at a velocity of 6 m/s from a location of $X_{F1} = 80m$, and vehicle L1 in the adjacent lane is moving forward at a velocity of 8 m/s from a location of $X_{L1} = 120m$. As shown by the blue solid line in Figure 17(a), the vehicle travels forward, detects the vehicle F1 in front of it, accelerates to travel in the adjacent lane until it detects the vehicle L1, then decelerates to the original lane, where it decelerates to the target speed. The analysis of speed and curvature in Figure 17 (b)(c) compared with Method I reveals that the absence of robust speed smoothness control in Method I results in abrupt speed fluctuations during the detection of obstacle vehicles. Notably, when maneuvering to avoid obstacles, the speed fluctuation becomes more pronounced, leading to a slower completion of avoidance maneuvers and reduced smoothness compared to the approach presented in this study. Concerning curvature, the method proposed in this paper demonstrates superior smoothness compared to Method I.

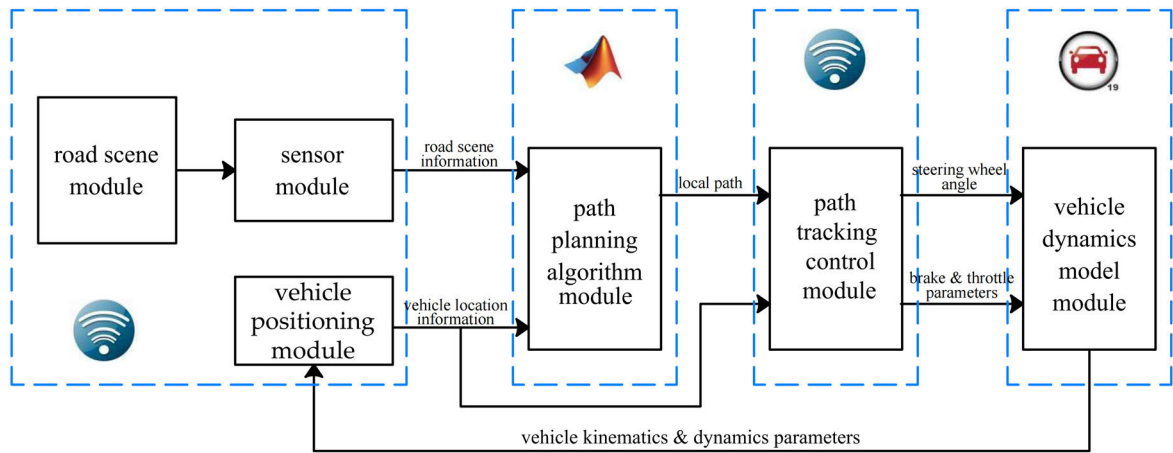


FIGURE 16. The platform framework of Matlab/Simulink-Prescan-Carsim.

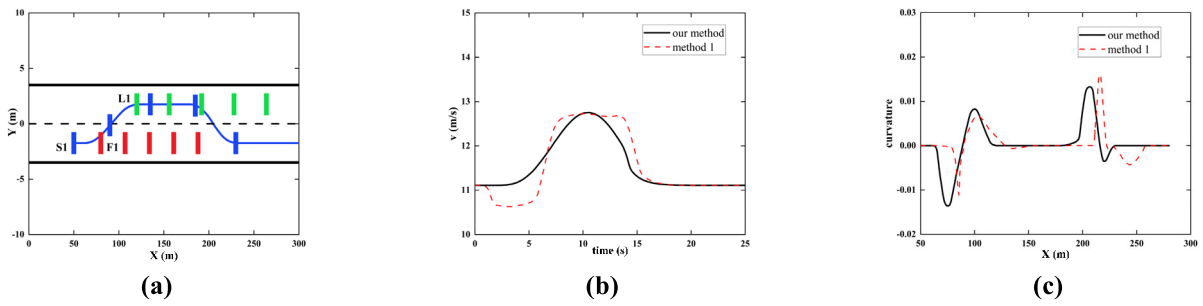


FIGURE 17. Continuous obstacle avoidance on straight roads. (a) Vehicle location and planning path, (b) Velocity curve, (c) Curvature curve.

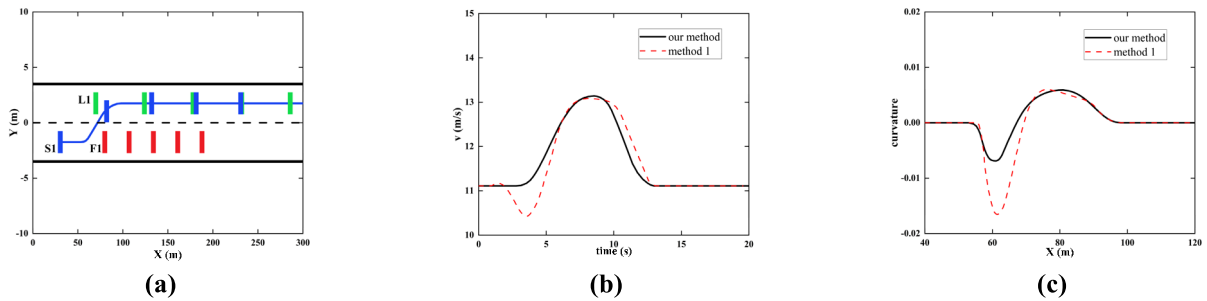


FIGURE 18. Direct lane change to avoid obstacles on straight roads. (a) Vehicle location and planning path, (b) Velocity curve, (c) Curvature curve.

In Figure 18, an obstacle vehicle is positioned in each of the traveling lanes and the adjacent lane. Vehicle F1, situated ahead, is moving forward at a speed of 6 m/s from a position of $X_{F1} = 80m$. In the adjacent lane, vehicle L1 is moving forward at a speed of 12 m/s from a position of $X_{L1} = 70m$, and the vehicle under observation is traveling on the road. The blue solid line in Figure 18(a) depicts the vehicle’s forward motion. It detects vehicle F1 in front, with a low obstacle distance cost from the vehicle in the adjacent lane, satisfying the vehicle obstacle avoidance condition. Consequently, the vehicle accelerates directly into the adjacent lane. An examination of the speed and curvature in Figures 18 (b)(c), compared

with Method I, reveals significant distinctions. Method I initiates a deceleration phase at the onset of a lane change to avoid obstacle vehicle F1. Subsequently, it accelerates the lane change, prolonging the temporal interaction with obstacle vehicle F1, thereby heightening the risk of accidents. This study introduces a velocity quadratic programming objective function to enhance the smoothness of speed transitions compared to Method I. Moreover, the curvature in this investigation exhibits superior smoothness.

In Figure 19, obstacle vehicles are present in each of the traveling lanes and the adjacent lane. Vehicle F1, positioned in front, is moving forward at a speed of 6 m/s from a

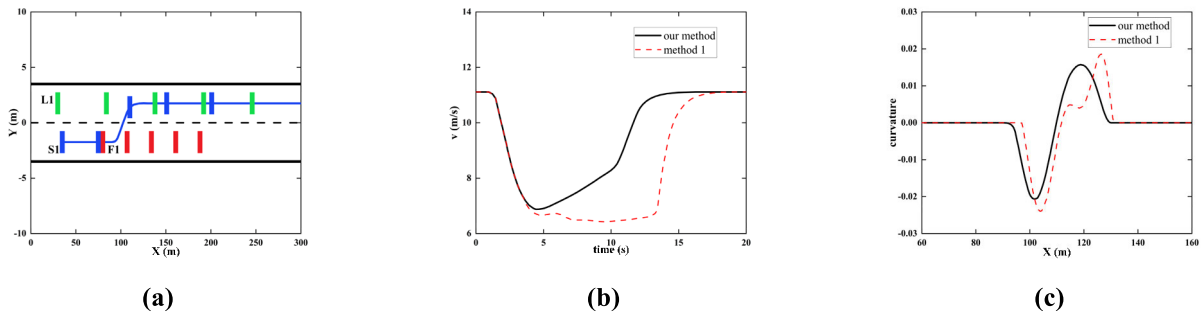


FIGURE 19. Slow down and change lanes to avoid obstacles on straight roads. (a) Vehicle location and planning path, (b) Velocity curve, (c) Curvature curve.

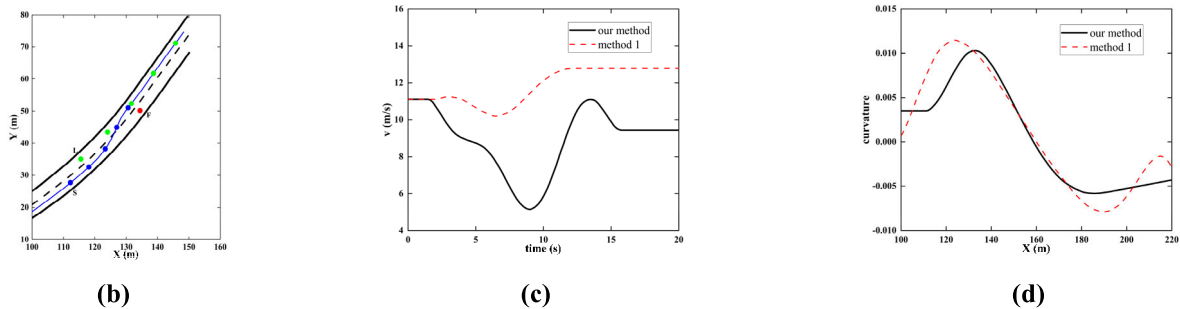
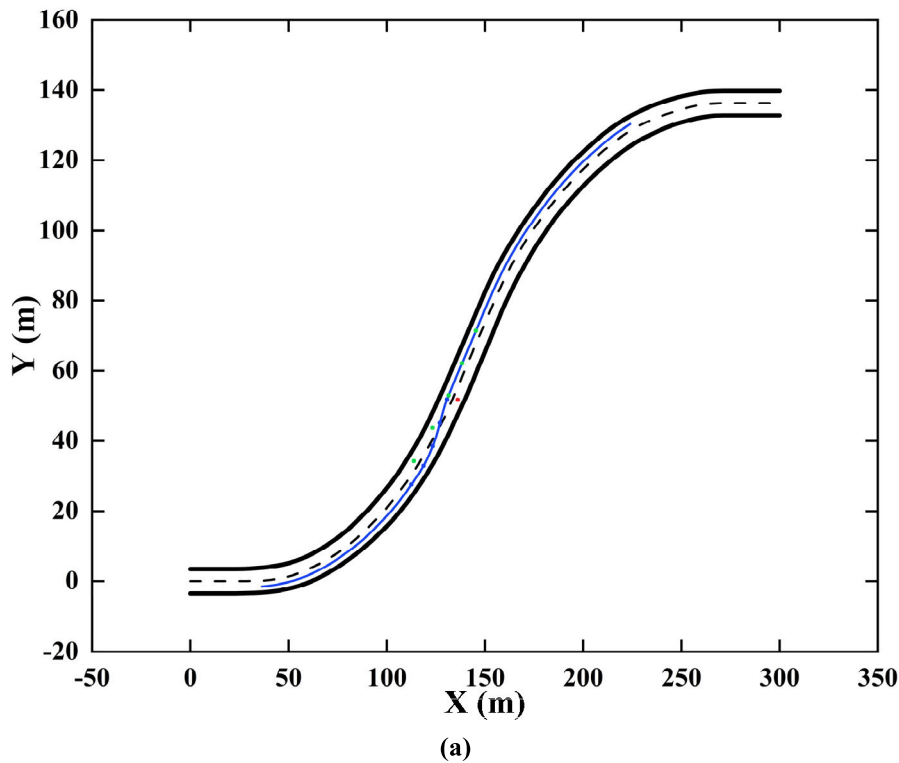


FIGURE 20. Lane change avoidance on S-curve road. (a) Vehicle location and planning path, (b) Zoom in on the local path, (c) Velocity curve, (d) Curvature curve.

location of $X_{F1} = 80\text{m}$. In the adjacent lane, vehicle L1 is moving forward at a speed of 12 m/s from a position of $X_{L1} = 30\text{m}$, and the observed vehicle is in motion on the road. Depicted by the blue solid line in Figure 19(a), the vehicle advances and detects vehicle F1 ahead. When the adjacent

lane fails to meet the vehicle obstacle avoidance condition, the vehicle initiates deceleration, continuing forward until the distance to the obstacle in the adjacent lane is both costly and low. Subsequently, the vehicle accelerates into the adjacent lane. A comparative analysis of the speed and curvature in

Figures 19 (b)(c), when contrasted with Method I, reveals that Method I trails obstacle vehicle F1 for an extended duration, resulting in reduced operational efficiency. This study introduces a collision objective function crafted to expedite lane-changing maneuvers when conditions permit. Furthermore, the curvature of Method I is less smooth than that achieved by the approach proposed in this paper.

C. S-CURVE SCENE

Figure 20 illustrates a driving scenario where a vehicle navigates around a stationary vehicle in front of it, encountering an obstacle vehicle on the S-curve road adjacent to the road. In this scenario, both the travel lane and the adjacent lane have obstacle vehicles, with Vehicle F ahead traveling at speed 0 and Vehicle L in the adjacent lane traveling at 15 m/s. Depicted by the blue solid line in Figure 20(a)(b), the vehicle progresses, detecting Vehicle F in front when the adjacent lane fails to meet the vehicle obstacle avoidance condition. The vehicle initially decelerates and continues forward until the distance to the obstacle in the adjacent lane becomes both costly and low. Subsequently, the vehicle accelerates into the adjacent lane. A comparative analysis of the speed and curvature in Figures 20 (c)(d), as opposed to Method I, reveals that Method I neglects the need to reduce the vehicle speed during curve navigation, persisting at an elevated speed post-obstacle avoidance. This oversight compromises vehicle stability, escalating the risk of accidents. Furthermore, in comparison with the methodology presented in this paper, Method I demonstrates increased curvature variation and diminished path stability.

VI. CONCLUSION

In this study, we propose a path planning methodology grounded in discrete optimization of sampled points and an enhanced cost function. The approach involves discretizing the obstacle avoidance space based on diverse driving environments, establishing nonlinear constraints, defining a convex space for vehicle obstacle avoidance, comprehensively considering the spatial relationship between the vehicle and obstacles, formulating an objective function, and determining the optimal obstacle avoidance path through quadratic programming within the convex space. Concurrently, a velocity metric is devised to plan an optimal velocity profile along the planned path, facilitating effective obstacle avoidance in various road scenarios. Test results affirm the adaptability of the approach to diverse driving environments and its ability to generate smooth paths. While the method demonstrates applicability to highways and urban roads, more intricate scenarios may necessitate the implementation of sophisticated cost-function-based path planning methods.

REFERENCES

[1] Y. Dingyi, W. Haiyan, and Y. Kaiming, "State-of-the-art and trends of autonomous driving technology," in *Proc. IEEE Int. Symp. Innov. Entrepreneurship (TEMS-ISIE)*. Beijing, China: IEEE, Mar. 2018, pp. 1–8, doi: 10.1109/TEMS-ISIE.2018.8478449.

[2] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, and X. Luo, "Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8546–8556, Sep. 2019, doi: 10.1109/TVT.2019.2930684.

[3] Y. Wang and Z. Lin, "Research on path planning for autonomous vehicle based on frenet system," *J. Eng. Res.*, vol. 11, no. 2, Jun. 2023, Art. no. 100080, doi: 10.1016/j.jer.2023.100080.

[4] B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, "A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving," *J. King Saud Univ-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7366–7390, Oct. 2022, doi: 10.1016/j.jksuci.2022.03.013.

[5] A. Puente-Castro, D. Rivero, E. Pedrosa, A. Pereira, N. Lau, and E. Fernandez-Blanco, "Q-learning based system for path planning with unmanned aerial vehicles swarms in obstacle environments," *Expert Syst. Appl.*, vol. 235, Jan. 2024, Art. no. 121240, doi: 10.1016/j.eswa.2023.121240.

[6] X. Wang, J. Lu, F. Ke, X. Wang, and W. Wang, "Research on AGV task path planning based on improved A* algorithm," *Virtual Reality Intell. Hardw.*, vol. 5, no. 3, pp. 249–265, Jun. 2023, doi: 10.1016/j.vrih.2022.11.002.

[7] F. Li, X. Fan, and Z. Hou, "A firefly algorithm with self-adaptive population size for global path planning of mobile robot," *IEEE Access*, vol. 8, pp. 168951–168964, 2020, doi: 10.1109/ACCESS.2020.3023999.

[8] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120254, doi: 10.1016/j.eswa.2023.120254.

[9] O. O. Martins, A. A. Adekunle, O. M. Olaniyan, and B. O. Bolaji, "An improved multi-objective a-star algorithm for path planning in a large workspace: Design, implementation, and evaluation," *Sci. Afr.*, vol. 15, Mar. 2022, Art. no. e01068, doi: 10.1016/j.sciaf.2021.e01068.

[10] R. Szczepanski, T. Tarczewski, and K. Erwinski, "Energy efficient local path planning algorithm based on predictive artificial potential field," *IEEE Access*, vol. 10, pp. 39729–39742, 2020, doi: 10.1109/ACCESS.2022.3166632.

[11] M. S. Das, S. Sanyal, and S. Mandal, "Navigation of multiple robots in formative manner in an unknown environment using artificial potential field based path planning algorithm," *Ain Shams Eng. J.*, vol. 13, no. 5, Sep. 2022, Art. no. 101675, doi: 10.1016/j.asej.2021.101675.

[12] J. Luo, Z.-X. Wang, and K.-L. Pan, "Reliable path planning algorithm based on improved artificial potential field method," *IEEE Access*, vol. 10, pp. 108276–108284, 2022, doi: 10.1109/ACCESS.2022.3212741.

[13] Z. Husain, A. Al Zaabi, H. Hildmann, F. Saffre, D. Ruta, and A. F. Isakovic, "Search and rescue in a maze-like environment with ant and Dijkstra algorithms," *Drones*, vol. 6, no. 10, p. 273, Sep. 2022, doi: 10.3390/drones6100273.

[14] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, p. 211, Mar. 2023, doi: 10.3390/drones7030211.

[15] S. Yang, Z. Shan, J. Cao, Y. Gao, Y. Guo, P. Wang, X. Wang, J. Wang, T. Zhang, and J. Guo, "Path planning of UAV base station based on deep reinforcement learning," *Proc. Comput. Sci.*, vol. 202, pp. 89–104, Jan. 2022, doi: 10.1016/j.procs.2022.04.013.

[16] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cogn. Robot.*, vol. 3, pp. 54–70, Apr. 2023, doi: 10.1016/j.cogr.2023.04.001.

[17] A. Sonny, S. R. Yeduri, and L. R. Cenkeramaddi, "Q-learning-based unmanned aerial vehicle path planning with dynamic obstacle avoidance," *Appl. Soft Comput.*, vol. 147, Nov. 2023, Art. no. 110773, doi: 10.1016/j.asoc.2023.110773.

[18] D. Schitz and H. Aschemann, "Path optimization for autonomous driving using deep learning," *IFAC-PapersOnLine*, vol. 55, no. 27, pp. 490–496, 2022, doi: 10.1016/j.ifacol.2022.10.560.

[19] Y. Yin, Z. Chen, G. Liu, and J. Guo, "A mapless local path planning approach using deep reinforcement learning framework," *Sensors*, vol. 23, no. 4, p. 2036, Feb. 2023, doi: 10.3390/s23042036.

[20] H. Hongyu, Z. Chi, S. Yuhuan, Z. Bin, and G. Fei, "An improved artificial potential field model considering vehicle velocity for autonomous driving," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 863–867, 2018, doi: 10.1016/j.ifacol.2018.10.095.

- [21] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.
- [22] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "HPPRM: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots," *IEEE Access*, vol. 8, pp. 221743–221766, 2020, doi: [10.1109/ACCESS.2020.3043333](https://doi.org/10.1109/ACCESS.2020.3043333).
- [23] C. Wu, S. Zhou, and L. Xiao, "Dynamic path planning based on improved ant colony algorithm in traffic congestion," *IEEE Access*, vol. 8, pp. 180773–180783, 2020, doi: [10.1109/ACCESS.2020.3028467](https://doi.org/10.1109/ACCESS.2020.3028467).
- [24] 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*.
- [25] M. Song, C. Ji, C. Wang, S. Li, N. Li, and F. Wang, "A novel dynamic programming based method for path planning with navigation error correction," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*. Chongqing, China: IEEE, Jun. 2020, pp. 111–117, doi: [10.1109/ITNEC48623.2020.9084672](https://doi.org/10.1109/ITNEC48623.2020.9084672).
- [26] J. Ren and X. Huang, "Dynamic programming inspired global optimal path planning for mobile robots," in *Proc. IEEE 4th Int. Conf. Inf. Syst. Comput. Aided Edu. (ICISCAE)*. Dalian, China: IEEE, Sep. 2021, pp. 461–465, doi: [10.1109/ICISCAE52414.2021.9590740](https://doi.org/10.1109/ICISCAE52414.2021.9590740).
- [27] R. Agrawal, B. Singh, and R. Kumar, "Classical approaches for mobile robot path planning: A review," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*. Greater Noida, India: IEEE, Nov. 2022, pp. 400–405, doi: [10.1109/ICCCIS56430.2022.10037620](https://doi.org/10.1109/ICCCIS56430.2022.10037620).
- [28] C. Park and S.-C. Kee, "Online local path planning on the campus environment for autonomous driving considering road constraints and multiple obstacles," *Appl. Sci.*, vol. 11, no. 9, p. 3909, Apr. 2021, doi: [10.3390/app11093909](https://doi.org/10.3390/app11093909).
- [29] L. Yu, X. Wang, Z. Hou, Z. Du, Y. Zeng, and Z. Mu, "Path planning optimization for driverless vehicle in parallel parking integrating radial basis function neural network," *Appl. Sci.*, vol. 11, no. 17, p. 8178, Sep. 2021, doi: [10.3390/app11178178](https://doi.org/10.3390/app11178178).
- [30] W. Li, M. Xia, and H. Li, "Some results on the upper bound of optimal values in interval convex quadratic programming," *J. Comput. Appl. Math.*, vol. 302, pp. 38–49, Aug. 2016, doi: [10.1016/j.cam.2016.01.044](https://doi.org/10.1016/j.cam.2016.01.044).
- [31] X. Xing, B. Zhao, C. Han, D. Ren, and H. Xia, "Vehicle motion planning with joint Cartesian–Frenét MPC," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10738–10745, Oct. 2022, doi: [10.1109/LRA.2022.3194330](https://doi.org/10.1109/LRA.2022.3194330).
- [32] H. Du, S. Leng, J. He, K. Xiong, and L. Zhou, "Digital twin empowered cooperative trajectory planning of platoon vehicles for collision avoidance with unexpected obstacles," *Digit. Commun. Netw.*, Jun. 2023, doi: [10.1016/j.dcan.2023.06.002](https://doi.org/10.1016/j.dcan.2023.06.002).
- [33] J. Pan, J. Wang, and S. Xing, "Dynamic trajectory planning and optimization for automated driving on ice and snow covered road," *IEEE Access*, vol. 11, pp. 36365–36378, 2023, doi: [10.1109/ACCESS.2023.3266006](https://doi.org/10.1109/ACCESS.2023.3266006).
- [34] Y. Zhang, H. Chen, S. L. Waslander, T. Yang, S. Zhang, G. Xiong, and K. Liu, "Speed planning for autonomous driving via convex optimization," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1089–1094, doi: [10.1109/ITSC.2018.8569414](https://doi.org/10.1109/ITSC.2018.8569414).
- [35] H. Haghshenas, A. Hansson, and M. Norrlöf, "Time-optimal path tracking for cooperative manipulators: A convex optimization approach," *Control Eng. Pract.*, vol. 140, Nov. 2023, Art. no. 105668, doi: [10.1016/j.conengprac.2023.105668](https://doi.org/10.1016/j.conengprac.2023.105668).
- [36] D. Zhao and W. Huang, "Improved path planning algorithm based on RRT algorithm and quintic B-spline curve," in *Proc. IEEE 11th Data Driven Control Learn. Syst. Conf. (DDCLS)*. Chengdu, China: IEEE, Aug. 2022, pp. 183–188, doi: [10.1109/DDCLS55054.2022.9858552](https://doi.org/10.1109/DDCLS55054.2022.9858552).
- [37] Z. Chen, X. Xu, W. Zha, A. Rodic, and P. B. Petrovic, "Motion planning of 7-DOF manipulator based on quintic B-spline curve," in *Proc. 6th IEEE Int. Conf. Adv. Robot. Mechatronics (ICARM)*. Chongqing, China: IEEE, Jul. 2021, pp. 858–862, doi: [10.1109/ICARM52023.2021.9536089](https://doi.org/10.1109/ICARM52023.2021.9536089).
- [38] J. Tan, Y. Xing, W. Fan, and P. Hong, "Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve," *J. Comput. Appl. Math.*, vol. 329, pp. 256–267, Feb. 2018, doi: [10.1016/j.cam.2017.07.007](https://doi.org/10.1016/j.cam.2017.07.007).



CHENGTAO ZHANG was born in Jiangsu, China, in 1978. He received the Ph.D. degree in vehicle engineering from China Agricultural University, in 2013. He is currently a Graduate Tutor with the School of Mechanical and Automotive Engineering, Guangxi University of Science and Technology, and the Director of the 8th Council of Guangxi Mechanical Engineering Society. His main research interests include vehicle intelligent control technology, in-vehicle embedded systems, ADAS algorithm simulation, and intelligent networked vehicles.



WEIHANG XU was born in Heilongjiang, China, in 2000. He is currently pursuing the master's degree in mechanical engineering with the Guangxi University of Science and Technology, Liuzhou, China. His main research interests include vehicle intelligent control technology.