International Journal of Automotive Technology (2024) 25:867–880
https://doi.org/10.1007/s12239-024-00087-7

Parallel Parking Path Planning and Tracking Control Based on Simulated Annealing Algorithm

Leiyan Yu[1] · Yongpeng Cai[1] · Xiangbo Feng[2] · Yuanxue Zhou[2] · Zihua Hu[1] · Meilan Tian[1] · Shaohua Sun[1]

Received: 22 January 2024 / Revised: 6 March 2024 / Accepted: 19 March 2024 / Published online: 11 April 2024

Abstract
To address the issues of curvature discontinuity and terminal tire non-return in the parallel parking of autonomous vehicles, a novel parallel parking path planning method based on the combination of the quintic polynomial curve and an improved sigmoid function was proposed. First, a vehicle kinematic model was established. Second, considering the position, front wheel angle, and yaw angle constraints during the parking process, a hybrid superimposed curve was designed. The parking path planning problem was converted into an optimal control problem, with the maximum curvature and curvature at both ends as objective functions, and the parameters were optimized using the simulated annealing algorithm. Subsequently, for path tracking control, the simulated annealing algorithm was used to optimize the prediction time horizon of the model predictive control algorithm. Finally, a series of actual vehicle experiments were conducted based on the Apollo Autonomous Driving Developer Suite, and the effectiveness of the proposed path planning method was validated. Therefore, this method can provide a certain reference for automatic parking path planning technology.

Keywords Parallel parking · Path planning · Simulated annealing algorithm · Model predictive control algorithm · Pure pursuit algorithm

1 Introduction

With the acceleration of urbanization and the increasing number of car ownership, while bringing convenience to transportation, the issue of “difficult parking” has become increasingly prominent (Tang et al., 2019). As a key part of autonomous driving technologies, automatic parking technology can not only effectively solve the occurrence of collisions, scrapes, and other accidents caused by lack of experience or technology of drivers, improve parking safety, but also alleviate the stress of drivers when parking, significantly enhancing parking efficiency and driving comfort (Li et al., 2020). Therefore, it is of great significance to conduct in-depth research on automatic path planning and path tracking technology.

Parking path planning methods are primarily categorized into four groups: geometric curve methods (Chen et al., 2023), graph search methods (Cao et al., 2023), random sampling methods (Zhang et al., 2021), and intelligent optimization methods (Su et al., 2019). In the geometric curve method, Helene et al. (2012) analyzed narrow parking spaces based on the clothoid curve and designed parking paths suitable for narrow parking spaces. However, the design of the clothoid curve was relatively complex requiring a large amount of computation. In the graph search method, Ren et al. (2022) proposed an automatic parking path planning method based on the Hybrid A* algorithm and RS curves. They optimized the path using Bezier curves and gradient descent. The results demonstrated that this method had improved path search capability and flexibility in complex scenarios. Although the method solved the problem of discontinuous curvature, the planned path was not smooth enough and the tracking effect was poor. In the random sampling method, Zhang et al. (2022) proposed a set of principles for selecting reference points and updating path nodes that conformed to the kinematic constraints of vehicles, improving the Rapidly Exploring Random Tree algorithm. The results showed that the proposed method was able

✉ Leiyan Yu
leiyanyu@upc.edu.cn

[1] School of Mechanical and Electrical Engineering, China University of Petroleum (East China), Qingdao 266580, China

[2] Weifang Airui Brake System Co., Ltd., Weifang 261057, China

to safely guide a car to complete parking tasks. However, the quality of the paths planned by this method was poor, and they were difficult to track. In the intelligent optimization method, Chen et al. (2021) proposed a staged training method based on curriculum learning using the coordinates of the center of the rear axle and the front wheel steering angle as state variables, achieved a planning success rate of 90.6% while satisfying comfort based on deep reinforcement learning. However, it required a large amount of data and had high real-time requirements.

Currently, most research on path planning for automatic parking can satisfy the curvature constraints and obstacle avoidance constraints during the parking process. However, the planned paths are complex, which is not conducive to real-time tracking control.

Common parking path tracking methods include pure pursuit (PP) algorithm (Shin et al., 2022), fuzzy control (Wang et al., 2022), and model predictive control (MPC) algorithm (Liu, 2022). In the pure pursuit algorithm, Erno et al. (2019) proposed an improved pure pursuit algorithm. By selecting multiple target points, they adjusted the forward distance based on curvature and lateral deviation, realizing dynamic optimization of the preview distance. Experimental verification showed good feasibility in practical applications and better tracking performance. In the pure fuzzy control, Naitik et al. (2018) proposed a parallel parking method based on hybrid fuzzy inference, which considered various parking scenarios, including sudden obstacle appearance during parking, the experimental results showed that the method could successfully complete parking in different parking scenarios, but its tracking accuracy was relatively poor. In the model predictive control algorithm, Yu et al. (2023) established a parking path tracking control method based on model predictive control, which tracked paths consisting of clothoid curves, arcs, and straight lines. The results indicated that the method had high tracking accuracy, with a maximum tracking error of 0.023 m.

The simulated annealing algorithm, which boasts high accuracy, strong universality, and flexibility in application, is widely utilized in research. Zhao et al., (2022a, 2022b) optimized the designed path planning method for moving targets through the simulated annealing algorithm. The results showed that the designed method was able to generate multi-UAV collaborative verification paths for moving ships that met time constraints. Zhao et al., (2022a, 2022b) proposed an artificial potential field method optimized by the simulated annealing algorithm. The results showed that this method enabled the mobile robot to escape from local minima positions, successfully reaching the target position with shorter time and greater stability.

To reduce the complexity of path planning, ensure the continuity of path curvature, and enhance parking efficiency, this paper proposes a parallel parking path planning method based on a hybrid superimposed curve of the quintic polynomial curve and improved sigmoid function. The contributions of this paper are threefold:

(1) The method proposed in this paper effectively addresses the issue of sudden changes in curvature that occur in conventional parallel parking path planning methods.
(2) The method utilizes a simulated annealing algorithm to optimize the maximum curvature and the curvature at both endpoints of the parking path, satisfying the maximum curvature constraint and solving the problem of the vehicle not being parallel to the parking space at the end of the parking maneuver.
(3) The path planned by the method proposed in this paper is easy to track, and its lateral tracking error is small, which has been verified by an optimized MPC algorithm.

In this paper, first, a vehicle kinematic model is established. Second, considering the position, front wheel angle, and yaw angle constraints during the parking process, a hybrid superimposed curve is designed. The parking path planning problem is converted into an optimal control problem, with the maximum curvature and curvature at both ends as objective functions, and the parameters are optimized using the simulated annealing algorithm. Finally, virtual simulation experiments and real-vehicle tests are conducted using the MPC algorithm optimized with the simulated annealing algorithm, verifying the effectiveness of the proposed path planning method.

2 Model Establishment and Vehicle Constraints

2.1 The Establishment of Vehicle Kinematics Model

During parking, vehicle speed is low, and side slip caused by tire lateral force can be ignored. The tires only perform pure rolling. According to the Ackerman steering principle, the entire vehicle is modeled as a rigid body, and a kinematic model of the vehicle is established, as shown in Fig. 1.

Here, $(x_\mathrm{f}, y_\mathrm{f})$ and $(x_\mathrm{r}, y_\mathrm{r})$ are the coordinates of the center points of the vehicle's front and rear axles; v_f and v_r are the speeds of the vehicle's front and rear axles centers; l_f, l, and l_r are the lengths of the vehicle's front suspension, wheelbase, and rear suspension, respectively; θ is the vehicle's yaw angle; δ is the vehicle's front wheel steering angle; a, b, c, and d are the four vertices of the vehicle's outline. Taking the center of the vehicle's rear axle as the reference point, a differential equation for the vehicle's kinematics is established:

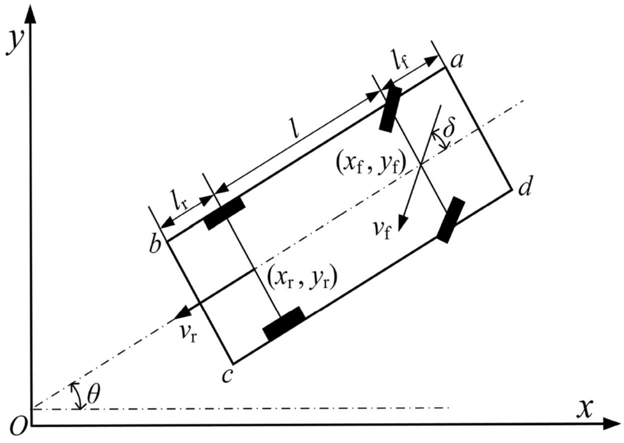


Fig. 1 Kinematic model of vehicle

$$\begin{cases} \dot{x} = v_r \cdot \cos\theta \\ \dot{y} = v_r \cdot \sin\theta \\ \dot{\theta} = v_r \cdot \tan\delta / l \end{cases} \quad (1)$$

Based on geometric relationships and the location of the vehicle's rear axle center point, the coordinates of the four vertices of the vehicle's outline can be calculated:

$$\begin{cases} x_a = x_r + (l + l_f)\cos\theta - \frac{W\sin\theta}{2} \\ y_a = y_r + (l + l_f)\sin\theta + \frac{W\cos\theta}{2} \\ x_b = x_r - l_r\cos\theta - \frac{W\sin\theta}{2} \\ y_b = y_r - l_r\sin\theta + \frac{W\cos\theta}{2} \\ x_c = x_r - l_r\cos\theta + \frac{W\sin\theta}{2} \\ y_c = y_r - l_r\sin\theta - \frac{W\cos\theta}{2} \\ x_d = x_r + (l + l_f)\cos\theta + \frac{W\sin\theta}{2} \\ y_d = y_r + (l + l_f)\sin\theta - \frac{W\cos\theta}{2} \end{cases} \quad (2)$$

Here, W is the vehicle width.

2.2 Vehicle Self-Constraints

During the automatic parking process, to ensure the traceability of the parking path, it is necessary to meet the corresponding constraints of the vehicle itself:

$$\begin{cases} |v| \le v_{\max} \\ |a| \le a_{\max} \\ |j| \le j_{\max} \\ |\delta| \le \delta_{\max} \\ |w| \le w_{\max} \end{cases} \quad (3)$$

Here, $v_{\max}$, $a_{\max}$, $j_{\max}$, $\delta_{\max}$, and $w_{\max}$ are the maximum allowable speed, maximum acceleration, maximum acceleration rate, maximum front wheel angle, and maximum front wheel angle rate during parking, respectively.

3 Parallel Parking Path Planning and Optimization

3.1 Parallel Parking Path Planning Based on Quintic Polynomial Curve

During the parking process, curvature constraints need to be satisfied to avoid the occurrence of tire scrubbing. At the same time, the yaw angle and curvature at the starting and ending points of parking should be as small as possible, so as to ensure that the vehicle body is parallel to the lane lines and the steering wheel is in the neutral position at these two points. The quintic polynomial curve, which has continuous curvature and small computational complexity, satisfies the above requirements. Therefore, a quintic polynomial curve is used to plan the parallel parking path (Yu et al., 2022).

The expression of the quintic polynomial curve:

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 \quad (4)$$

Here, $a_0 \sim a_5$ are the coefficients of the quintic polynomial curve.

The coordinates of the parallel parking starting point, P_1, are given as (x_1, y_1), and the coordinates of the ending point, P_2, are given as (x_2, y_2). Based on the coordinates of these two points, a vehicle position constraint equation is established:

$$y_1 = a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + a_4x_1^4 + a_5x_1^5 \quad (5)$$

$$y_2 = a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3 + a_4x_2^4 + a_5x_2^5 \quad (6)$$

The body of the parallel parking vehicle should be parallel to the lane line at both the starting and ending points. This implies that the first derivative of the path curve at these two points is zero. The vehicle yaw angle constraint equation is established as follows:

$$0 = a_1 + 2a_2x_1 + 3a_3x_1^2 + 4a_4x_1^3 + 5a_5x_1^4 \quad (7)$$

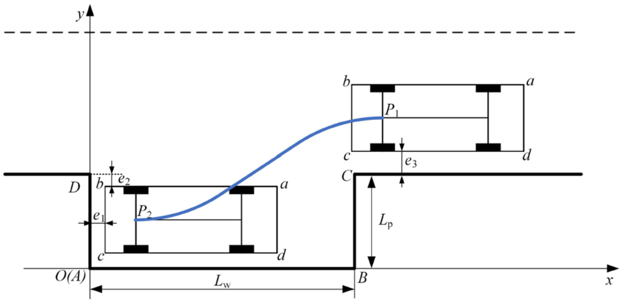


Fig. 2 Selection of parallel parking starting point and ending point

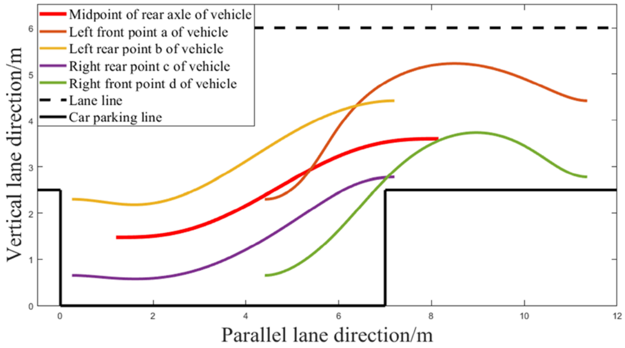


Fig. 3 Track of the rear axle center of the vehicle

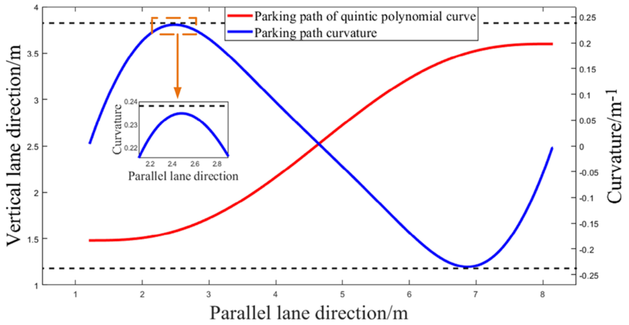


Fig. 4 Parking path and curvature of quintic polynomial curve

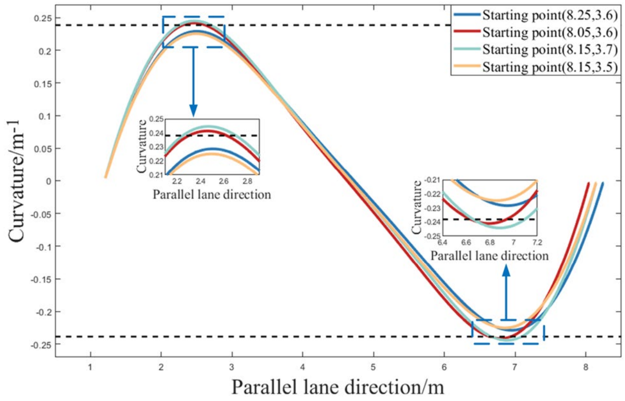


Fig. 5 Comparison of curvature at different parking starting points

$$0 = a_1 + 2a_2x_2 + 3a_3x_2^2 + 4a_4x_2^3 + 5a_5x_2^4 \tag{8}$$

The front wheel angle at the starting and ending points of parallel parking is expected to be aligned, implying that the second derivative of the path curve at these two points is required to be zero. A constraint equation for the vehicle front wheel angle is established as follows:

$$0 = 2a_2 + 6a_3x_1 + 12a_4x_1^2 + 20a_5x_1^3 \tag{9}$$

$$0 = 2a_2 + 6a_3x_2 + 12a_4x_2^2 + 20a_5x_2^3 \tag{10}$$

A parallel parking space is set up with a length of $L_w = 7$ m and a width of $L_p = 2.5$ m. The starting and ending points of the parking are presented as shown in Fig. 2. The distances from the rear end and left end of the vehicle outline at the parking ending point to the left side and top side of the parking space are designated as e_1 and e_2. The distance from the right end of the vehicle outline at the parking starting point to the parking space line is known as e_3. In consideration of safety and curvature variations during the parking process, e_1, e_2, and e_3 are set at 0.25 m, 0.2 m, and 0.3 m, respectively.

The starting point P_1 coordinate of the quintic polynomial curve is set as (8.15 m, 3.60 m), and the ending point P_2 coordinate is set as (1.20 m, 1.48 m). Equations (5) through (10) are combined to calculate the parallel parking quintic polynomial curve:

$$\begin{aligned} f(x) = {} & 0.00078445x^5 - 0.0183x^4 + 0.1399x^3 \\ & -0.3587x^2 + 0.3752x + 1.3407 \end{aligned} \tag{11}$$

As shown in Fig. 3, collisions between the trajectory of the vehicle's rear axle center and the surrounding lane lines and parking space lines are avoided, thereby complying with the obstacle avoidance constraints. As shown in Fig. 4, the parking path and curvature, planned via the quintic polynomial curve, are smooth and continuous, without any sudden changes. Moreover, compliance with the constraint set by the vehicle's structural parameters is maintained for the maximum curvature.

During the actual parking process, the vehicle may not be able to accurately reach the starting point due to positioning errors. Figure 5 shows the variations in parking path curvature when positioning errors in all four directions at the parking starting point are 0.1 m. When the parking starting points are positioned at (8.05 m, 3.60 m) and (8.15 m, 3.70 m), it is observed that the maximum curvature of the path exceeds the curvature constraint, which increases the risk of collision with surrounding obstacles during the parking process. Consequently, this may result in the vehicle's inability to complete the parking smoothly.

3.2 Parallel Parking Path Planning Based on Improved Sigmoid Function

The quintic polynomial curve exhibits the problem of excessive curvature, and when the parking starting point is not properly selected, it fails to meet the curvature constraint conditions. Consequently, an improved Sigmoid function is used to plan the parallel parking path (Xiong et al., 2017).

The expression of the improved Sigmoid function:

$$g(x) = \frac{b_0}{1 + e^{(b_1 x + b_2)}} + b_3 x \tag{12}$$

Here, $b_0 \sim b_3$ are the coefficients of the improved Sigmoid function.

The position constraints and yaw angle constraints should be satisfied by the improved Sigmoid function:

$$\begin{cases} g(x_1) = y_1, g(x_2) = y_2 \\ \dot{g}(x_1) = 0, \dot{g}(x_2) = 0 \end{cases} \tag{13}$$

Taking the parking starting point (8.05 m, 3.60 m) and the ending point (1.20 m, 1.48 m) as an example, Eq. (13) should be solved to calculate the expression of the improved Sigmoid function:

$$g(x) = \frac{12.7444}{1 + e^{(-0.4156x + 1.9219)}} - 0.8286x \tag{14}$$

As shown in Fig. 6, the parking path and curvature planned by the improved Sigmoid function are compared with the path curvature planned by the above quintic polynomial curve. Although it satisfies the maximum curvature constraint, the curvature at the starting and ending points is too large, which leads to the occurrence of tire steering in place.

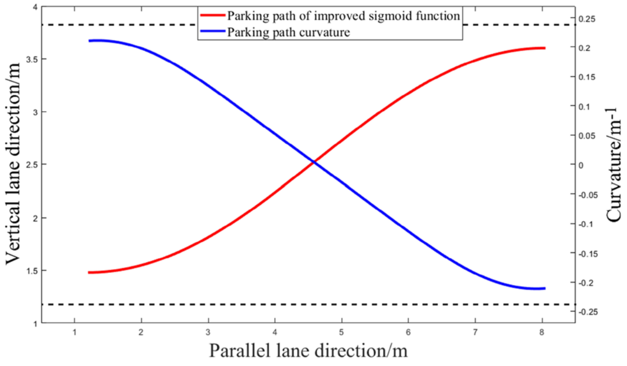


Fig. 6 Parking path and curvature of improved sigmoid function

3.3 Parallel Parking Path Planning Based on Superposition Curve Optimized by Simulated Annealing Algorithm

To solve the problem of the maximum curvature of the quintic polynomial curve exceeding the defined value and the excessive curvature of the endpoint of the improved Sigmoid function, the two curves are superimposed according to a certain ratio:

$$h(x) = w_1 f(x) + (1 - w_1) g(x) \tag{15}$$

Here, w_1 is the proportional coefficient.

The parking path must satisfy the conditions of continuous curvature and maximum curvature constraints. The magnitude of the parking path curvature is closely related to ride comfort; the smaller the maximum curvature of the path, the higher the comfort. To improve ride comfort, the maximum curvature of the parking path, $k_{\max}$, should be made as small as possible. When the maximum curvature $k_{\max}$ is smaller, that is, w_1 is smaller, the front wheel angle of the vehicle during path tracking is also smaller, thereby improving the smoothness and ride comfort during parking. However, the corresponding curvatures at the parking starting and ending points are too large, which increases tire wear. When w_1 is larger, it solves the problem of tire steering in place better, but the maximum curvature also increases. To balance the relationship between the maximum curvature and the curvatures at the parking starting and ending points, a corresponding objective function is designed:

$$J_1 = w_2(k_{\text{start}} + k_{\text{goal}}) + w_3 k_{\max} \tag{16}$$

Here, k_{start} is the curvature corresponding to the parking starting point; k_{goal} is the curvature corresponding to the parking ending point; w_2 and w_3 are the weighting coefficients for the curvatures at the parking starting and ending points and the maximum curvature. To increase the impact of the maximum curvature on the objective function, w_2 is set to 2 and w_3 is set to 13.

The simulated annealing algorithm, which is easy to implement and robust, is a general probabilistic optimization algorithm inspired by the annealing process of solids. The algorithm starts at a high initial temperature and gradually anneals, performing local search at each temperature until reaching the optimal solution (Xu et al., 2024). The acceptance probability based on the Metropolis criterion is:

$$P = \begin{cases} 1, & E_{t+1} < E_t \\ e^{\frac{-(E_{t+1} - E_t)}{\alpha T}}, & E_{t+1} \geq E_t \end{cases} \tag{17}$$

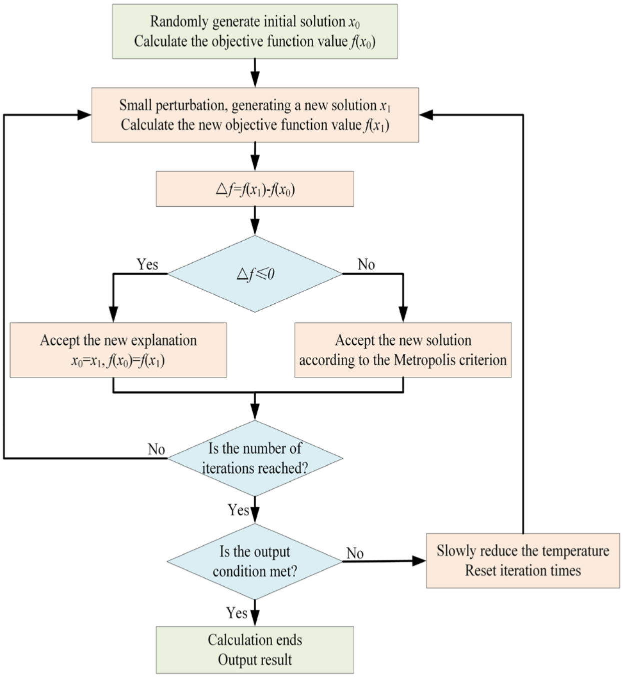

Randomly generate initial solution x_0
Calculate the objective function value $f(x_0)$
Small perturbation, generating a new solution x_1
Calculate the new objective function value $f(x_1)$
$\Delta f=f(x_1)-f(x_0)$
$\Delta f\leqslant 0$
Yes
No
Accept the new explanation
$x_0=x_1, f(x_0)=f(x_1)$
Accept the new solution
according to the Metropolis criterion
No
Is the number of
iterations reached?
Yes
Is the output
condition met?
No
Slowly reduce the temperature
Reset iteration times
Yes
Calculation ends
Output result

Fig. 7 Flowchart of simulated annealing algorithm

Table 1 Settings of simulated annealing algorithm parameters

Parameter	Symbol	Numerical value
Initial temperature	T	100
Temperature attenuation coefficient	α	0.9
Maximum number of iterations	M	50
The number of iterations at each temperature	L_k	50

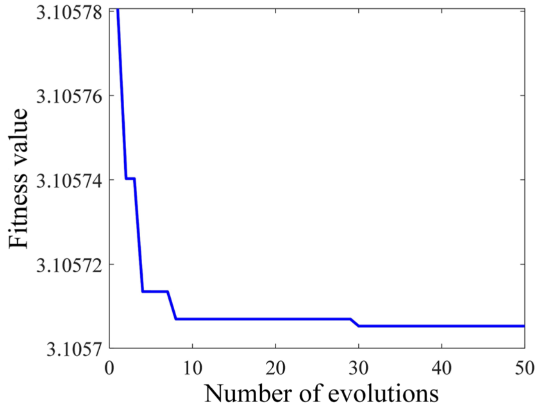
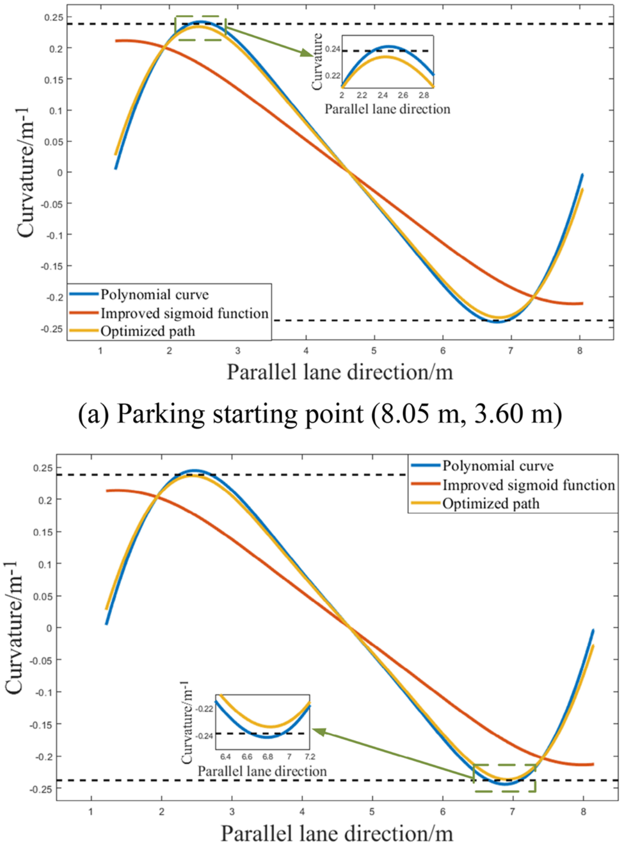


Fig. 8 Iteration process of simulated annealing algorithm



(a) Parking starting point (8.05 m, 3.60 m)

(b) Parking starting point (8.15 m, 3.70 m)

Fig. 9 Parking path and curvature at different starting points

Here, E_t is the system energy at time t, which is the value of the objective function; E_{t+1} is the system energy at time $t+1$; α is the temperature decay coefficient; T is the initial temperature.

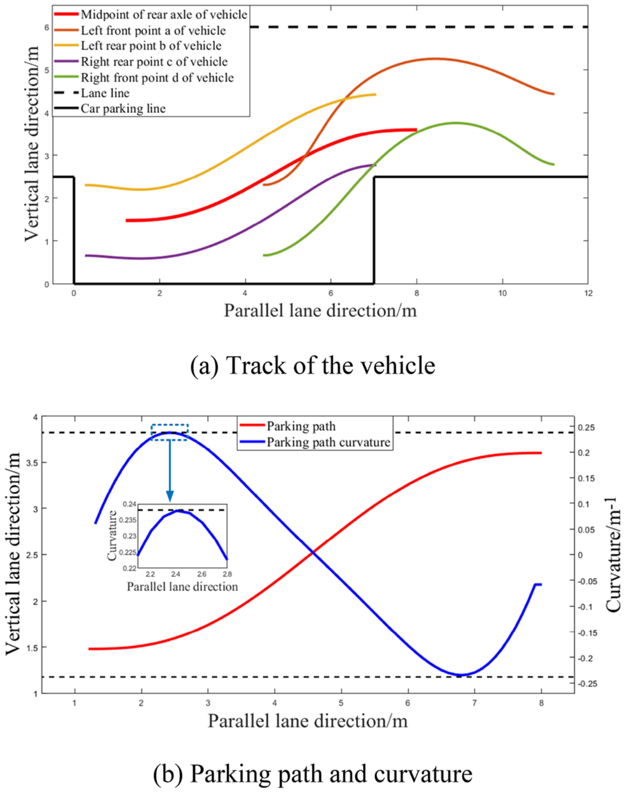
As shown in Fig. 7, the flowchart of the simulated annealing algorithm is presented. The main parameter settings are listed in Table 1.

As shown in Fig. 8, the iterative process for solving the proportional coefficient w_1 using the simulated annealing algorithm is presented. When the iteration reaches the 30th time, the optimal solution is obtained, characterized by a fitness value of 3.105708, and the proportional coefficient w_1 has a value of 0.887. When the parking starting point (8.05 m, 3.60 m) is used as an example and substituted into Eq. (15), the optimized parallel parking path expression is derived:

$$h(x) = 0.887\,f(x) + 0.113\,g\,(x) \tag{18}$$

As shown in Fig. 9, the curvature of the parallel parking path after optimization using the simulated annealing algorithm is presented. The problems of the quintic polynomial curve and the improved Sigmoid function are solved by the optimized path, which improves ride comfort while avoiding steering tire in place. The optimized maximum curvature $k_{\max}$ is less than $1/R_{\min}$, satisfying the maximum curvature constraint.

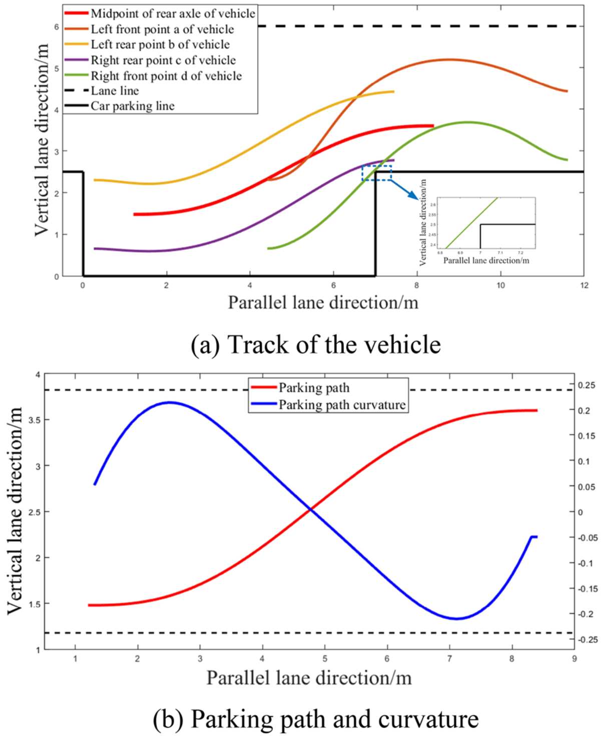
The path planning method mentioned has validated its effectiveness for specific starting and ending points. To ensure the safety of the parking process, the coordinates of the ending position and the longitudinal coordinate of the starting position are kept constant. The study examines the extreme values of the starting point's horizontal coordinate under the conditions that meet both obstacle avoidance and curvature constraints. As shown in Fig. 10, the curvature constraint is not satisfied when the horizontal coordinate of the starting point is less than 8.01 m, thus the minimum value for the starting point's horizontal coordinate is established at 8.01 m. Figure 11 shows that when the horizontal coordinate of the starting point is 8.41 m, it satisfies the curvature constraint while just meeting the requirements for obstacle avoidance, thereby setting the maximum value for the starting point's horizontal coordinate at 8.41 m.



(a) Track of the vehicle

(b) Parking path and curvature

Fig. 10 The track and curvature of the vehicle when the parking starting point is (8.01 m, 3.60 m)



(a) Track of the vehicle

(b) Parking path and curvature

Fig. 11 The track and curvature of the vehicle when the parking starting point is (8.41 m, 3.60 m)

4 Parking Path Tracking Optimized by Simulated Annealing Algorithm

4.1 The Design of Model Predictive Controller

The model predictive control algorithm comprises three parts: a predictive model, a rolling optimization, and a feedback correction. The core idea of it is the prediction of the system's behavior based on input information, which allows for adjustments to the system's state to achieve targeted control (Gong et al., 2020).

Based on the kinematic model of the vehicle, the state quantity is given as $\xi = (x, y, \theta)$, and the control quantity as $u = (\delta, v)$. Consequently, the reference system's state quantity and control quantity are expressed as follows:

$$\dot{\xi} = f(\xi, u) \tag{19}$$

The state quantity and control quantity of the reference point are to be substituted at any time:

$$\dot{\xi}_r = f(\xi_r, u_r) \tag{20}$$

Here, ξ_r and u_r are state quantity and control quantity at any point.

Based on the kinematics model, the state-space equation is a nonlinear system, which does not meet the requirements of model predictive control. Therefore, Taylor series expansion is used to linearize the equation at point (20) of Eq. (19), ignoring the higher order terms and retaining the first-order terms:

$$\dot{\xi} = f(\xi_r, u_r) + \frac{\partial f(\xi, u)}{\partial \xi}(\xi - \xi_r) + \frac{\partial f(\xi, u)}{\partial u}(u - u_r) \tag{21}$$

Equation (20) is to be subtracted from Eq. (21):

$$\dot{\xi} - \dot{\xi}_r = \frac{\partial f(\xi, u)}{\partial \xi}(\xi - \xi_r) + \frac{\partial f(\xi, u)}{\partial u}(u - u_r) \tag{22}$$

$\Delta\xi = \xi - \xi_{\mathrm{r}}$, $\Delta u = u - u_{\mathrm{r}}$ are defined, and Eq. (22) is rearranged to obtain:

$$\Delta\dot{\xi} = A\Delta\xi + B\Delta u \tag{23}$$

Here, $A = \begin{bmatrix} 0 & 0 & -v_{\mathrm{r}}\sin\theta_{\mathrm{r}} \\ 0 & 0 & v_{\mathrm{r}}\cos\theta_{\mathrm{r}} \\ 0 & 0 & 0 \end{bmatrix}$

$B = \begin{bmatrix} \cos\theta_{\mathrm{r}} & 0 \\ \sin\theta_{\mathrm{r}} & 0 \\ \tan\delta_{\mathrm{r}}/l & v_{\mathrm{r}}/\left(l\cos^2\delta_{\mathrm{r}}\right) \end{bmatrix}$

Equation (23) is to be discretized:

$$\Delta\dot{\xi} = \frac{\Delta\xi(k+1) - \Delta\xi(k)}{T} = A\Delta\xi(k) + B\Delta u(k) \tag{24}$$

Here, T is the sampling time, k is the current sampling instant, and $k+1$ is the subsequent sampling instant.

Based on Eq. (24), the discrete linear state-space equation can be derived as:

$$\tilde{\xi}(k+1) = A_k\tilde{\xi}(k) + B_k\tilde{u}(k) \tag{25}$$

Here, $A_k = T * A + E$, $B_k = T * B$.

Equation (25) is processed for the next step:

$$\Phi(k|t) = \begin{bmatrix} \tilde{\xi}(k|t) \\ \tilde{u}(k-1|t) \end{bmatrix} \tag{26}$$

$$\eta(k|t) = C_k\tilde{\xi}(k) \tag{27}$$

Here, $C_k = [1 \quad 0]$.

It can be deduced that:

$$\Phi(k+1|t) = \tilde{A}_k\Phi(k|t) + \tilde{B}_k\Delta u(k|t) \tag{28}$$

$$\eta(k|t) = \tilde{C}_k\Phi(k|t) \tag{29}$$

Here, $\widetilde{A}_k = \begin{bmatrix} A_k & B_k \\ 0_{m\times n} & I_m \end{bmatrix}$, $\widetilde{B}_k = \begin{bmatrix} B_k \\ I_m \end{bmatrix}$, $\widetilde{C}_k = \left[C_k 0\right]$, m is the control dimension, n is the state dimension.

The predictive horizon and control horizon of the system are designated as N_{p} and N_{c}, respectively. The state of the system is predicted based on Eqs. (28) and (29):

$$\begin{aligned} \Phi(k+N_p|t) = {} & \tilde{A}_k^{N_p}\Phi(k|t) + \tilde{A}_k^{N_p-1}\tilde{B}_k\Delta u(k|t) \\ & + \cdots + \tilde{A}_k^{N_p-N_c-1}\tilde{B}_k\Delta u(k+N_c|t) \end{aligned} \tag{30}$$

$$\begin{aligned} \eta(k+N_p|t) = {} & \tilde{C}_k\tilde{A}_k^{N_p}\Phi(k|t) + \tilde{C}_k\tilde{A}_k^{N_p-1}\tilde{B}_k\Delta u(k|t) \\ & + \cdots + \tilde{C}_k\tilde{A}_k^{N_p-N_c-1}\tilde{B}_k\Delta u(k+N_c|t) \end{aligned} \tag{31}$$

The output of the whole system in the future moment is expressed in the form of a matrix:

$$Y(k) = \Psi\Phi(k|t) + \Theta\Delta U(k) \tag{32}$$

Here, $Y(k) = \begin{bmatrix} \eta(k+1|t) \\ \eta(k+2|t) \\ \cdots \\ \eta\left(k+N_c|t\right) \\ \cdots \\ \eta\left(k+N_p|t\right) \end{bmatrix}$, $\Psi = \begin{bmatrix} \widetilde{C}_k\widetilde{A}_k \\ \widetilde{C}_k\widetilde{A}_k^2 \\ \cdots \\ \widetilde{C}_k\widetilde{A}_k^{N_c} \\ \cdots \\ \widetilde{C}_k\widetilde{A}_k^{N_p} \end{bmatrix}$,

$\Delta U(k) = \begin{bmatrix} \Delta u(k|t) \\ \Delta u(k+1|t) \\ \cdots \\ \Delta u(k+N_c|t) \end{bmatrix}$,

$$\Theta = \begin{bmatrix} \widetilde{C}_k\widetilde{B}_k & 0 & 0 & 0 \\ \widetilde{C}_k\widetilde{A}_k\widetilde{B}_k & \widetilde{C}_k\widetilde{B}_k & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots \\ \widetilde{C}_k\widetilde{A}_k^{N_c-1}\widetilde{B}_k & \widetilde{C}_k\widetilde{A}_k^{N_c-2}\widetilde{B}_k & \cdots & \widetilde{C}_k\widetilde{B}_k \\ \widetilde{C}_k\widetilde{A}_k^{N_c}\widetilde{B}_k & \widetilde{C}_k\widetilde{A}_k^{N_c-1}\widetilde{B}_k & \cdots & \widetilde{C}_k\widetilde{A}_k\widetilde{B}_k \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{C}_k\widetilde{A}_k^{N_p-1}\widetilde{B}_k & \widetilde{C}_k\widetilde{A}_k^{N_p-2}\widetilde{B}_k & \cdots & \widetilde{C}_k\widetilde{A}_k^{N_p-N_c-1}\widetilde{B}_k \end{bmatrix}.$$

In MPC, the objective function is designed to optimize control performance. It can be established as follows:

$$\begin{aligned} \min J(k) = {} & \sum_{i=1}^{N_p} ||\eta(k+i|t) - \eta_r(k+i|t)||_Q^2 \\ & + \sum_{i=1}^{N_c-1} ||\Delta U(k+i|t)||_R^2 + \rho\varepsilon^2 \end{aligned} \tag{33}$$

Here, $\eta_r(k+1|t)$ is the reference output; Q is the state weighting matrix; R is the control weighting matrix; ρ is the weighting coefficient; ε is the relaxation factor.

In the objective function, the system's ability to follow the desired path is represented by the first term, the constraint on control increment to avoid sudden changes in control inputs that might affect the continuity of control increments is represented by the second term, and the relaxation factor, which can prevent the situation of not obtaining optimal solutions during the solution process, is represented by the third term.

The objective function is transformed into a standard quadratic form:

$$\begin{aligned}&\min J(\xi(k), u(k-1), \Delta U(k)) \\ &= [\Delta U(k)^T, \varepsilon]^T H [\Delta U(k)^T, \varepsilon] + G[\Delta U(k)^T, \varepsilon]\end{aligned} \tag{34}$$

Here, $H = \begin{bmatrix} \Theta^T Q \Theta + R & 0 \\ 0 & \rho \end{bmatrix}$, $G = \begin{bmatrix} 2e^T Q \Theta & 0 \end{bmatrix}$, $e = \begin{bmatrix} \eta(k+1|t) - \eta_r(k+1|t) \end{bmatrix}$ is the tracking error within the prediction horizon.

The system's safety, stability, and efficiency can be improved by setting the control input constraints and control increment constraints in MPC reasonably:

$$u_{\min} \le u(k) \le u_{\max} \tag{35}$$

$$\Delta u_{\min} \le \Delta u(k) \le \Delta u_{\max} \tag{36}$$

4.2 Model Predictive Control Algorithm Prediction Time Horizon Optimization

In MPC, the prediction time horizon N_p determines the time span of future state prediction by the controller. When N_p is larger, the controller can predict farther future states, which helps to improve system stability. However, larger N_p also increases computational complexity and computation workload, and may introduce greater prediction errors. When N_p is smaller, the computational workload is relatively smaller, but it may reduce system stability and control effectiveness. This is because a smaller prediction time horizon may not fully consider future influencing factors, leading to less optimized control actions.

With the parking speed set to 1 m/s, and based on experience, the control time horizon N_c is determined as 20% of the prediction time horizon N_p. By selecting different N_p values, the optimized path from the starting point (8.05 m, 3.60 m) is tracked. Figure 12 shows the comparison of tracking errors under different prediction time horizons. As N_p increases, the maximum tracking error first decreases and then increases. To find the optimal N_p, the simulated annealing algorithm is used to optimize the prediction time horizon. The parameter selection is shown in Table 1.

When employing the simulated annealing algorithm for optimization, it is necessary to design a reasonable and effective fitness function to solve for the optimal parameters. In the process of parking, to reduce the path tracking error, the designed optimization fitness function is as follows:

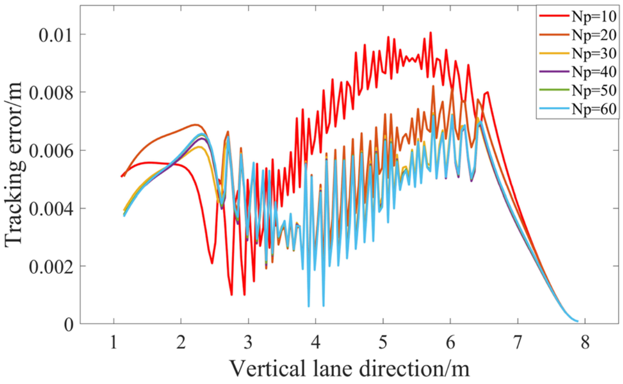


Fig. 12 Comparison of tracking errors under different prediction time horizons

$$J_2 = \int_0^t \sqrt{\left(\frac{e(t)}{\hat{E}}\right)^2}\, dt \tag{37}$$

Here, J_2 is the fitness value; t is the total parking time; $e(t)$ is the lateral error during tracking; $\hat{E}$ is the lateral error threshold, which is set to 0.35 m.

Based on the simulation results and experience, the optimization range of the prediction time horizon N_p is set to [25, 55], and the constraint conditions are set to:

$$\text{s.t.} \begin{cases} N_\mathrm{p} \in \mathbb{R} \cup N_\mathrm{p} \in [25, 55] \\ N_\mathrm{c} = \mathrm{round}\,(0.2 * N_\mathrm{p}) \end{cases} \tag{38}$$

When the simulated annealing algorithm reaches the 24th iteration of optimization, the optimal fitness value obtained is 4.021, and the optimized prediction time horizon N_p is 46, with the corresponding control time horizon N_c being 9. Table 2 shows the comparison of the maximum tracking error and the total tracking error before and after optimization when the sampling time is 0.05 s. The analysis indicates that both the maximum tracking error and total tracking error for the optimized N_p are lower than those for the initial N_p, verifying the reliability of the N_p optimized through the simulated annealing algorithm.

4.3 Pure Pursuit Algorithm Preview Distance Optimization

To verify the superiority of the MPC algorithm in path tracking, a simple, easy-to-use, and highly stable pure pursuit algorithm is utilized for comparison. In the case of the pure pursuit algorithm, the selection of the preview distance is identified as an important parameter influencing tracking accuracy:

Table 2 Comparison of the maximum tracking error and total tracking error before and after optimization

Time parameter	Horizon			
	Maximum tracking error	Total tracking error	Reduction rate of maximum tracking error after optimization %	Reduction rate of total tracking error after optimization %
(46, 9)	0.0072 m	0.6119 m	–	–
(25, 5)	0.0075 m	0.6591 m	4.00	7.16
(35, 7)	0.0074 m	0.6344 m	2.70	3.55
(45, 9)	0.0072 m	0.6192 m	0	1.18
(55, 11)	0.0073 m	0.6336 m	1.37	3.42

$$\delta = \arctan\left(\frac{2le}{l_{\mathrm{d}}^2}\right) \tag{39}$$

Here, e is the lateral deviation between the vehicle axis and the preview point; l_{d} is the preview distance.

When the desired path is being tracked, if l_{d} is assigned a larger value, significant vibrations are not experienced by the vehicle, and the steering action tends to be relatively smooth. However, during the process of approaching the desired path, an increase in local tracking errors may be experienced. With the assignment of a smaller l_{d}, a lack of predictive tracking might occur, which can easily result in vibrations and may lead to deviations in vehicle tracking or even failures.

The objective function is designed to consider the magnitude of tracking error and the amplitude of front wheel angle changes. The smaller the total tracking error of the path, the higher the precision of the vehicle in following the desired path; the smaller the variance of the front wheel angle, the more gradual the changes in the front wheel angle, indicating a smoother and more stable vehicle operation. The objective function for optimizing the preview distance in the pure pursuit algorithm is designed as follows:

$$J_3 = \min(w_4 E_{\mathrm{tot}} + w_5 \sigma_\delta) \tag{40}$$

Here, E_{tot} is the total path tracking error; σ_δ is the variance of the front wheel angle; w_4 and w_5 are the weighting coefficients for E_{tot} and σ_δ. To increase impact of tracking error on the objective function, w_4 is set to 10, w_5 is set to 1.

By considering the size of the tracking error and the amplitude of the front wheel angle change, the optimal preview distance is obtained as -0.5 m through simulated annealing algorithm optimization.

5 Experimental Verification

5.1 Simulation Experiment

A certain vehicle model is selected as the research object, and its vehicle parameters are shown in Table 3.

Table 3 Vehicle parameters

Parameter	Numerical value	Unit
Length	4.155	m
Width	1.645	m
Wheelbase	2.405	m
Front suspension	0.800	m
Rear suspension	0.950	m
Minimum turning radius	4.200	m

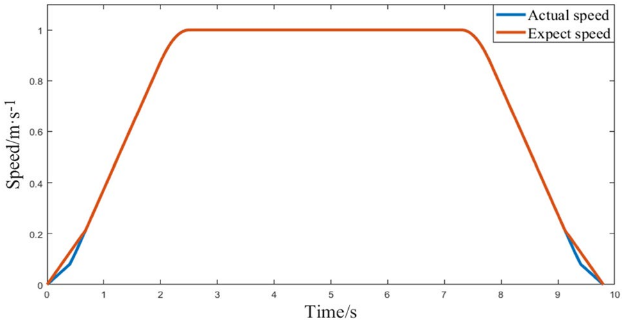
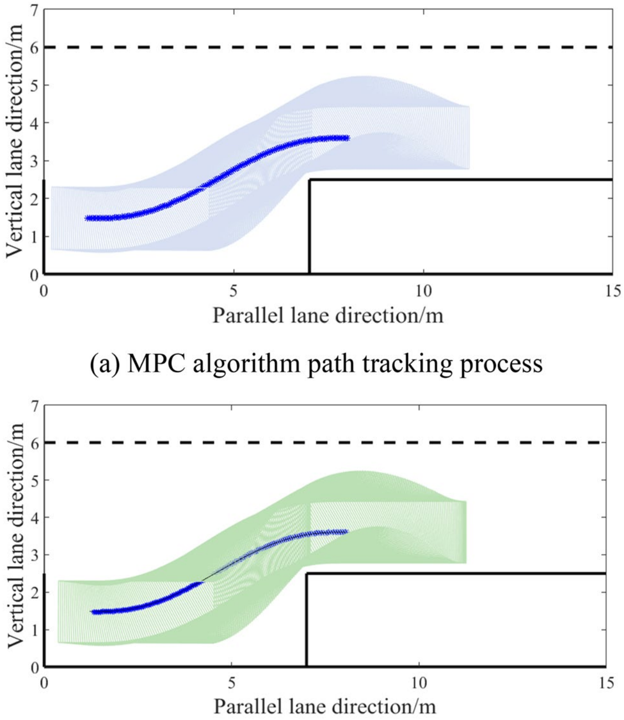


Fig. 13 Parallel parking speed tracking process

In the parallel parking simulation scenario, the parking space length is 7 m and the width is 2.5 m. The parking speed is set to 1 m/s, the prediction time horizon N_{P} is 46, the control time horizon N_{C} is 9, the sampling time is 0.05 s, and the starting coordinates for parallel parking are (8.05 m, 3.60 m), and the ending coordinates are (1.20 m, 1.48 m).

Automatic parking speed tracking requires the combined action of the drive system and the braking system. Based on the PID algorithm, longitudinal speed tracking is achieved to facilitate rapid and accurate control of the reference parking speed. As shown in Fig. 13, this method enables quick and precise tracking control of the target vehicle speed with minimal error.

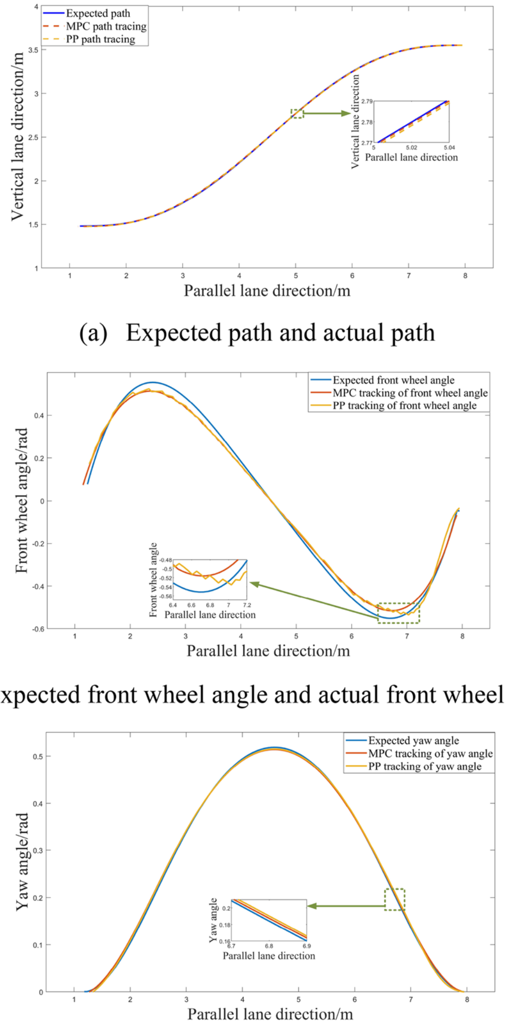
The optimized MPC algorithm, enhanced with simulated annealing algorithm for path tracking, is compared with the optimized PP algorithm. As shown in Fig. 14, the process



(a) MPC algorithm path tracking process

(b) PP algorithm path tracking process

Fig. 14 Parallel parking path tracking process



(a) Expected path and actual path

(b) Expected front wheel angle and actual front wheel angle

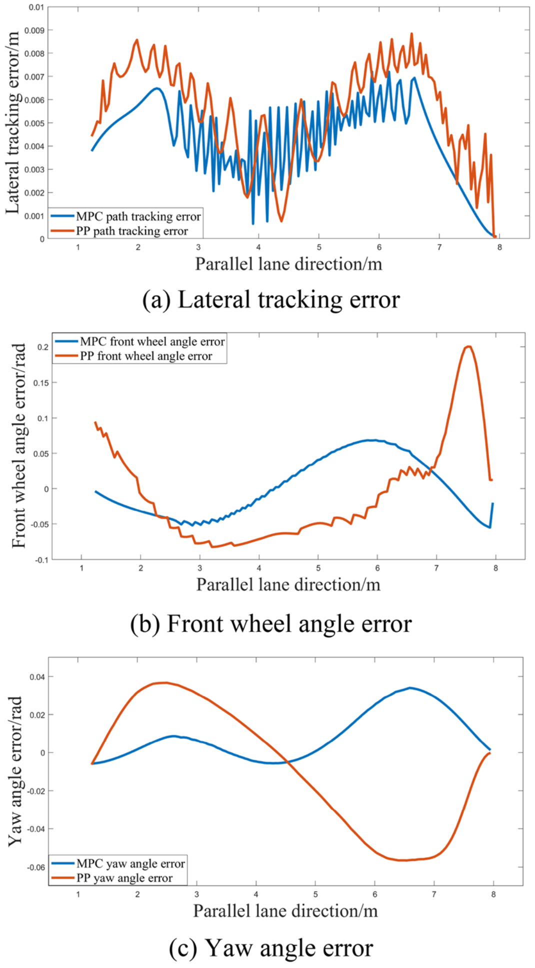
(c) Expected yaw angle and actual yaw angle

Fig. 15 Parallel parking path tracking results

of tracking the parallel parking path through the use of both algorithms manages to accurately follow the desired path. This is achieved without any collisions with road boundaries, obstacles in front of or behind the parking space, etc., during the tracking process. Upon completion of parking, the vehicle maintains its position inside the parking space, adhering to its obstacle avoidance constraints and endpoint position constraints.

As shown in Fig. 15a, both control algorithms can quickly and accurately reach the target point from the starting position while satisfying the obstacle avoidance constraints. From the local enlarged view, it can be seen that the tracking effect of MPC algorithm is better than that of PP algorithm. As shown in Fig. 15b, the maximum front wheel angle of MPC algorithm is 0.5173 rad, which does not exceed the maximum constraint value of front wheel angle of 0.5546 rad. At the same time, MPC algorithm has good tracking performance for front wheel angle, and the change of front wheel angle is smooth, without obvious mutation. Compared with PP algorithm, the maximum front wheel angle of PP algorithm is 0.5262 rad, which has good tracking performance at small front wheel angles. As shown in the local enlarged view, PP algorithm has limited tracking ability for large angles, and the front wheel angle will vibrate during tracking. As shown in Fig. 15c, the comparison between the desired yaw angle and the actual yaw angle is presented. At the parking endpoint, the yaw angles tracked by the MPC algorithm and the PP algorithm are − 0.0054 rad and − 0.0059 rad, respectively, both approaching 0, which indicates that the vehicle is parallel to the lane line at the parking endpoint. Overall, from the entire curve, the fitting degree of the yaw angle tracked based on the MPC algorithm to the desired yaw angle is higher than that of the yaw angle tracked based on the PP algorithm.

As shown in Fig. 16a, the maximum and average tracking errors of MPC algorithm in the tracking process are 0.0072 m and 0.0043 m, while the maximum and average tracking errors of PP algorithm are 0.0087 m and 0.0056 m. The lateral errors at the parking end of the two algorithms are 0.0036 m and 0.0045 m. Therefore, MPC algorithm has higher lateral tracking accuracy compared to PP algorithm. As shown in Fig. 16b, the maximum and average front wheel angle errors of MPC algorithm in the tracking process are



(a) Lateral tracking error

(b) Front wheel angle error

(c) Yaw angle error

Fig. 16 Parallel parking path tracking error

0.0685 rad and 0.0361 rad, while the maximum and average front wheel angle errors of PP algorithm are 0.2002 rad and 0.0596 rad. The front wheel angle errors at the parking end of the two algorithms are – 0.0035 rad and 0.0745 rad. The front wheel angle error change tracked by MPC algorithm is more gentle and smooth compared to PP algorithm, so MPC algorithm has higher front wheel angle tracking accuracy compared to PP algorithm. As shown in Fig. 16c, the maximum and average yaw angle errors of MPC algorithm in the tracking process are 0.0339 rad and 0.0111 rad, while the maximum and average yaw angle errors of PP algorithm are 0.0367 rad and 0.0291 rad. Therefore, MPC algorithm has higher yaw angle tracking accuracy compared to PP algorithm.

The MPC algorithm optimized by simulated annealing algorithm has better tracking performance in all aspects compared to the optimized PP algorithm. However, the MPC algorithm has a large amount of computation and requires a high amount of computing power on the processing unit.

When the actual vehicle parameters exceed those listed in Table 3, the planned path may not satisfy the obstacle avoidance constraints. To meet the obstacle avoidance constraints and enhance the universality of the proposed planning method, the maximum vehicle dimensions should be defined. Multiple simulation experiments have confirmed that when the vehicle length does not exceed 4.45 m and the vehicle width does not exceed 1.84 m, the obstacle avoidance constraints can be satisfied. Figure 17 shows the path tracking process of the MPC algorithm with a vehicle length of 4.45 m and a vehicle width of 1.84 m, where the right profile of the vehicle overlaps with the parking space line of the valet parking space, thereby validating the correctness of the maximum vehicle dimensions.

5.2 Real Vehicle Experiment

As shown in Fig. 18, the actual vehicle of the Apollo self-driving developer kit is equipped with hardware devices such as LiDAR, ultrasonic radar, camera, GPS + IMU combined inertial navigation, and other sensors.

The dimensions of the Apollo Autonomous Driving Developer Kit are 1.43 × 0.89 m. Based on the size ratio

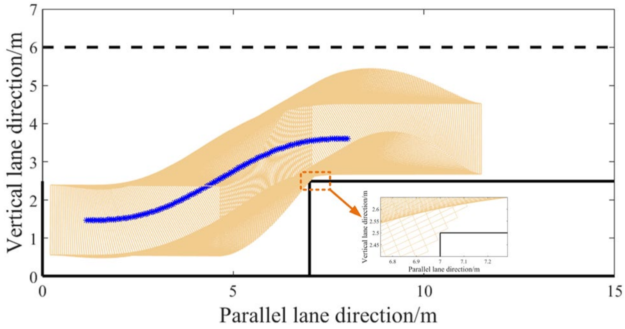


Fig. 17 The path tracking process of the MPC algorithm with a vehicle length of 4.45 m and a vehicle width of 1.84 m

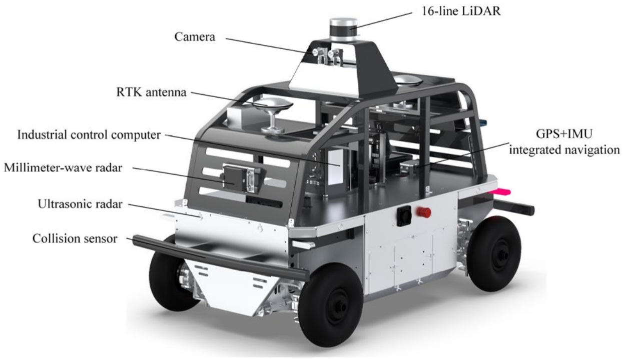


Fig. 18 Apollo autonomous driving developer kit

between the self-driving vehicle and the parking space in the simulation environment, the parallel parking space is set to be 3.0 m × 1.5 m. As shown in Fig. 19, a parallel parking space is set up in an open field, and yellow and black warning tape is used to simulate the parking space lines.

In the Dreamview's Module Controller interface, the Prediction, Routing, Planning, Control, and Canbus communication modules are started. The experimental vehicle is driven to the starting position of the parking lot, the parking destination is selected, the parking request is sent, and the parking process is started. As shown in Fig. 20, the parking process is displayed on the Dreamview interface. The red rectangle is the target parking space, and the blue solid line is the planned parking path.

At the beginning of the parking process, the Record module is turned on to record relevant data of the Apollo Autonomous Driving Developer Kit during the parking process. The parallel parking path tracking results are shown in Fig. 21. The Apollo Autonomous Driving Developer Kit can perform good tracking control on the desired path, with a maximum tracking error of 0.057 m, which is within the allowable error range. The parking process is safe, smooth, and successfully completed.

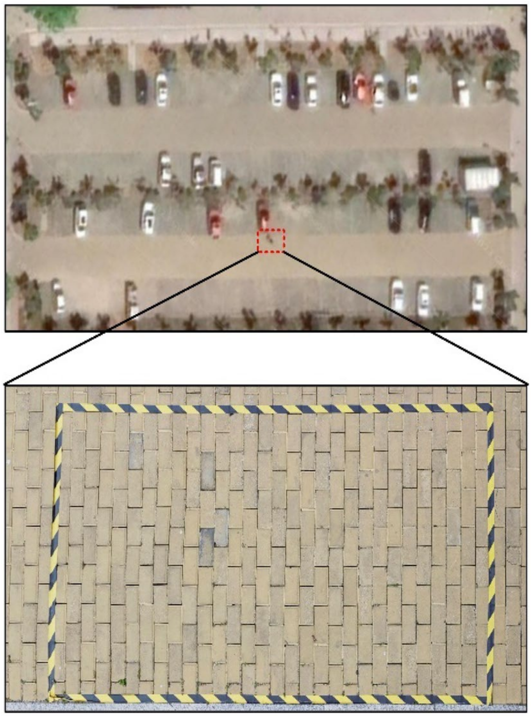
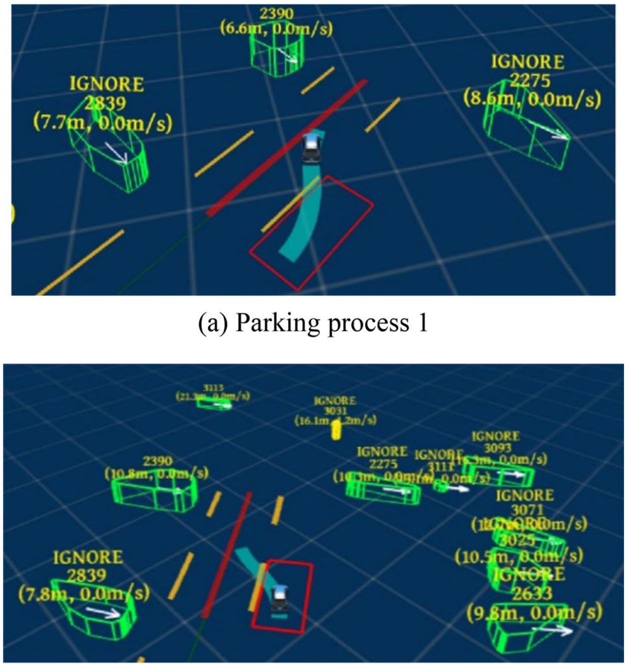


Fig. 19 Parallel parking experiment scene construction



(a) Parking process 1

(b) Parking process 2

Fig. 20 Dreamview interface parking process

6 Conclusion

A parallel parking path planning method based on a hybrid superposition curve is proposed to address the issues of maximum curvature exceeding the defined value for quintic polynomial curves and excessive curvature at the endpoints of the improved Sigmoid function. Based on the establishment of a kinematic model that satisfies position, curvature,

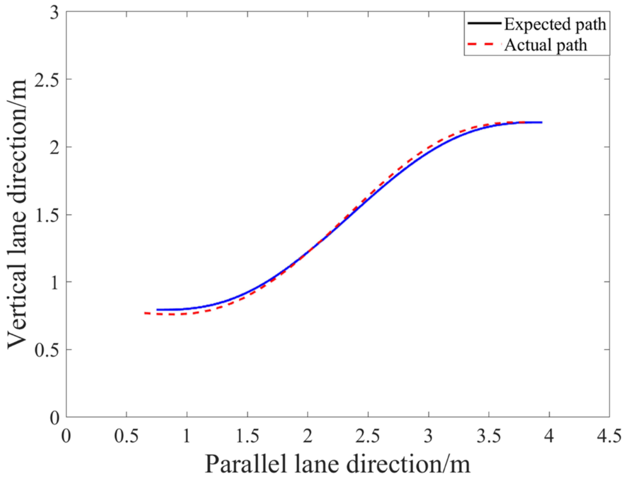


Fig. 21 Parallel parking path tracking results

and yaw angle constraints, the method optimizes the curvature at the start and end points as well as the maximum curvature during parking using simulated annealing algorithm. In addition, the simulated annealing algorithm is employed to optimize the prediction time horizon in the model predictive control algorithm. In simulation validation, the maximum parking tracking error is 0.0072 m, and the front wheel angle error and yaw angle error do not exceed 0.0685 rad and 0.0339 rad, respectively. A comparison with the optimized pure pursuit algorithm demonstrates the higher accuracy and better robustness of the model predictive control algorithm. The feasibility and effectiveness of the proposed path planning method are further verified through real-vehicle testing, where the maximum tracking error is 0.057 m.

In the future, research should be conducted on path planning methods for perpendicular parking and diagonal parking, as well as on real-time obstacle avoidance control methods for automatic parking.

Acknowledgements This research was supported by the Shandong Province graduate quality professional degree teaching case library project (Project No. SDYAL2023026), China University of Petroleum graduate education and teaching reform project (Project No. YJG2022036) and China University of Petroleum Graduate course demonstration course construction project (Project No. UPCYKS-2023-02).

Data availability The data are available from the corresponding author on reasonable request.

References

Cao, Y., Yan, J., Li, X., & Cao, L. (2023). Path search method for automatic parking system based on improved hybrid A* algorithm. *Automotive Technology, 06*, 37–41.

Chen, Q., Gan, L., Chen, B., Liu, Q., & Zhang, X. (2023). Parallel parking path planning based on improved arctangent function optimization. *International Journal of Automotive Technology, 24*, 23–33.

Chen, X., Lan, F., & Chen, J. (2021). Autonomous parking path planning based on improved deep reinforcement learning. *Journal of Chongqing University of Technology (natural Science), 35*(07), 17–27.

Erno, H., Csaba, H. and Peter, K. (2019). Novel pure-pursuit trajectory following approaches and their practical applications// 2019 1st IEEE International Conference on Gridding and Polytope Based Modeling and Control. IEEE 597–602

Gong, J., Liu, K., & Qi, J. (2020). *Second edition of model predictive control for unmanned vehicles*. Beijing Institute of Technology Press.

Helene, V., Sebastien, G., Nicoleta, M., & Said, M. (2012). Geometric path planning for automatic parallel parking in tiny spots. *IFAC Proceedings Volumes, 45*(24), 36–42.

Li, C., Jiang, H., Ma, S., Jiang, S., & Li, Y. (2020). Automatic parking path planning and tracking control research for intelligent vehicles. *Applied Sciences, 10*(24), 9100.

Liu, C. (2022). Research on vehicle parking space recognition and path planning for automatic parking. Chongqing Jiaotong University.

Naitik, N., & Maulin, J. (2018). An intelligent fuzzy based hybrid approach for parallel parking in dynamic environment-science direct. *Procedia Computer Science, 133*, 82–91.

Ren, B., Wang, X., Deng, W., Nan, J., Zong, R., & Ding, J. (2022). Automatic parking path optimization method based on hybrid A* and variable radius RS curves. *China Journal of Highway and Transport, 35*(07), 317–327.

Shin, H., Kim, M., Baek, S., Crane, C., & Kim, J. (2022). Perpendicular parking path generation and optimal path tracking algorithm for auto-parking of trailers. *International Journal of Control, Automation and Systems, 20*, 3006–3018.

Su, B., Yang, J., Li, L., & Wang, Y. (2019). Secondary parallel automatic parking of endpoint regionalization based on genetic algorithm. *Cluster Computing, 22*(03), 7515–7523.

Tang, C., Yang, J., & Wang, K. (2019). Three-layer structural mode of elevated cities and urban spatial development. *Chinese Journal of Underground Space and Engineering, 15*(03), 637–642.

Wang, W., & Hou, Z. (2022). An automatic parking scheme based on model-free adaptive control. *Control and Decision, 37*(08), 2056–2066.

Xiong, Z., Yang, W., Ding, K., Liang, F., & Zheng, L. (2017). Research on automatic parking algorithm based on preview fuzzy control. *Journal of Chongqing University of Technology (natural Science), 31*(02), 14–22.

Xu, Q., Xu, J., Hu, Y., Chen, H. and Zhang, X. (2024). Trajectory optimization of robotic arms based on improved simulated annealing genetic algorithm. Journal of System Simulation 1–10

Yu, L., Hou, Z., Wu, B., Cai, Y., & Hu, Z. (2023). Experimental design and evaluation of vertical parking path planning and tracking for unmanned vehicles. *Experimental Technology and Management, 40*, 55–64156.

Yu, L., Wang, X., Wu, B., Hou, Z., & Wu, Y. (2022). Optimization and tracking of parallel parking path planning for unmanned vehicles. *Journal of Jiangsu University (natural Science Edition), 43*(05), 519–523.

Zhang, J., Zhou, S., Liu, Y., Guo, C., & Zhao, J. (2022). Autonomous valet parking path planning based on improved rapidly-exploring random tree. *Journal of Hunan University (natural Sciences), 49*(05), 194–200.

Zhang, W., & Xiao, J. (2021). Application of improved RRT algorithm in path planning of intelligent vehicle in complex environment. *China Journal of Highway and Transport, 34*(03), 225–234.

Zhao, B., Jia, F., Cao, Y., Sun, Y., & Liu, Y. (2022a). Research on artificial potential field path planning based on simulated annealing algorithm. *Computer Engineering and Science, 44*(04), 746–752.

Zhao, X., Zhao, Y., Kang, Y., Chen, Z., & Yang, S. (2022b). A multi-UAV collaborative verification path planning method for moving ships. *Science, Technology and Engineering, 22*(28), 12518–12524.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.