

Research on automatic driving path optimization algorithm based on Frenet coordinate system and Cartesian coordinate system

Kaiqi Zhang

School of Mechanical Engineering
Nanjing University of Science and Technology
Nanjing, China
zhangkaiqi0319@163.com

Dejun Yin *

School of Mechanical Engineering
Nanjing University of Science and Technology
Nanjing, China
xhsxhs2000@163.com

Abstract—Under complex traffic conditions, autonomous driving has high requirements for the real-time and accuracy of path planning. Existing methods have shortcomings: the Frenet coordinate system simplifies calculations but is difficult to meet kinematic constraints, and the Cartesian coordinate system adapts to complex road conditions but is inefficient. To this end, this study proposes a dual-coordinate system collaborative path planning method, which first preliminarily solves the problem in the Frenet coordinate system and then optimizes it in the Cartesian coordinate system. Combining the gradient method with curve interpolation (GCI), the system considers kinematic constraints, dynamic obstacle avoidance, and trajectory smoothness, constructs an efficient iterative framework, and greatly improves the path planning performance.

Keywords—Autonomous driving Path Planning; Frenet; Cartesian; Gradient Curve Interpolation

I. INTRODUCTION

Autonomous driving technology has great potential in improving driving safety, comfort and efficiency, and the quality of trajectory planning is particularly critical^[1]. Existing methods are divided into road driving planning and parking planning, which focus on smooth obstacle avoidance and precise parking respectively. However, the continuous change of complex road curvature puts higher requirements on trajectory planning, and it is difficult to take into account both kinematic constraints and real-time performance. Methods based on the Frenet coordinate system are often used for trajectory planning. For example, Huang Jing et al. decomposed the vehicle kinematic constraints through longitudinal and transverse polynomials, but their performance was limited under high curvature curves^[2]. In contrast, methods based on the Cartesian coordinate system, such as Xu J, use virtual controllers to filter the trajectory generated by RRT, but their computational power is insufficient in modeling collision avoidance constraints^[3]. To this end, this study proposes a trajectory planning method suitable for continuous curves. First, a safe path is generated by dynamic planning in the Frenet coordinate system to ensure the coverage of the feasible domain; then a multi-objective optimization function is constructed in the Cartesian coordinate system, the constraints are converted into gradient form for iterative optimization, and the path is smoothed by curve interpolation. This method

reduces the dependence on the initial trajectory, solves the coordinate conversion problem in curve scenes, achieves global optimization, and significantly improves the autonomous driving capability in complex scenes.

II. OPTIMIZATION PROBLEM

A. Vehicle Kinematics

This paper transforms these constraints into optimization objectives, thereby establishing an unconstrained optimization problem^[4]. The complex road curvature optimization problem consists of vehicle kinematic constraints, boundary condition constraints, and obstacle avoidance constraints. The vehicle kinematic constraints are presented as^[5]:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ v_{i+1} \\ \phi_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ v_i \\ \phi_i \\ \theta_i \end{bmatrix} + \Delta t \cdot \begin{bmatrix} v_i \cdot \cos\theta_i \\ v_i \cdot \sin\theta_i \\ a_i \\ \omega_i \\ \frac{v_i \cdot \tan\phi_i}{L_w} \end{bmatrix} \quad (1)$$

Wherein, Point P is the midpoint of the rear wheel axis, v is the velocity, a is the acceleration, ϕ is the steering angle, ω is the angular velocity, θ is the direction angle, and L_w is the wheelbase. Other geometric parameters such as L_F (front overhang length), L_R (rear overhang length) and L_B (width) are shown in Figure 1^[6].

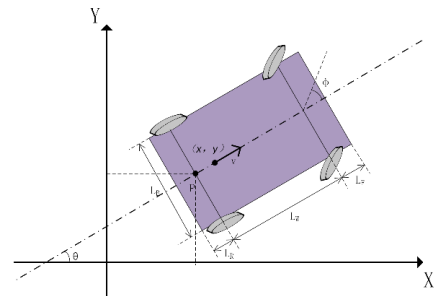


Figure 1. Vehicle kinematics and geometric

In path planning, the flexibility of curvature must meet the vehicle kinematic limits to ensure steering constraints. The maximum curvature constraint of vertex i is determined by the adjacent points $i-1$, i and $i+1$, and its tangent angle change can be expressed as:

$$\Delta\phi_i = \cos^{-1} \frac{\Delta X_i^T \Delta X_{i+1}}{|\Delta X_i| |\Delta X_{i+1}|} \quad (2)$$

$$\begin{aligned} \frac{\partial \kappa_i}{\partial \mathbf{x}_i} &= -\frac{1}{|\Delta \mathbf{x}_i|} \frac{\partial \Delta\phi_i}{\partial \cos(\Delta\phi_i)} \frac{\partial \cos(\Delta\phi_i)}{\partial \mathbf{x}_i} - \frac{\Delta\phi_i}{(\Delta \mathbf{x}_i)^2} \frac{\partial \Delta \mathbf{x}_i}{\partial \mathbf{x}_i} \\ \frac{\partial \kappa_i}{\partial \mathbf{x}_{i-1}} &= -\frac{1}{|\Delta \mathbf{x}_i|} \frac{\partial \Delta\phi_i}{\partial \cos(\Delta\phi_i)} \frac{\partial \cos(\Delta\phi_i)}{\partial \mathbf{x}_{i-1}} - \frac{\Delta\phi_i}{(\Delta \mathbf{x}_i)^2} \frac{\partial \Delta \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} \\ \frac{\partial \kappa_i}{\partial \mathbf{x}_{i+1}} &= -\frac{1}{|\Delta \mathbf{x}_i|} \frac{\partial \Delta\phi_i}{\partial \cos(\Delta\phi_i)} \frac{\partial \cos(\Delta\phi_i)}{\partial \mathbf{x}_{i+1}} \end{aligned} \quad (3)$$

The curvature formula is: $\kappa_i = (\Delta \theta_i) / |\Delta X_i|$, where $\Delta \theta_i$ is the change in tangent angle at vertex i , and $|\Delta X_i|$ is the displacement modulus between adjacent nodes. The curvature gradient can be obtained by differentiating the coordinates of nodes $i-1$, i , and $i+1$ to ensure that the path is smooth and meets kinematic constraints.

B. Constraints

At this time, the **curvature mutation constraint gradient** can be presented:

$$\begin{aligned} \nabla f_{curvature}(X_i) &= a(k_{i-1} - k_{max}) \frac{\partial \kappa_i}{\partial \mathbf{x}_{i-1}} + \\ & b(k_i - k_{max}) \frac{\partial \kappa_i}{\partial \mathbf{x}_i} + c(k_{i+1} - k_{max}) \frac{\partial \kappa_i}{\partial \mathbf{x}_{i+1}} \end{aligned} \quad (4)$$

Among them, a , b , and c represent the distribution coefficients of the curvature changes of three adjacent points respectively. In path optimization, the ego vehicle is simplified into a rectangle, and a driving corridor is constructed along the reference path to ensure that the ego vehicle stays within the corridor to avoid collisions with obstacles and road boundaries.

$$\begin{aligned} x_i^{lbj} &\leq x_i^{vej} \leq x_i^{ubj}, \\ y_i^{lbj} &\leq y_i^{vej} \leq y_i^{ubj}, \\ j &= 1, \dots, N. \end{aligned} \quad (5)$$

Among them, x_i^{lbj} , x_i^{ubj} , y_i^{lbj} , y_i^{ubj} are the upper and lower boundaries of the corridor.

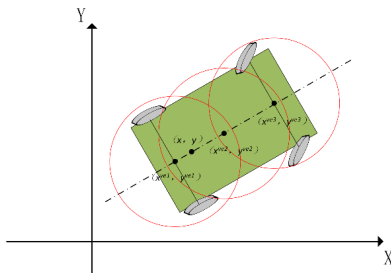


Figure 2. Multi-circle envelope of the vehicle

As shown in Figure 2, for the ego vehicle, the rectangular body is evenly covered by N disks with the same radius^[7], where (x_i^{vej}, y_i^{vej}) represents the position of the center of the j th disk:

$$\begin{aligned} x_i^{egoj} &= x_i + \left(\frac{2j-1}{2N} \cdot (L_R + L_W + L_F) - L_R \right) \cdot \cos \theta_i, \\ y_i^{egoj} &= y_i + \left(\frac{2j-1}{2N} \cdot (L_R + L_W + L_F) - L_R \right) \cdot \sin \theta_i, \\ j &= 1, \dots, N \end{aligned} \quad (6)$$

$$R^{ego} = \sqrt{\left(\frac{L_R + L_W + L_F}{2N} \right)^2 + \left(\frac{L_B}{2} \right)^2} \quad (7)$$

Among them, R^{ego} represents the radius of each disk.

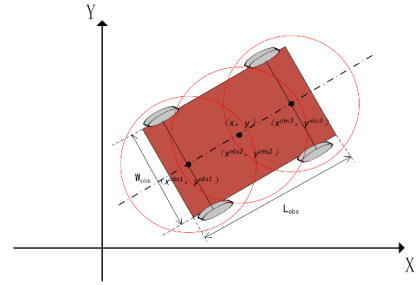


Figure 3. Obstacle multi-circle envelope

As shown in Figure 3, the geometric center of the obstacle is its coordinate center, and the derivation method is similar to that of the vehicle^[8]. Let the coordinates of the nearest obstacle be $o_k = (x_k^{obs}, y_k^{obs})$, and (x_k^{obsj}, y_k^{obsj}) be the center position of the j th disk of the obstacle.

$$\begin{aligned} x_k^{obsj} &= x_k^{obs} + (j-2) \cdot \left(\frac{2j-1}{2N} \cdot L_{obs} + W_{obs} \right) \cdot \cos \theta_{obs}, \\ y_k^{obsj} &= y_k^{obs} + (j-2) \cdot \left(\frac{2j-1}{2N} \cdot L_{obs} + W_{obs} \right) \cdot \sin \theta_{obs}, \\ j &= 1, \dots, N \end{aligned} \quad (8)$$

$$R^{obs} = \sqrt{\left(\frac{L_{obs}}{2N} \right)^2 + \left(\frac{W_{obs}}{2} \right)^2} \quad (9)$$

Among them, R^{obs} represents the radius of each obstacle disk.

Each iteration checks if the minimum distance between vehicle (x_i^{vej}, y_i^{vej}) and obstacle (x_k^{obs}, y_k^{obs}) envelope coordinates is less than the sum of their radii. If true, a collision risk is detected, activating the penalty term. The **obstacle gradient** is calculated to adjust the path, ensuring the vehicle avoids the obstacle:

$$\nabla f_{obs}(X_i) = 2(|X_i - o_k| - R^{obs} - R^{ego}) \frac{X_i - o_k}{|X_i - o_k|} \quad (10)$$

In iterative optimization, road boundaries are detected via sensor fusion, and the distance between the vehicle and boundary points is calculated^[9]. If this distance is less than the envelope radius, a collision risk triggers the penalty term, and the collision avoidance gradient is computed based on the nearest boundary point r_k :

$$\nabla f_{boundary}(X_i) = 2(|X_i - r_k| - R^{ego}) \frac{X_i - r_k}{|X_i - r_k|} \quad (11)$$

The initial path is the optimal path rough solution calculated in the previous section. The path sequence is defined as $X_i = (x_i, y_i)$, where $i \in [1, n]$. In each iteration, the gradient direction of the i -th point is expressed as:

$$g_k(X_i) = \alpha_1 \nabla f_{curvature}(X_i) + \alpha_2 \nabla f_{obs}(X_i) + \alpha_3 \nabla f_{boundary}(X_i) \quad (12)$$

Among them, $\alpha_1, \alpha_2, \alpha_3$ are gradient direction weights, $\nabla f_{curvature}(X_i)$, $\nabla f_{obs}(X_i)$, $\nabla f_{boundary}(X_i)$ are curvature constraint, obstacle avoidance and boundary avoidance gradients respectively.

C. Curve Interpolation

Traditional interpolation methods cannot guarantee curvature continuity, which may cause vehicle jitter and unstable control. Cubic curve interpolation generates smooth and efficient driving trajectories through second-order derivative continuity and precise boundary control, improving safety and comfort^[10].

A cubic polynomial in the given interval $[s_i, s_{i+1}]$:

$$\begin{aligned} l(s) &= l_i + dl_i \Delta s + \frac{ddl_i}{2} \Delta s^2 + \frac{ddl_{i+1} - ddl_i}{6} \Delta s^3 \\ l'(s) &= dl_i + ddl_i \Delta s + \frac{ddl_{i+1} - ddl_i}{2h} \Delta s^2 \\ l''(s) &= ddl_i + \frac{ddl_{i+1} - ddl_i}{h} \Delta s \end{aligned} \quad (13)$$

In order to meet the requirements of real-time and densely spaced autonomous driving scenarios, a segmented jerk optimization method is used, that is, the calculation of $l'(s)$ is described by linearly approximating the acceleration change, and the accuracy of $l''(s)$ is maintained, so that $l(s)$ can be rewritten as:

$$l(s) = l_i + dl_i \Delta s + \left(\frac{ddl_i}{3} + \frac{ddl_{i+1}}{6} \right) \Delta s^2 \quad (14)$$

Where $h = s_{i+1} - s_i$, $\Delta s = s - s_i$, and l_i, dl_i, ddl_i define the initial state (position, velocity, acceleration) of the path segment respectively.

III. SIMULATION ANALYSIS

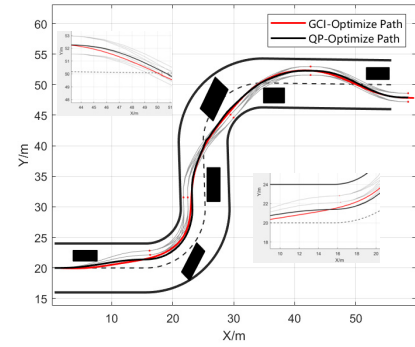
This study verifies the proposed path planning method through the MATLAB simulation platform. The experimental

design adopts a composite road scene containing continuous straight roads and curves, and sets static and dynamic obstacles in it. The specific experimental parameter configuration is shown in Table 1.

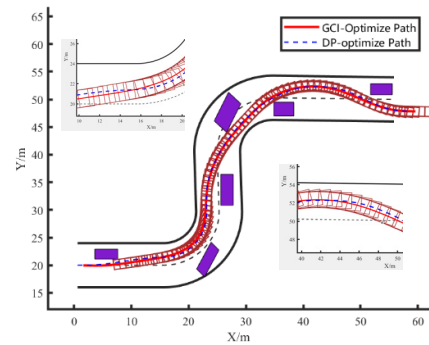
TABLE I. Main Parameters of Iterative optimization based on GCI

Parameter	Mean	Value
L_W	Wheelbase	2.80m
L_B	Vehicle width	1.942m
L_F	Front overhang length	0.96m
L_R	Rear overhang length	0.929m
α_1	Curvature Weight	2.8
α_2	Obstacle Weight	0.01
α_3	Boundary Weight	0.01
$iter_{MAX}$	Maximum iterations	1000
N	Enveloping circles	3
T	Planning duration	10s

This study compares two path planning methods: GCI and Frenet-QP. Both generate the initial path through dynamic programming, then construct a quadratic programming problem and optimize it using the OSQP solver.



(a) Comparison between GCI and QP



(b) Comparison between GCI and DP

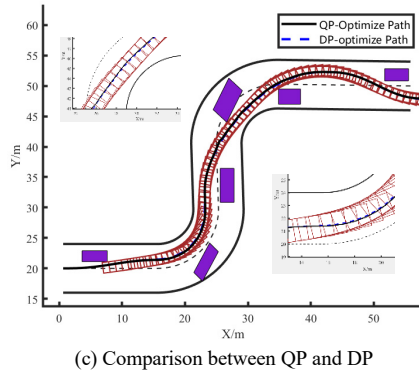


Figure 4. Comparison of various optimization methods

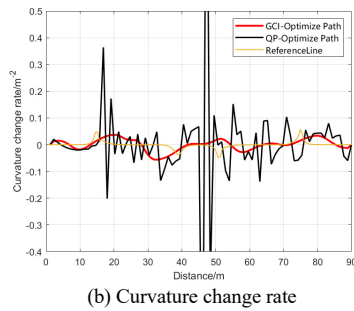
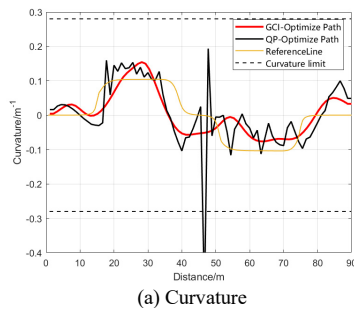


Figure 5. Geometric property comparison

As shown in Figure 4, the QP method is constrained by the Frenet framework, resulting in vector distortion and curvature violations in high-curvature areas (Figure 5a). In contrast, the GCI method uses dual coordinate system optimization to strictly enforce curvature constraints (Figure 5b), achieving smoother curvature, better kinematic adaptability, and reduced collision risk.

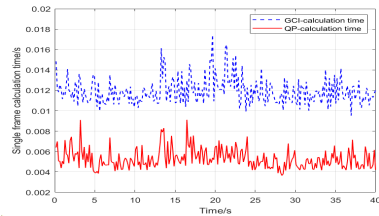


Figure 6. Comparison of optimized calculation time

As shown in Figure 6, the GCI method is more efficient than the quadratic programming method, solving a single-frame path in 0.0068 seconds versus 0.0139 seconds, by avoiding matrix inversion through iterative optimization.

IV. CONCLUSIONS

This study proposes a dual-coordinate trajectory planning method combining the strengths of Frenet and Cartesian systems. It generates an initial solution in the Frenet system and optimizes it with gradient interpolation in the Cartesian system, ensuring smoothness and kinematic constraints. Experiments show this method effectively handles continuous curvature changes, avoids distortion from traditional transformations, and improves computational efficiency while ensuring obstacle avoidance and accuracy, offering a reliable solution for autonomous driving.

ACKNOWLEDGMENT

This article represents a phased accomplishment within the framework of the National Natural Science Foundation of China General Program titled "Road Friction Coefficient Estimation for Each Wheel Based on Multi Degree of Freedom Data Fusion and Torque Control Method" (52272398).

REFERENCES

- [1] Cheng Z , Zeng X , Fang H ,et al.Hierarchical MPC-based Motion Planning for Automated Vehicles in Parallel Autonomy[J].Unmanned Systems, 2024, 12(4).DOI:10.1142/S2301385024500286.
- [2] Liu L S , Lin J F , Yao J X ,et al.Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach[J].Wireless Communications and Mobile Computing, 2021.DOI:10.1155/2021/8881684.
- [3] Xu J , Park K S .Kinematic performance-based path planning for cable-driven parallel robots using modified adaptive RRT*[J].Microsystem Technologies, 2022, 28(10):12.DOI:10.1007/s00542-022-05319-3.
- [4] Leng B .An Optimized Trajectory Planner and Motion Controller Framework for Autonomous Driving in Unstructured Environments[J].Sensors, 2021, 21.DOI:10.3390/s21134409.
- [5] Li B , Zhang Y .Fast Trajectory Planning in Cartesian rather than Frenet Frame: A Precise Solution for Autonomous Driving in Complex Urban Scenarios[J].IFAC-PapersOnLine, 2020, 53(2):17065-17070.DOI:10.1016/j.ifacol.2020.12.1549.
- [6] Zeng D , Yu Z , Xiong L ,et al.Driving-behavior-oriented trajectory planning for autonomous vehicle driving on urban structural road[J].Jnl of Automobile Engineering: Part D, 2021, 235(4):21.DOI:10.1177/0954407020969992.
- [7] Li H , Wu C , Chu D ,et al.Combined Trajectory Planning and Tracking for Autonomous Vehicle Considering Driving Styles[J].Quality Control, Transactions, 2021, 9(1):9453-9463.
- [8] Li B , Shao Z .A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles[J].Knowledge-Based Systems, 2015, 86:11-20.DOI:10.1016/j.knsys.2015.04.016.
- [9] Chen Z , Liu Y , Liu Y ,et al.Dynamic Lane-Changing Trajectory Planning for Autonomous Vehicles Based on Discrete Global Trajectory[J].IEEE Transactions on Intelligent Transportation Systems, 2021.DOI:10.1109/TITS.2021.3083541.
- [10] Yang L , Lu C , Xiong G M ,et al.A Hybrid Motion Planning Framework for Autonomous Driving in Mixed Traffic Flow[J].Green Energy and Intelligent Transportation, 2022.DOI:10.1016/j.geits.2022.100022.