IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 29, 2021, accepted June 26, 2021, date of publication July 1, 2021, date of current version July 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093930

Hierarchical Control of Trajectory Planning and Trajectory Tracking for Autonomous Parallel Parking

DUOYANG QIU[1], DUOLI QIU[2], BING WU[3], MAN GU[1], AND MAOFEI ZHU[1]

[1]School of Advanced Manufacturing Engineering, Hefei University, Hefei 230601, China
[2]School of Architecture, Huaibei Normal University, Huaibei 235000, China
[3]School of Automotive and Traffic Engineering, Hefei University of Technology, Hefei 230009, China

Corresponding author: Duoli Qiu (qiuduoli@sina.com)

This work was supported in part by the Major Special Science and Technology Project of Anhui Province, China, under Grant 201903a05020033, and in part by two Key Projects of University Natural Science Foundation of Anhui Province, China, under Grant KJ2019A0959 and Grant KJ2020A1065, and in part by the Key Project of Hefei University, China, under Grant 20ZR02ZDB.

ABSTRACT For the parallel parking problem in narrow space, this paper proposes a trajectory tracking control method with a novel trajectory planning layer for autonomous parallel parking based on a numerical optimization algorithm and model predictive control. In the trajectory planning layer, the vehicle kinematics model suitable for the low-velocity parking scene is established. Considering the vehicle physical constraints, boundary condition constraints, and obstacle avoidance constraints during the parking process, the parking trajectory planning task is described as an optimal control problem, further transformed into a nonlinear programming problem by Gauss pseudo-spectral method. Taking the shortest parking completion time as the optimization objective function, the parking trajectories of the large, medium and small parking spaces are obtained, respectively. A parking trajectory tracking controller based on the model predictive control algorithm is designed in the trajectory tracking control layer. The linear error model is used as the prediction model, and the quadratic programming is adopted as the rolling optimization algorithm in the tracking controller. The velocity and front-wheel swing angle are obtained as control signals for parking trajectory tracking. Through CarSim and Simulink's co-simulation, the feasibility and effectiveness of the proposed parallel parking trajectory planning and tracking control method are verified. The co-simulation results show that the maximum tracking errors of horizontal and longitudinal positions are less than 0.15m. The maximum tracking errors of heading angle are less than 2° under three different parking spaces. Real vehicle tests are carried out to verify the effectiveness of the proposed hierarchical control method. The test results show that the vehicle can park in the parking space safely, quickly and accurately when the actual parking space is detected. The proposed method can plan the parking trajectory with the constraints and the shortest time and control the vehicle to complete the parking operation accurately along the planned trajectory.

INDEX TERMS Trajectory planning, trajectory tracking, parallel parking, Gauss pseudo-spectral, model predictive control.

I. INTRODUCTION

With the continuous improvement of automobile intelligence and network, assisted driving and autonomous driving technology have become the research hotspot of major automobile manufacturers at home and abroad. As the key technical link of autonomous driving technology, autonomous parking technology can not only effectively solve the problem of parking difficulty caused by the lack of driver's experience or technology in narrow space, but also significantly improve the efficiency, safety, and comfort of parking. Therefore, the research and development of autonomous parking systems have fundamental practical significance.

An autonomous parking system includes environment perception, decision-planning, control execution, and information interaction modules. Among them, the

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

decision-planning module mainly uses the information obtained by the perception device or network to make decisions and judgments. It then plans the best reference trajectory or path for vehicles. It is the vehicle's brain, the key to the success of the destination, and the core of the autonomous parking system. It is worth mentioning that the difference between trajectory and path is that the planning result of the former is related to time, while the latter has nothing to do with time parameters. However, both of them belong to the category of motion planning. Many scholars and research institutions have carried out a lot of research in terms of motion planning methods. According to their implementation in automated driving, these planning techniques were classified into six groups: geometry method, graph search method, random sampling method, curve interpolation method, machine learning method, and optimization algorithm, such as dynamic optimization. The most relevant planning algorithms implemented in motion planning for automated driving are described below. In literature [1]–[4], according to the geometric relationship among vehicles, surrounding obstacles, and target parking spaces, the parking path was obtained by geometric methods such as multi-section arc or tangent arcs and straight lines. The advantages were simple algorithms and high solving efficiency, but the discontinuous curvature of the planned path led to difficulty tracking control. The spline curve method proposed in the literature [5] and [6] was one of the most commonly used methods to generate a path. Literature [7], [8] adopted improved heuristic graph search algorithms to generate safe and smooth autonomous driving paths on complex unstructured roads. In reference [9], the Depth First Search (DFS) algorithm was used to plan the optimal path based on processing the vertices of the directed graph. The path needed to be refined to achieve higher accuracy and be suitable for tracking control. Literature [10]–[12] adopted rapid exploration random tree (RRT) to generate required local paths within the permitted driving area while ensuring real-time performance in a dynamic environment. The ability of this algorithm to deal with complex constraints needed to be further improved. In literature [13]–[17], the curve interpolation methods such as Bezier curve, clothoid curve, spline curve, and arctangent polynomial curve were respectively used for parking path planning. The motion path generated by this method has good continuity, smoothness, and traceability. However, it had strict requirements on the size of parking space. Literature [18]–[20] respectively applied Reinforcement Learning (RL), special deep neural network, Midden Markov Model (HMM), and other machine learning methods to motion planning. Their advantages were fast calculation speed and specific generalization ability, but a large number of training samples required to be collected.

The state and control parameter values of the planned trajectory or path will be used as the reference in the control execution module. The function of the control execution model is to use an advanced algorithm to control the steering wheel, accelerator, brake, and other actuators so as to achieve the purpose of driving, braking, and steering, and finally make the actual driving trajectory or path of the vehicle consistent with the planned trajectory or path as much as possible. In terms of trajectory tracking control algorithm, reference [21]–[23] designed path tracking controllers based on feedback linearization control and Sliding Mode Control (SMC), respectively. In reference [24], the Linear Quadratic Regulator (LQR) was used to realize the closed-loop optimal control of path tracking for the parking system. In references [25], [26], the path tracking control method based on nonlinear model prediction was applied to the path tracking of agricultural vehicles. In reference [27], this method was further combined with PID control technology as a hybrid control strategy for tracking control of intelligent vehicles. The computational efficiency and adaptability of the path tracking control algorithm based on preview theory were further improved in reference [28]. The above planning and tracking algorithms are of great positive significance to the realization of the planning layer and tracking layer of unmanned vehicles. In terms of state estimation, literature [29] proposed a dual unscented Kalman filter (DUKF) approach, where two UKFs run in parallel to simultaneously estimate vehicle states and parameters.

Gauss pseudo-spectral Method (GPM) can transform the continuous optimal trajectory problem into a nonlinear programming problem on the basis of discretization of variables. The interior-point method is further adopted to solve the nonlinear programming problem. Many advantages, such as fast convergence speed and low sensitivity to the initial value, make this method more suitable for autonomous parking trajectory planning. The application of model predictive control (MPC) is more and more in control scenarios, which requires less model accuracy, but has better control stability and accuracy. It is more suitable for solving the trajectory tracking control of autonomous parking with obvious features, such as low speed, a significant change of reference path curvature, and heading; many constraints exist in the solution process. Under the MPC framework, the longitudinal and lateral control can be transformed into the same optimization problem with constraints to fully consider the coupling effect of vehicle motion. Literature [29] proposed two control architectures based on MPC for vehicle obstacle avoidance. One was to realize planning and control in double layers, and the other was to realize planning and control in the same MPC controller. The advantage of hierarchical control is that the planning can adopt a relatively simple model, and the amount of calculation is relatively small, while the integrated control can avoid planning the infeasible trajectory, but the real-time application is more complicated. This paper proposes a GPM-MPC based hierarchical control method for autonomous parking trajectory planning and tracking according to the above architecture.

The entire autonomous parking system is built from several modules, and simplified system architecture is illustrated in Fig.1. The trajectory planning and tracking control layers are marked with blue and green sub modules respectively,

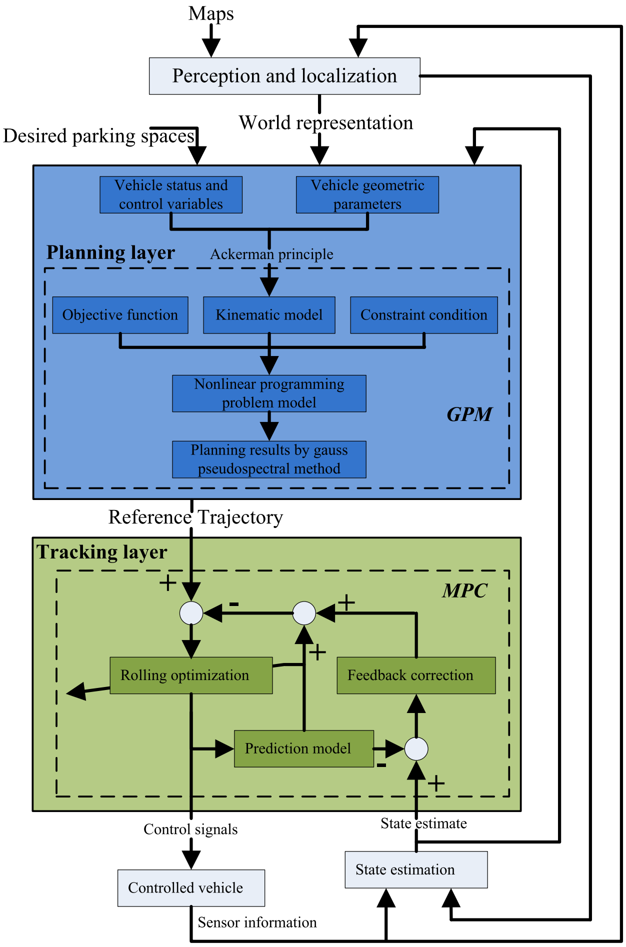


FIGURE 1. The simplified architecture of the autonomous parking system.

where the method of planning and trajectory tracking control are considered as the main contributions of this work. The main work of this paper is as follows: Firstly, the kinematics model suitable for low-velocity parking scene is established. Considering the physical constraints, obstacle avoidance constraints, and boundary constraints of parking conditions, the parking trajectory planning task is described in the form of an optimal control problem, which is further transformed into a nonlinear programming problem by GPM. Taking the shortest parking completion time as the optimization objective function, the parking trajectories of large, medium and small parking spaces are solved respectively by the interior-point method. In the trajectory tracking controller based on the MPC algorithm, the center point of the rear-axle of the vehicle is taken as the state observation point, and the front wheel angle and velocity are taken as the control variables. Firstly, the nonlinear kinematics model is linearized and discretized, and then the state prediction model is established. In the prediction horizon, the quadratic programming algorithm is adopted to solve the control parameter sequence. The CarSim and Simulink co-simulation platforms are used to verify the trajectory tracking effect, and the deviation between the actual output by CarSim and the output by the prediction model is taken as the feedback quantity to correct the MPC further.

II. ANALYSIS OF VEHICLE PARKING MOVEMENT PROCESS

A. ESTABLISHMENT OF VEHICLE KINEMATICS MODEL

Under the condition of autonomous parking on a flat road, the dynamic characteristics such as sideslip caused by tire lateral force can be ignored due to the low driving velocity. Instead, vehicle kinematics should be considered. According to Ackerman's front-wheel steering principle, a pure rolling vehicle kinematics model is established in the ground coordinate system with the center point of the vehicle's rear-axle as the state observation point for autonomous parking trajectory planning, as shown in Fig. 2.

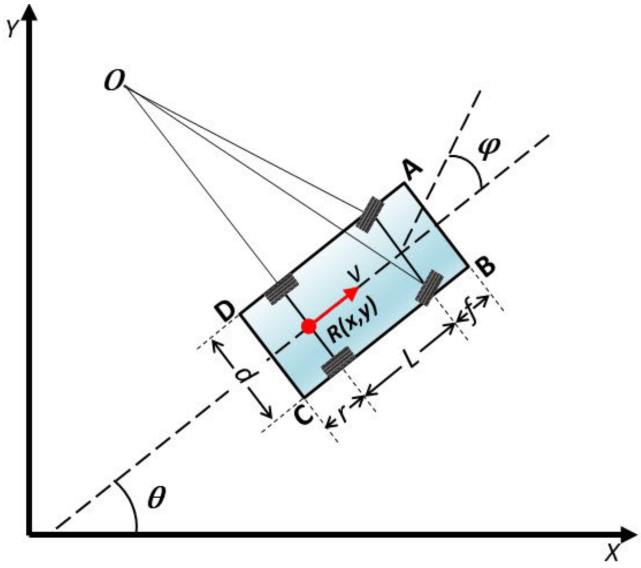


FIGURE 2. Vehicle kinematics model.

In the figure, L, f, r and d respectively denote vehicle wheelbase, front suspension, rear suspension and width; (x, y) is the mid-point of rear-wheel axis(see the reference point R in figure 2); v denotes the linear velocity of point R; O denotes the instantaneous rotation center of vehicle; φ denotes the steering angle of front wheels; θ denotes the heading angle; A, B, C and D are the four contour vertices of vehicle body. Kinematics of a front-steering automobile is described through differential equations:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ a(t) \\ \theta(t) \\ \varphi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cdot\cos\theta(t) \\ v(t)\cdot\sin\theta(t) \\ a(t) \\ j(t) \\ v(t)\cdot\tan\varphi(t)/L \\ \omega(t) \end{bmatrix}, \quad t \in [t_0, t_f] \qquad (1)$$

Herein: a and j stand for vehicle acceleration and the change rate of acceleration; t_0 and t_f refer to the starting and finishing time of parking respectively; ω refers to the angular velocity corresponding to the φ. According to the vehicle dimensions, the position of the rear axle center point and the geometric relationship, the horizontal and vertical coordinates of 4 contour vertices A, B, C and D can be deduced as follows:

$$\begin{bmatrix} A_x(t) \\ A_y(t) \\ B_x(t) \\ B_y(t) \\ C_x(t) \\ C_y(t) \\ D_x(t) \\ D_y(t) \end{bmatrix} = \begin{bmatrix} x(t) + (L+f) \cdot \cos\theta(t) - \frac{d}{2} \cdot \sin\theta(t) \\ y(t) + (L+f) \cdot \sin\theta(t) + \frac{d}{2} \cdot \cos\theta(t) \\ x(t) + (L+f) \cdot \cos\theta(t) + \frac{d}{2} \cdot \sin\theta(t) \\ y(t) + (L+f) \cdot \sin\theta(t) - \frac{d}{2} \cdot \cos\theta(t) \\ x(t) - r \cdot \cos\theta(t) + \frac{d}{2} \cdot \sin\theta(t) \\ \mathrm{y}(t) - r \cdot \sin\theta(t) - \frac{d}{2} \cdot \cos\theta(t) \\ x(t) - r \cdot \cos\theta(t) - \frac{d}{2} \cdot \sin\theta(t) \\ \mathrm{y}(t) - r \cdot \sin\theta(t) + \frac{d}{2} \cdot \cos\theta(t) \end{bmatrix}, \quad t \in [t_0, t_f] \tag{2}$$

B. CONSTRAINT CONDITION

1) VEHICLE PHYSICAL CONSTRAINTS

In the process of autonomous parking, the physical and mechanical constraints of the vehicle need to be satisfied, which are transformed into inequality constraints on some state variables and control variables as follows:

$$\begin{cases} |v(t)| \le v_{\max} \\ |a(t)| \le a_{\max} \\ |j(t)| \le j_{\max} \qquad t \in [t_0, t_f] \\ |\varphi(t)| \le \varphi_{\max} \\ |\omega(t)| \le \omega_{\max}, \end{cases} \tag{3}$$

where $v_{\max}, a_{\max}, j_{\max}, \varphi_{\max}$ and $\omega_{\max}$ refer to the maximum velocity, the maximum acceleration, the maximum change rate of acceleration, the maximum steering angle of the front wheel, the corresponding maximum angular velocity allowed in the parking process.

2) BOUNDARY CONDITION CONSTRAINTS

In the case of ignoring the dynamic obstacles such as pedestrians and other vehicles, the external constraint during the autonomous parking process is the boundary condition constraint, which should be considered first. Taking parallel parking as an example, the schematic diagram of the autonomous parking area is shown in Fig.3, where L_p and W_p refer to the length and width of parking spaces; W_r denotes the width of the road.

At the starting and finishing time of autonomous parking, the vehicle is considered to be in a static state, and the steering angle of the front wheel is zero. The coordinate of the point R at the starting time is (x_0, y_0) and the heading angle is θ_0. When parking is completed, the state is changed to that the coordinate of point R is (x_f, y_f) and the heading angle is zero.

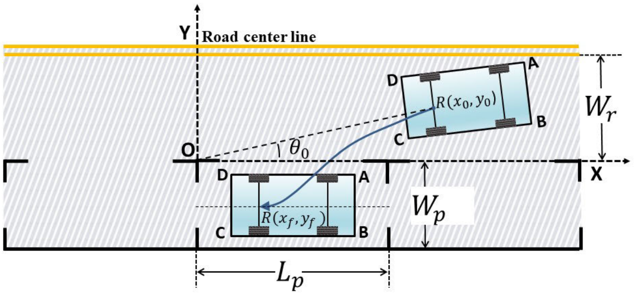


FIGURE 3. Schematic diagram of autonomous parallel parking area.

The four contour vertices of the vehicle are strictly required to be within the parking space. To sum up, the state variables, control variables and contour vertex coordinates should meet the equation (4) and inequality (5):

$$\begin{bmatrix} v(t_0) & v(t_f) \\ a(t_0) & a(t_f) \\ j(t_0) & j(t_f) \\ \omega(t_0) & \omega(t_f) \\ \varphi(t_0) & \varphi(t_f) \\ \theta(t_0) & \theta(t_f) \\ x(t_0) & x(t_f) \\ y(t_0) & y(t_f) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \theta_0 & 0 \\ x_0 & x_f \\ y_0 & y_f \end{bmatrix} \tag{4}$$

$$\begin{cases} 0 < A_x(t_f) < L_p, -W_p < A_y(t_f) < 0 \\ 0 < B_x(t_f) < L_p, -W_p < B_y(t_f) < 0 \\ 0 < C_x(t_f) < L_p, -W_p < C_y(t_f) < 0 \\ 0 < D_x(t_f) < L_p, -W_p < D_y(t_f) < 0 \end{cases} \tag{5}$$

3) OBSTACLE AVOIDANCE CONSTRAINTS

In addition to the above constraints, the vehicle should avoid collision with obstacles and road edges in the parking area during the whole parking time domain $[t_0, t_f]$. The contour of the vehicle is similar to that of the rectangle, so it is better to describe the vehicle as a rectangle for parking scenes with small activity area. Obstacle avoidance idea is shown in Fig. 4. Taking vehicle vertex A and the point P_4 of the obstacle for example, the constraint requires that all points of the obstacle composed of P_1, P_2, P_3, P_4 do not contact the point A, and the four vertices A, B, C, D should strictly located outside the obstacle. The above constraint is expressed by the following inequality:

$$\begin{cases} S_{\Delta AP_1P_2} + S_{\Delta AP_2P_3} + S_{\Delta AP_3P_4} + S_{\Delta AP_1P_4} > \alpha \cdot S_{P_1P_2P_3P_4} \\ S_{\Delta P_4AB} + S_{\Delta P_4BC} + S_{\Delta P_4CD} + S_{\Delta P_4DA} > \alpha \cdot S_{ABCD} \end{cases} \tag{6}$$

Herein: S_Δ refer to the area of triangle; $S_{P_1P_2P_3P_4}$ refer to the area of rectangle P_1, P_2, P_3, P_4; S_{ABCD} refer to the area of rectangle A, B, C, D; α refer to the safety factor greater than 1. The area of rectangle can be calculated by vehicle dimension parameters and the area of triangle can be calculated by its three vertex coordinates. For example, the area of ΔP_iAB can

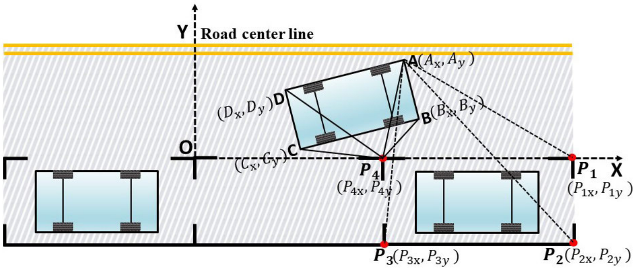


FIGURE 4. Diagram of obstacle avoidance constraints in parking process.

be calculated by following equation:

$$S_{\Delta P_i AB} = \frac{1}{2} \cdot [P_{ix} \cdot A_y + A_x \cdot B_y + B_x \cdot P_{iy} - P_{ix} \cdot B_y - A_x \cdot P_{iy} - B_x \cdot A_y] \tag{7}$$

The collision between vehicle and parking obstacles should be avoided, so as to the edge of the road. To avoid collision with the road edge, it is equivalent to that the four vertices of the vehicle are all on one side of the road edge, as the following inequality:

$$\begin{cases} -L_p \le A_x(t) \le 2 \cdot L_p, -W_p \le A_y(t) \le W_r \\ -L_p \le B_x(t) \le 2 \cdot L_p, -W_p \le B_y(t) \le W_r \\ -L_p \le C_x(t) \le 2 \cdot L_p, -W_p \le C_y(t) \le W_r \\ -L_p \le D_x(t) \le 2 \cdot L_p, -W_p \le D_y(t) \le W_r, \end{cases} \quad t \in [t_0, t_f] \tag{8}$$

III. AUTONOMOUS PARKING TRAJECTORY PLANNING BASED ON GPM

A. DESCRIPTION OF AUTONOMOUS PARKING PROBLEM

Without losing generality, the Bolza problem of nonlinear optimal control can be described as:

$$\min \psi = \phi(\xi(t_0), t_0, \xi(t_f), t_f) + \int_{t_0}^{t_f} g(\xi(t), \mu(t), t)dt \tag{9}$$

$$s.t. \begin{cases} \dot{\xi}(t) = f(\xi(t), \mu(t), t) \\ M(\xi(t), \mu(t), t) \le 0 \\ N(\xi(t_0), t_0, \xi(t_f), t_f) = 0 \end{cases} \tag{10}$$

where: $\xi(t)$ and $\mu(t)$ are the state variables and control variables of the system respectively, and satisfy the following conditions: $\xi(t) \in R^n$, $\mu(t) \in R^m$; $\min \psi$ is the objective function of the optimal control problem. In equation (10), the first row describes the state differential equation of the system; The second row is the inequality constraint of system variables in the process of solving the objective function; The third row is the corresponding equality constraint. Trajectory planning of autonomous parking is actually a generalized Bolza problem of nonlinear optimal control. In order to improve the parking efficiency, reduce the road occupation time and ensure the road capacity, the objective function of the optimal control problem is set to achieve the shortest parking time, that is $\min t_f$. Combined with the constraints above, the nonlinear optimal control problem of autonomous parking trajectory planning can be described as the following mathematical model:

$$\min t_f \tag{11}$$

$$s.t. \begin{cases} \text{Equation (1)} \\ \text{Equation (3), (5), (6), (8)} \\ \text{Equation (4)} \end{cases} \tag{12}$$

B. SOLUTION OF PARKING TRAJECTORY BASED ON GPM

GPM is a direct method for solving nonlinear optimal control problems. The continuous optimization problem is discretized at the Legendre-Gauss (LG) collocation point, and Lagrange interpolation polynomials are constructed to approximate the state variables and control variables. Algebraic constraints are used to replace the constraints of the differential equation. As a result, the optimal control problem can be transformed into a Nonlinear Programming (NLP) problem. The advantages of GPM are high adaptability to initial value, fast convergence rate and high accuracy.

1) TIME DOMAIN TRANSFORMATION

The starting and ending time range of autonomous parking $[t_0, t_f]$ is divided into p subintervals, namely $\left\{[t_0, t_1], [t_1, t_2], \cdots, [t_{q-1}, t_q], \cdots, [t_{p-1}, t_p]\right\}, t_p = t_f, q \in [1, p]$. In GPM algorithm, the interval of discrete points of the time variable is $\tau \in [-1, 1]$, so it is necessary to transform each sub interval in time domain. Any time $t \in [t_{q-1}, t_q]$ can be transformed into $\tau \in [-1, 1]$ by equation (13).

$$\tau = \frac{2 \cdot t - (t_{q-1} + t_q)}{t_q - t_{q-1}} \tag{13}$$

Based on time domain transformation, the optimal control problem in the subinterval $[t_{q-1}, t_q]$ can be described as follows:

$$\min \psi^{(q)} = \phi^{(q)}(\xi^{(q)}(\tau_0), t_{q-1}, \xi^{(q)}(\tau_f), t_q) + \frac{t_q - t_{q-1}}{2} \cdot \int_{\tau_0}^{\tau_f} g(\xi^{(q)}(\tau), \mu^{(q)}(\tau), \tau : t_{q-1}, t_q)d\tau \tag{14}$$

$$s.t. \begin{cases} \dot{\xi}^{(q)}(\tau) = \frac{t_q - t_{q-1}}{2} \cdot f(\xi^{(q)}(\tau), \mu^{(q)}(\tau), \tau : t_{q-1}, t_q) \\ M(\xi^{(q)}(\tau), \mu^{(q)}(\tau), \tau : t_{q-1}, t_q) \le 0 \\ N(\xi^{(q)}(\tau_0), \tau_0, \xi^{(q)}(\tau_f), \tau_f) = 0 \end{cases} \tag{15}$$

where, $\xi^{(q)}(\tau)$ and $\mu^{(q)}(\tau)$ refer to the state variables and control variables which are obtained by transforming the time t into τ in the q-th subinterval.

2) DISCRETIZATION OF VARIABLES

The state variables and control variables are discretized at the Legendre-Gauss (LG) collocation point, and the Lagrange polynomial is constructed to approximate the state variables ξ and control variables μ. In the interval $[\tau_0, \tau_f]$, the Legendre polynomials of order N are:

$$P_N(\tau) = \frac{1}{2^N \cdot N!} \cdot \frac{d^N}{d\tau^N} \cdot [(\tau^2 - 1)^N] \qquad (16)$$

$N + 1$ collocation points are made up of zero point $(\tau_1, \tau_2, \cdots, \tau_N)$ of equation (16) and the initial point $\tau_0 = -1$, the state variables and control variables at each collocation point are as follows:

$$\begin{cases} \xi(\tau_0), \xi(\tau_1), \cdots, \xi(\tau_N) \\ \mu(\tau_0), \mu(\tau_1), \cdots, \mu(\tau_N) \end{cases} \qquad (17)$$

Lagrange interpolation polynomial is constructed at the collocation point to interpolate state variables and control variables. The Lagrange basis functions are as follows:

$$\begin{cases} L_i^{(p)}(\tau) = \Pi_{j=0,j\neq i}^{N} \dfrac{\tau - \tau_j^{(p)}}{\tau_i - \tau_j^{(p)}} \\ \tilde{L}_i^{(p)}(\tau) = \Pi_{j=1,j\neq i}^{N} \dfrac{\tau - \tau_j^{(p)}}{\tau_i - \tau_j^{(p)}} \end{cases} \qquad (18)$$

According to the equations (16)-(18), the approximate polynomials of state variables and control variables in the subinterval $[t_{q-1}, t_q]$ can be obtained, which are expressed as follows:

$$\begin{cases} \xi^{(q)}(\tau) \approx \sum_{i=0}^{N} L_i^{(p)}(\tau) \cdot \zeta^{(q)}(\tau_i) \\ \mu^{(q)}(\tau) \approx \sum_{i=0}^{N} \tilde{L}_i^{(p)}(\tau) \cdot U^{(q)}(\tau_i) \end{cases} \qquad (19)$$

The differential of the approximate polynomial can be used to approximate the differential of the state variable by taking the derivative of Equation (19), such as the derivative of the state variable at the point τ_k can be expressed as:

$$\dot{\xi}^{(p)}(\tau_k) \approx \sum_{i=0}^{N} \dot{L}_i^{(p)}(\tau_k) \cdot \zeta^{(q)}(\tau_i) \qquad (20)$$

Combined with Equations 14, 15 and 20, the kinematic differential equation constraint can be transformed into algebraic equation constraint:

$$\dot{\xi}^{(p)}(\tau_k) - \frac{t_q - t_{q-1}}{2} \cdot f(\zeta(\tau_k), U(\tau_k), \tau_k) = 0 \qquad (21)$$

3) CONSTRAINTS UNDER DISCRETE CONDITIONS

The equality and inequality constraints of the system variables are as follows with discretization:

$$\begin{cases} M(\zeta^{(q)}(\tau_k), U^{(q)}(\tau_k), \tau_k : t_{q-1}, t_q) \leq 0 \\ N(\zeta^{(q)}(\tau_0), \tau_0, \zeta^{(q)}(\tau_f), \tau_f) = 0 \end{cases} \qquad (22)$$

The curvature of the trajectory needs to be continuous to adapt to the actual parking process, so the state variables should also be continuous correspondingly. In short, the value of the state variables at the end of each sub interval is required to be the same as that at the initial time of the next sub interval. However, the change of the control variables could be discontinuous. The specific constraints are shown in equation (23):

$$\zeta^{(q-1)}(\tau_f) = \zeta^{(q)}(\tau_0) \qquad (23)$$

It can be seen from Equation (17) that the LG collocation point does not include the terminal time of each subinterval. Therefore, the value of terminal state can be approximated by Gauss integral according to equation (23), as follows:

$$\zeta^{(q)}(\tau_f) \approx \zeta^{(q)}(\tau_0) + \frac{t_q - t_{q-1}}{2} \sum_{k=0}^{N} \rho_k \cdot f(\zeta(\tau_k), U(\tau_k), \tau_k) \qquad (24)$$

where, the ρ_k denotes the weight coefficient of Gaussian integral.

4) THE OBJECTIVE FUNCTION IN THE DISCRETE CASE

Through the above discretization process, the optimal control problem of autonomous parking trajectory planning can be transformed into a nonlinear programming problem. By approximating the integral term of the objective function by Gauss integral, the objective function in the discrete case can be obtained:

$$\begin{aligned} \min \psi^{(q)} &= \phi^{(q)}(\xi^{(q)}(\tau_0), t_{q-1}, \xi^{(q)}(\tau_f), t_q) \\ &\quad + \frac{t_q - t_{q-1}}{2} \cdot \sum_{k=0}^{N} \rho_k \cdot g(\zeta(\tau_k), U(\tau_k), \tau_k) \end{aligned} \qquad (25)$$

IV. PARKING TRAJECTORY TRACKING CONTROL BASED ON MPC

MPC consists of three modules: predictive model, rolling solution and feedback correction. The advantages of the algorithm are a low requirement on model precision, good robustness and high stability. It is more suitable for solving the problem of trajectory tracking control of autonomous parking with obvious features, such as low speed; a large change of reference path curvature and heading; many constraints exist in the solution process.

A. LINEARIZATION OF STATE DIFFERENTIAL EQUATIONS

It can be seen from equation (1) that the trajectory tracking of autonomous parking can be regarded as a nonlinear control system with control variables $\mu = [v, \varphi]^T$ and state variables $\xi = [x, y, \theta, a, j, \omega]$, and its state differential equation is expressed as $\dot{\xi} = f(t, \xi, \mu)$. It is necessary to linearize the state differential equation and use the linearized linear error model as the MPC prediction model. The state differential equation is expanded by the Taylor series at the reference point (ξ_r, μ_r). Ignoring the higher order term, the following equation can be obtained with the only first-order term:

$$\begin{aligned} \dot{\xi} \approx f(t_r, \xi_r, \mu_r) &+ \left.\frac{\partial f(t, \xi, \mu)}{\partial \xi}\right|_{\substack{\xi = \xi_r \\ \mu = \mu_r}} (\xi - \xi_r) \\ &+ \left.\frac{\partial f(t, \xi, \mu)}{\partial \mu}\right|_{\substack{\xi = \xi_r \\ \mu = \mu_r}} (\mu - \mu_r) \end{aligned} \qquad (26)$$

By subtracting equation (26) from the planned state parameters trajectory $\dot{\xi}_r = f(t_r, \xi_r, \mu_r)$, the state error model can be obtained:

$$\dot{\tilde{\xi}} = \dot{\xi} - \dot{\xi}_r = A(t) \cdot \tilde{\xi} + B(t) \cdot \tilde{\mu} \qquad (27)$$

Herein:

$$A(t) = \left.\frac{\partial f(t,\xi,\mu)}{\partial \xi}\right|_{\substack{\xi=\xi_r \\ \mu=\mu_r}}, \quad B(t) = \left.\frac{\partial f(t,\xi,\mu)}{\partial \mu}\right|_{\substack{\xi=\xi_r \\ \mu=\mu_r}}$$

refer to the Jacobian matrices of function f relative to ξ and μ; $\tilde{\xi} = \xi - \xi_r, \tilde{\mu} = \mu - \mu_r$.

B. DISCRETIZATION OF STATE DIFFERENTIAL EQUATIONS

Equation (27) is a linearized equation of state and a linear time-varying model. However, it cannot be directly used in the design of model prediction controller due to its continuity. According to the forward Euler equation, the difference quotient is adopted instead of the differential, which can be discretized as follows:

$$A(k)\cdot\tilde{\xi}(k) + B(k)\cdot\tilde{\mu}(k) = \frac{\tilde{\xi}(k+1) - \tilde{\xi}(k)}{T} \tag{28}$$

$$\tilde{\xi}(k+1) = (I + T\cdot A(k))\cdot\tilde{\xi}(k) + T\cdot B(k)\cdot\tilde{\mu}(k) \tag{29}$$

$$\tilde{\xi}(k+1) = A_{kin}(k)\cdot\tilde{\xi}(k) + B_{kin}(k)\cdot\tilde{\mu}(k) \tag{30}$$

Herein: T is the sampling time.

C. PREDICTION EQUATIONS

According to the discrete and linearized model of equation (30), state variables and control variables are combined to make the following settings:

$$\begin{cases} \varepsilon(k) = \begin{bmatrix} \tilde{\xi}(k) \\ \tilde{\mu}(k-1) \end{bmatrix} \\ \tilde{A}(k) = \begin{bmatrix} A_{kin}(k) & B_{kin}(k) \\ 0_{m\times n} & I_m \end{bmatrix} \\ \tilde{B}(k) = \begin{bmatrix} B_{kin}(k) \\ I_m \end{bmatrix} \\ \tilde{C}(k) = \begin{bmatrix} C(k) & 0_{1\times m} \end{bmatrix} \end{cases} \tag{31}$$

Herein: I_m is the identity matrix, m and n are the number of control parameters and state parameters respectively. The lateral displacement x, longitudinal displacement y and heading angle θ of the rear-axle center vehicle are taken as the system output, the corresponding $C(k) = [1\ 1\ 0\ 0\ 1\ 0]$. According to Equations (30) and (31), the incremental equation of state can be used as the prediction equation. The derivation process is as follows:

$$\varepsilon(k+1) = \begin{bmatrix} \tilde{\xi}(k+1) \\ \tilde{\mu}(k) \end{bmatrix} = \begin{bmatrix} A_{kin}(k)\tilde{\xi}(k) + B_{kin}(k)\cdot(\tilde{\mu}(k-1) + \Delta\mu(k)) \\ \tilde{\mu}(k-1) + \Delta\mu(k) \end{bmatrix} \tag{32}$$

$$\Delta\mu(k) = \tilde{\mu}(k) - \tilde{\mu}(k-1) \tag{33}$$

$$\begin{cases} \varepsilon(k+1) = \tilde{A}(k)\varepsilon(k) + \tilde{B}(k)\Delta\mu(k) \\ \eta(k) = \tilde{C}(k)\varepsilon(k) \end{cases} \tag{34}$$

Herein: $\eta(k)$ is the output of the system at time k. The prediction horizon and control horizon are set to N_p and $N_c(N_c \le N_p)$ respectively. As a result, the system output sequence Y in the prediction horizon can be calculated by the following equation:

$$Y(k) = \Omega_k\varepsilon(k\,|k) + \Theta_k\Delta\mu(k) \tag{35}$$

Herein $Y(k)$, Θ_k, as shown at the bottom of the next page.

By equation (35), the value of the state parameter in the prediction horizon can be calculated by the value of the current state parameter $\xi(k\,|k)$ and the control increment $\Delta\mu(k)$. This is also the realization of the "prediction" function in the model predictive control algorithm.

D. ROLLING SOLUTION

In fact, the control increment of the system is unknown. The control sequence in the control horizon can be obtained by solving the appropriate optimization objective. In order to ensure that the vehicle can track the desired trajectory smoothly and accurately while minimizing the changes of input parameters, the objective function includes the deviation of system output and the increment of control as follows:

$$J(k) = \sum_{i=1}^{N_p}\|\eta(k+i\,|k) - \eta_r(k+i\,|k)\|_Q^2 + \sum_{i=1}^{N_c-1}\|\Delta\mu(k+i\,|k)\|_R^2 + \lambda\cdot\sigma^2 \tag{36}$$

Herein: Q and R refer to weight matrix; λ denotes weight coefficient; σ denotes relaxation factor. In equation (36), the first item reflects the tracking ability of the system to the reference trajectory, and the second item reflects the requirement for the smooth change of the control parameters. The second-order difference $\Delta\mu$ of the control variable is used to replace the control variable μ, and the relaxation factor σ is added, which can not only limit the control variable directly, but also prevent the situation that there is no feasible solution in the execution process. At the same time, the system control parameters and control increment need to meet the following constraints in actual control:

$$\begin{cases} \mu_{\min}(k+i) \le \mu(k+i) \le \mu_{\max}(k+i), \\ \quad i = 0, 1, \cdots, N_c - 1 \\ \Delta\mu_{\min}(k+i) \le \Delta\mu(k+i) \le \Delta\mu_{\max}(k+i), \\ \quad i = 0, 1, \cdots, N_c - 1 \end{cases} \tag{37}$$

The objective function (36) can be transformed into a quadratic programming problem in the following form:

$$J(k) = \frac{1}{2}\chi^T(k)H(k)\chi(k) + f^T(k)\chi(k) \tag{38}$$

where

$$\chi = \begin{bmatrix} \Delta U \\ \sigma \end{bmatrix}, \quad H = \begin{bmatrix} \Theta^T Q\Theta + R & 0 \\ 0 & \lambda \end{bmatrix},$$
$$f = \begin{bmatrix} 2E^T Q\Theta & 0 \end{bmatrix}, \quad E = \Omega_k\varepsilon(k\,|k) - Y_r.$$

E. FEEDBACK MECHANISM

By solving equation (36) in each control cycle, a series of control input increments in the control horizon can be obtained: $\Delta U_k = [\Delta\mu_k, \Delta\mu_{k+1}, \cdots, \Delta\mu_{k+N_c-1}]^T$. According to the basic principle of model predictive control, the first element of the control sequence is acted on the controlled system. The sum of the first-order difference $\tilde{\mu}(k)$ obtained by equation (33) and the control parameters in reference trajectory $\mu_r(k)$ acts on the controlled vehicle as the final output, as the follows equation:

$$\mu(k) = \mu_r(k) + \tilde{\mu}(k) \tag{39}$$

After performing this step, the system enters into the next control cycle and repeats the above process, which realizes the tracking control of the reference trajectory.

V. SIMULATION AND VERIFICATION

A. SIMULATION RESULTS OF TRAJECTORY PLANNING BASED ON GPM

The trajectory solving program based on GPM was written on the Matlab platform and embedded in the S-funtion module. The planning results are saved as data files in MAT format so that they can be called by the tracking layer at any time. A certain SUV is taken as the simulation object. The vehicle parameters and physical constraints are shown in Table 1.

Taking the parallel parking condition as the research scene, three parking spaces with different lengths are designed in order to verify the robustness and universal applicability of the GPM. The length of parking space L_p is 4.85m, 6.17m and 7.50m, respectively, i.e. 1.1, 1.4 and 1.7 times of the vehicle length. The center point of the rear-axle is regarded as the observation point of the parking process and the initial state is set to $x_0 = 10$; $y_0 = 1.75$; $\theta_0 = 0$, which is consistent in three cases. The width of the parking space W_p is 2.5m, and the road width W_r is 3.5m according to relevant legal standards. The boundary conditions of state parameters and control parameters are described in equation (4), and the corresponding vehicle body position boundary conditions are defined in equation (5). The planning results are shown in Fig. 5.

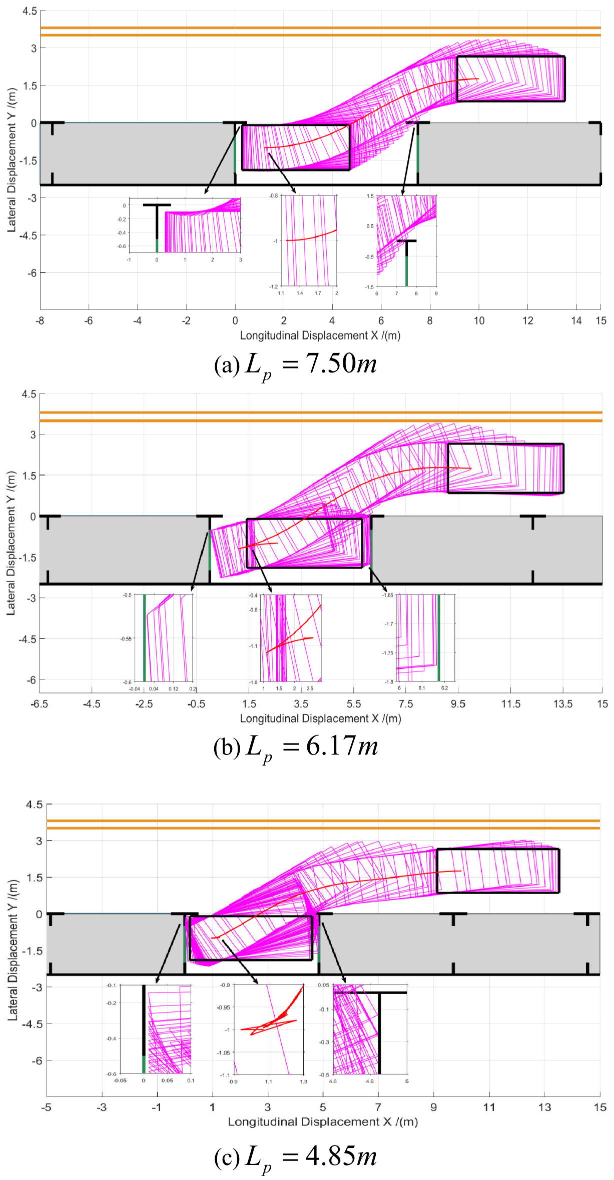
TABLE 1. Vehicle parameters and physical constraints.

Parameter	Parameter description	value
L	Wheelbase	2.620 m
f	Front suspension	0.905 m
r	Rear suspension	0.885 m
d	Width	1.05
α	Safety factor	1.800 m
$v_{\max}$	Maximum driving velocity	3.0 m/s
$a_{\max}$	Maximum acceleration	1 m/s^2
$j_{\max}$	Maximum change rate of acceleration	0.3 m/s^3
$\varphi_{\max}$	Maximum steering angle of the front wheel	0.56 rad
$\omega_{\max}$	Maximum angular velocity of steering angle	0.56 rad/s

As can be seen from Fig. 5 and the partial enlarged figure, in the case of three parking spaces, there is no collision between the vehicle and the road boundary, or between the vehicle and the obstacles in front of and behind the parking space during the parking process. When the parking is completed, the vehicle stays in the parking space and the vehicle body keeps horizontal, which meets the obstacle avoidance constraints and endpoint constraints of the parking system. The results show that GPM can solve the optimal control problem of autonomous parking trajectory planning and obtain the parking trajectory satisfying the constraints. When the length of the parking space is 1.7 times that of the vehicle, the vehicle can complete the parking process with one

$$Y(k) = \begin{bmatrix} \eta(k+1\,|k) \\ \eta(k+2\,|k) \\ \vdots \\ \eta(k+N_c\,|k) \\ \vdots \\ \eta(k+N_p\,|k) \end{bmatrix} \quad \Omega_k = \begin{bmatrix} \tilde{C}(k)\tilde{A}(k) \\ \tilde{C}(k)\tilde{A}(k)^2 \\ \vdots \\ \tilde{C}(k)\tilde{A}(k)^{N_c} \\ \vdots \\ \tilde{C}(k)\tilde{A}(k)^{N_p} \end{bmatrix} \quad \Delta\mu(k) = \begin{bmatrix} \Delta\mu(k\,|k) \\ \Delta\mu(k+1\,|k) \\ \vdots \\ \Delta\mu(k+N_c\,|k) \end{bmatrix}$$

$$\Theta_k = \begin{bmatrix} \tilde{C}(k)\tilde{B}(k) & 0 & 0 & 0 \\ \tilde{C}(k)\tilde{A}(k)\tilde{B}(k) & \tilde{C}(k)\tilde{B}(k) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}(k)\tilde{A}(k)^{N_c-1}\tilde{B}(k) & \tilde{C}(k)\tilde{A}(k)^{N_c-2}\tilde{B}(k) & \cdots & \tilde{C}(k)\tilde{B}(k) \\ \tilde{C}(k)\tilde{A}(k)^{N_c}\tilde{B}(k) & \tilde{C}(k)\tilde{A}(k)^{N_c-1}\tilde{B}(k) & \cdots & \tilde{C}(k)\tilde{A}(k)\tilde{B}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{C}(k)\tilde{A}(k)^{N_p-1}\tilde{B}(k) & \tilde{C}(k)\tilde{A}(k)^{N_p-2}\tilde{B}(k) & \cdots & \tilde{C}(k)\tilde{A}(k)^{N_p-N_C-1}\tilde{B}(k) \end{bmatrix}$$

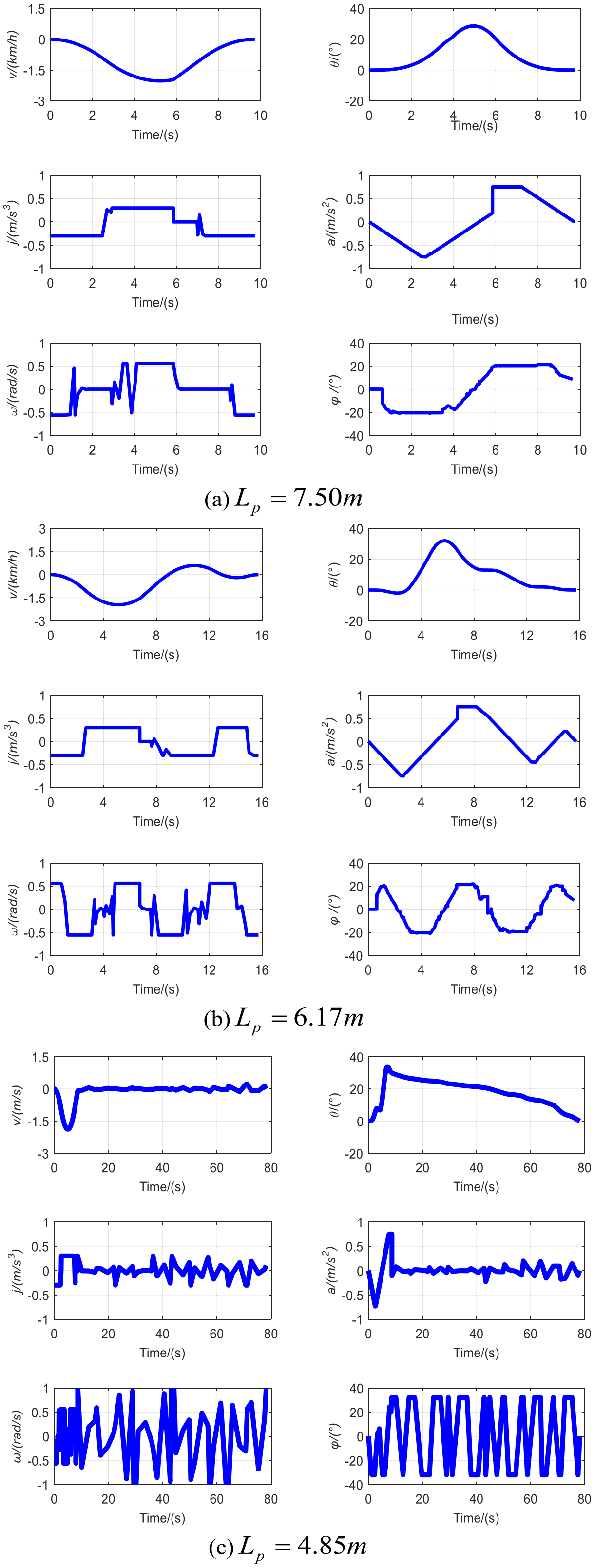


(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 5. Planning results of longitudinal and lateral displacement for autonomous parallel parking in three parking spaces.



(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 6. Planning results of state parameters and control parameters in three parking spaces.

reverse, as the curve of the rear-axle center shown in Fig.5(a), and the parking trajectory is continuous and smooth, which is consistent with the negative sign of velocity v in Fig.6(a). When the length of the parking space is reduced to 6.17m, the vehicle first enters the parking space through a reverse, and then completes the parking process through a forward and backward fine-tuning respectively as the curve of the rear-axle center shown in Fig.5(b), which is also consistent with the positive and negative sign change of the velocity v in Fig.6(b). When the length of the parking space is only 4.85m, it can be seen from the partial enlarged figure in Fig.5(c) that the vehicle moves back and forth repeatedly within the range of horizontal coordinates $[0\ L_p]$ and longitudinal coordinates $[-W_p\ W_r]$, and parking is completed through 20 times of fine-tuning. As shown in Fig.6(c), the control parameters velocity v and front-wheel steering angle present Bang-Bang control mode, the corresponding parameters a, j and ω are the same. This is a typical structure for the solution of the shortest time optimal control problem, which indirectly demonstrates the effectiveness of GPM for solving the optimal control problem. It can be seen in Fig.7 that the planning results of state parameters and control parameters are strictly limited in the constraint range of Table 1, which indicates that GPM strictly implements the constraints in the solution process.

Generally speaking, with the decrease of parking space length, the number of the forward and backward maneuver in the range of longitudinal coordinate [0 L_p] and lateral coordinate [$-W_p$ W_r] increases gradually. Accordingly, in the whole optimization time domain, the number of positive and negative sign changes of velocity v and the parking completion time also increase gradually. The parking time and the number of maneuvers in the three cases are shown in Table 2. There is a positive correlation between the number of vehicle maneuvers and the parking completion time. These results are consistent with the actual driving experience.

TABLE 2. Parking time and number of maneuvers in different parking spaces.

Length of parking space (m)	Parking time (s)	Number of maneuvers
7.497	9.724	0
6.174	15.733	2
4.851	78.154	19

B. SIMULATION RESULTS OF TRAJECTORY TRACKING BASED ON MPC

The co-simulation of vehicle dynamics software CarSim and MATLAB/Simulink is adopted to verify the effectiveness of MPC based trajectory tracking controller in this section. As shown in Fig.7, the model includes a coordinate conversion module, gear control module, brake control module and the embedded CarSim module, which is used to describe the vehicle dynamics performance. Parking scenes and a tracking vehicle are built in the CarSim software. The values of vehicle parameters are shown in Table 1. MPC algorithm program is also written in the S-funtion module. In the simulation process, the S-funtion module reads the planning results in the MAT file format from the workspace in real-time as the reference trajectory. The prediction horizon N_p and control horizon N_c is set to 30 and 2 respectively. The relaxation factor σ is set to 10, and the sampling time T is set to 0.05s. The constraint conditions are consistent with the trajectory planning process. In order to show the superiority of MPC algorithm in trajectory tracking control for parallel parking, PID control method is compared with MPC algorithm. It is worth mentioning that the reason why it is compared with PID control method is that PID control method is one of the most representative control methods in classical control theory.

Fig. 8 shows the tracking results of longitudinal and lateral positions of vehicles under three parking spaces. The pink curve is the reference trajectory planned by GPM. The broken blue curve is the mass center trajectory output by CarSim based on MPC. The blue dot-and-dash curve is the corresponding rear-axle center trajectory. The orange dot-and-dash curve is the rear-axle center trajectory output by CarSim based on PID. When $L_p = 7.50m$, both of the two control methods can make the vehicle quickly and stably reach the target parking space from the initial position. However, the consistency between the tracking trajectory and the reference trajectory of MPC algorithm is better than that of PID method. Compared with MPC algorithm, the tracking

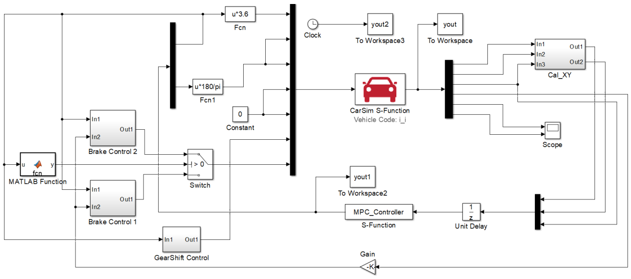
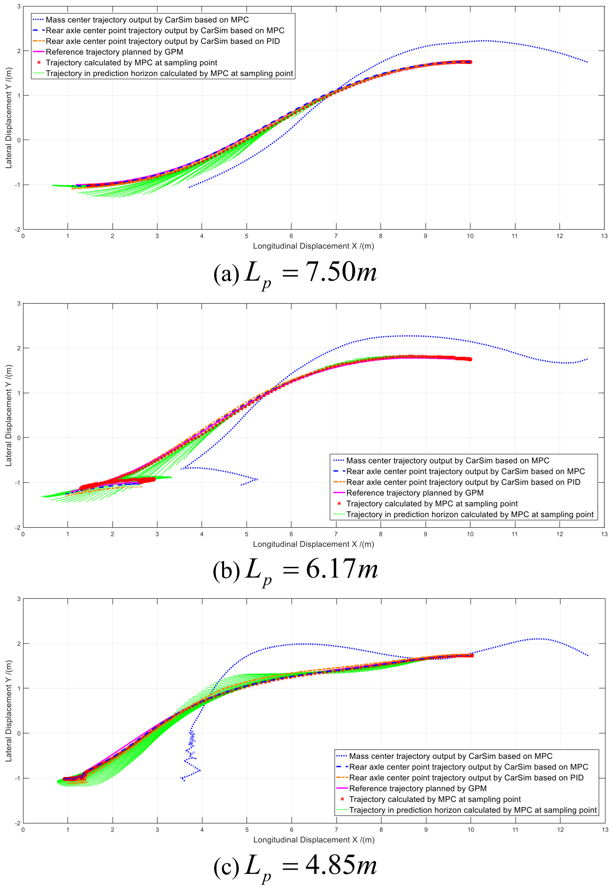


FIGURE 7. Co-simulation platform of autonomous parking based on CarSim and Simulink.



(a) $L_p = 7.50m$

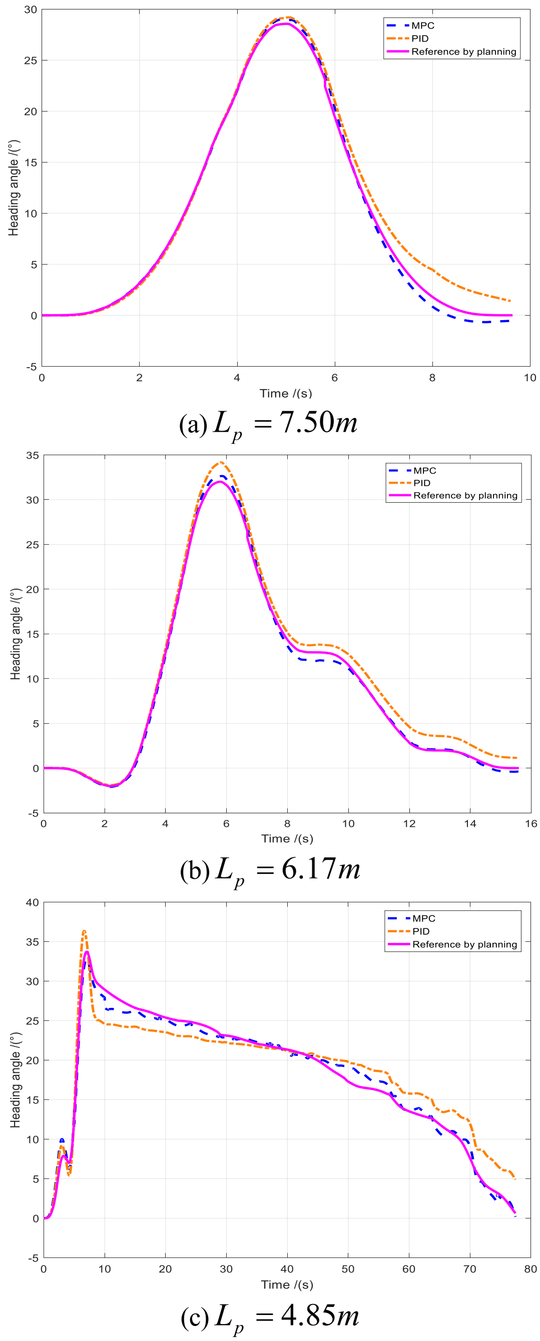
(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 8. Longitudinal and lateral trajectory tracking results based on MPC.

error of Y by PID method is obviously larger at the end of the trajectory. As can be seen from Fig. 8(b) and Fig. 8(c), with the gradual reduction of parking space length, the tracking error of Y based on PID method is getting larger and larger, which eventually leads to the fact that when $L_p = 4.85m$, the vehicle does not stop completely in the parking space when parking is completed. There is no doubt that errors always exist, even in MPC algorithm. The reason for such errors is that the vehicle model adopted in the planning layer is the kinematics model. In contrast, the model described by CarSim in the tracking layer is the vehicle dynamics model. The mass center trajectory curve reflects that when the parking space is medium or small, the vehicles move back and forth to achieve the purpose of parking safely. With the gradual reduction of the length of the parking space, the number of maneuvers is increasing, which is also consistent with the planning results. The red * curve is the longitudinal and lateral position trajectory calculated at each sampling point according to the kinematics model and the control signal output by the MPC. In three cases, the curve is highly consistent with the reference trajectory and the trajectory output by CarSim. But errors still exist, which include not only the errors caused by the differences between the kinematic model and the dynamic model, but also the differences between the linear error prediction model and the nonlinear kinematic model in the MPC algorithm. The green curve is the trajectory of the longitudinal and lateral position in the prediction horizon calculated by the predictive model of MPC controller at each sampling point, which also reflects an essential feature of MPC. The trajectory in the prediction horizon can be obtained at any sampling point. Although the deviation between the predicted trajectory and the reference trajectory is larger and larger, it does not affect the tracking effect of MPC, because only the first value in the control signal sequence is taken to act on the vehicle at each sampling time, and the feedback correction is carried out according to the output of the vehicle, so as to reduce the tracking error.

Fig.9 shows the tracking results of vehicle heading angle. The tracking curve of the heading angle based on MPC is in better agreement with the reference curve in three cases, and the car body can be kept level when parking is completed. That is, the heading angle is finally kept at zero. However, the maximum deviation of the heading angle tracking based on PID is larger than that of MPC algorithm, and the heading angles are not kept at zero in three cases, indicating that the car body is not kept level. With the decrease of parking space length, the tracking deviation of heading angle based on PID control method is larger and larger, which shows that the tracking effect of PID is poor in the low speed and large steering condition, while the MPC algorithm has a weak sensitivity to the length of the parking space. In the place where the heading angle curve has a big turning point, it is easy to have a large tracking error, which is mainly due to the fact that it takes some time for the steering wheel to turn in the actual control process. This process may cause an inevitable delay in the heading angle tracking, but the deviation of the heading angle has little influence on the longitudinal and lateral displacement tracking.



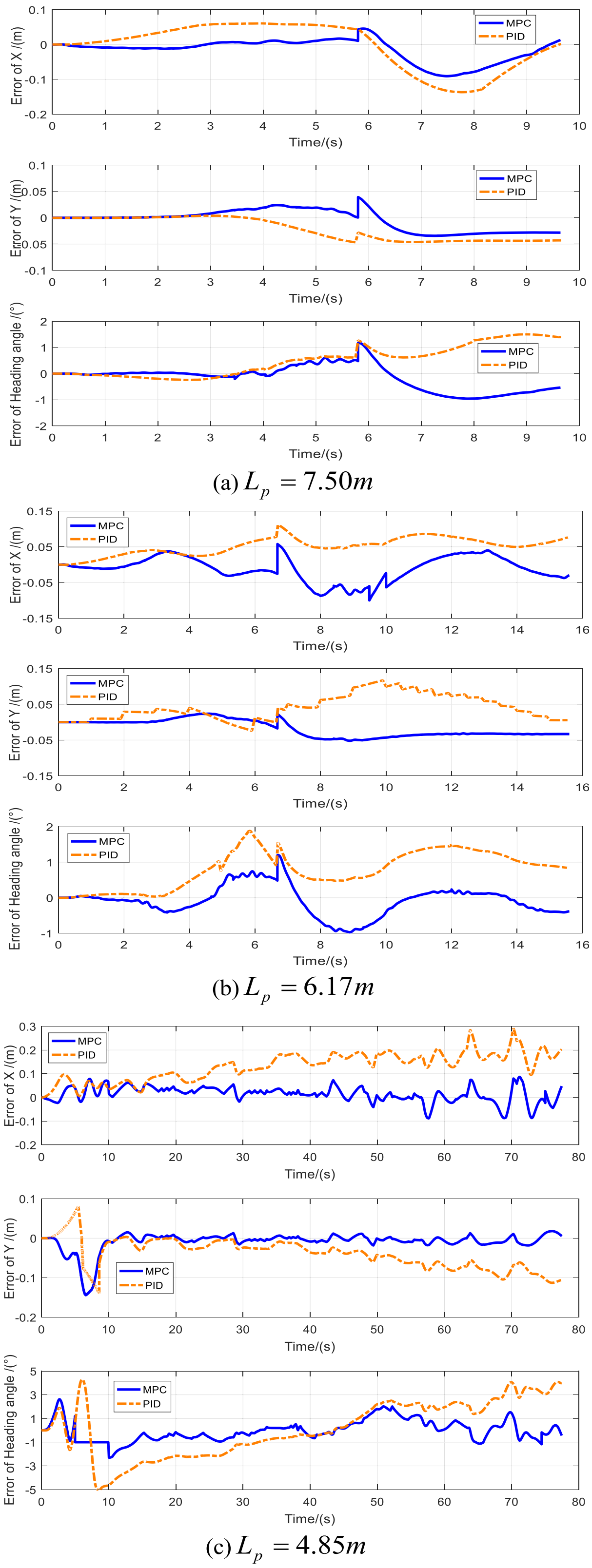
(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 9. Heading angle tracking results in three cases.

Fig.10 shows the time-variation curve of the tracking error of the main state parameters in three cases. The maximum tracking errors of longitudinal and lateral displacement based on MPC are not more than 0.1m and the corresponding maximum heading angle errors are not more than 1.5° in medium and large parking spaces. In these two parking spaces, the longitudinal and lateral errors of MPC algorithm are less than those of PID algorithm, but this advantage is



(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

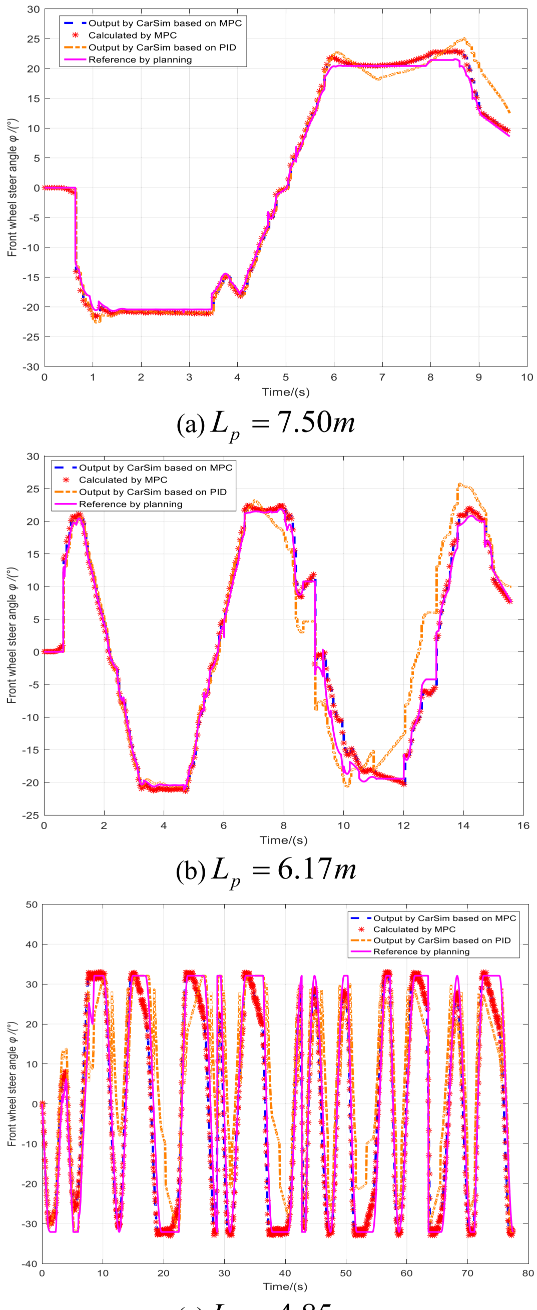
(c) $L_p = 4.85m$

FIGURE 10. Relationship between tracking error of state parameters and time.

not obvious. While in a small parking space, the tracking errors of the three state parameters by two control methods all increase. However, the deviation of PID method increases more sharply, especially the heading angle. The maximum errors of longitudinal and lateral displacement based on MPC are not more than 0.15m, and the corresponding heading angle errors are not more than 2°. The longitudinal error of PID method is 0.3m, and the heading angle deviation is close to 5°. The increase of error is mainly due to that the vehicle need to move back and forth in a narrow space when the parking space is small. This kind of maneuver has obvious characteristics, such as the large steering angle and low velocity, which requires high stability and accuracy of the steering system and braking system. Consistent with the previous analysis, MPC algorithm is better than PID algorithm in the error performance of large, medium and small parking spaces.

Fig.11 shows the curves of the relationship between the steering angle of the front wheel and time in three cases. The steering angle does not exceed the maximum allowable value of the constraints in three cases based on MPC and PID. In three cases, MPC algorithm keeps good tracking performance for steering angle. In contrast, PID algorithm also has good tracking performance in the early stage, but the tracking error gradually increases with the passage of time, especially in small parking spaces. As can be seen from Fig.11(c), the tracking ability of PID method to a large angle is limited. In MPC, the output by CarSim and calculation value keep the same. This indicates that the control signal output by MPC is strictly implemented in CarSim and the different models still cause the error with the reference trajectory. In both of the two control methods, the maximum angle tracking error occurs at the upper and lower boundaries of the front wheel steering angle during the fore-and-aft maneuver, where the execution delay of the steering system is easily caused by a too large angle. However, due to the feedback mechanism of MPC, the error can be eliminated quickly, and the trajectory gradually approaches the reference trajectory. On the contrary, PID method has limited correction ability due to the uncertainty of its parameters.

Fig.12 shows the curves of the relationship between the velocity and time in three cases. In large parking spaces, the velocity tracking results of MPC and PID methods are basically the same, which is mainly due to the fast velocity and there is no back and forth maneuver, so the requirements of the controller are not high. In the medium parking space, the situation of back and forth maneuver exists, MPC still shows good tracking performance when the velocity direction changes, while PID method has a large tracking error at the velocity switching position. This phenomenon is very obvious in small parking spaces. In small parking spaces, there are many back and forth maneuvers, the velocity calculated by MPC and output by CarSim are basically consistent with the reference trajectory. MPC shows excellent low velocity tracking ability. However, some error still exists in the small space. The reason is similar to the front wheel angle. The stability and accuracy of the braking system are required to be higher under low-velocity conditions. However, this small error has no decisive influence on the success of parking.

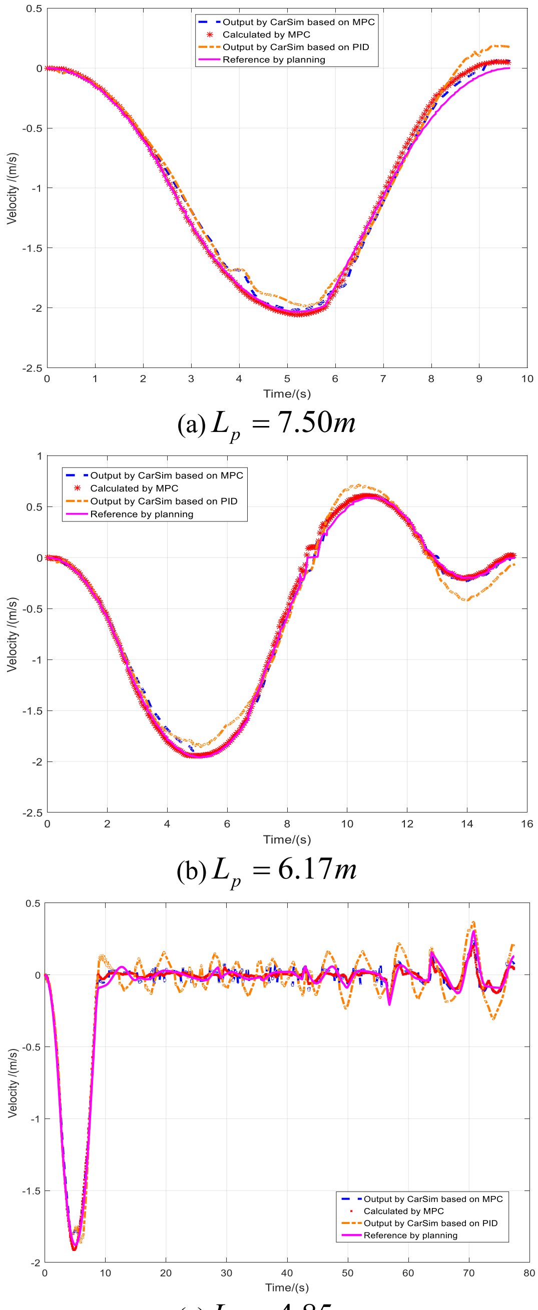


(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 11. Relationship between steering angle of front wheel and time.



(a) $L_p = 7.50m$

(b) $L_p = 6.17m$

(c) $L_p = 4.85m$

FIGURE 12. Relationship between velocity and time.

In general, the MPC algorithm shows good robustness and tracking performance in terms of state parameters.

VI. REAL VEHICLE TEST VALIDATION

A medium-sized car was selected to build an autonomous parking test platform. The ultrasonic radars were regarded as the environmental sensing sensor, and the idle parking space could be detected by the lateral sensor. The main task of the host computer was to run the hierarchical control algorithm for trajectory planning and tracking, including GPM algorithm and MPC algorithm. The control system of the execution parts of the test vehicle was reformed by wire control, including steering system, braking system, power system and gear system, which respectively realized the control functions of active steering, velocity following and direction control of movement, so that the lateral and longitudinal movement of the test vehicle could be directly controlled by signals on CAN bus. The steering wheel angle signal is provided by the angle sensor of the electric power steering(EPS) system, and the wheel speed pulse signal, longitudinal acceleration

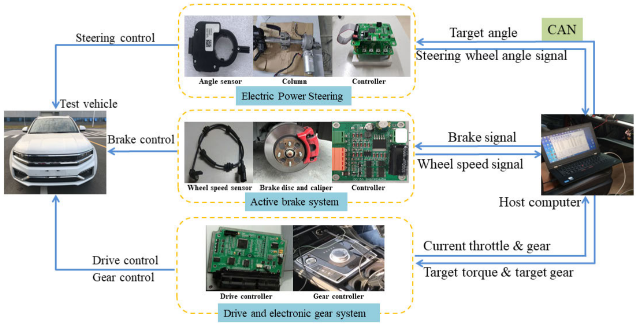


FIGURE 13. The architecture of the real vehicle test system.

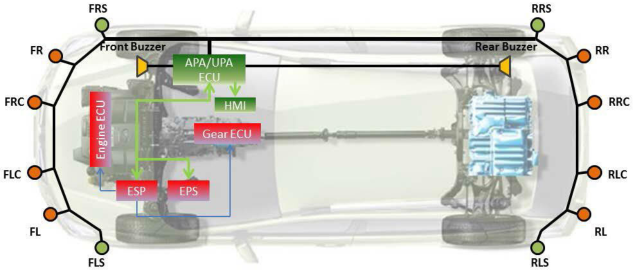


FIGURE 14. Installation diagram of ultrasonic radar.

signal and yaw rate signal are provided by the wheel speed sensor and the acceleration sensor of the braking system. The host computer processes the above signals and parking space information to obtain the target parking trajectory information and control variables, including target steering angle, target velocity and target gear, and sends the target control signal to each executive control system. The signal communication between each system is carried out through the CAN bus. The system architecture is shown in Fig.13.

The test vehicle is equipped with 12 ultrasonic radars, of which 4 are short-range mode radars at the front and rear, and 2 are long-range mode radars at the left and right sides. Elmos e524.3 chip is used to control the sensors to send and receive signals of ultrasonic radars. As shown in Fig. 14, the long-range mode radars on both sides in front of the vehicle are mainly used to detect the parking space. Due to the considerable depth of the parking space, the long-range mode radars are required to be able to detect long enough. Therefore, the detection distance of the radars assembled on the vehicle in this experiment is 0.3-6m.

For the detection of idle parking space, the edge of the parking space is determined by judging the mutation of the return distance data of the radar. As shown in Fig.15, when the vehicle passes through the idle parking space, the radar detection distance data shows a falling edge and a rising edge near point A and point B. Based on this, these two points can be considered as the starting point and the end point of the free area respectively. The space size can be calculated by combining the velocity and driving time of the vehicle passing through this area, and then whether the idle area is an effective parking space can be determined according to the requirements of the vehicle for the minimum parking space. The ultrasonic radar is susceptible to the interference of the external environment and its own multiple echoes during the detection process, which leads to the existence of error measurement values. As a result, it is necessary to smooth and filter the obtained detection data. In addition, considering the uncertainty of obstacle contour, invalid inflection points will appear in the process of parking space detection, which can be eliminated by the amplitude of the data jump.

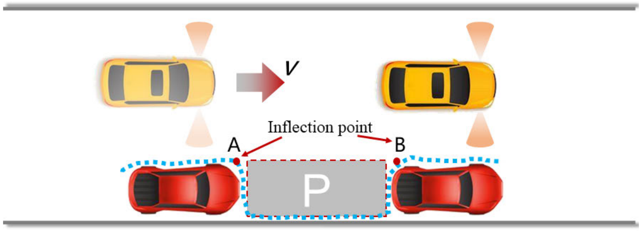


FIGURE 15. Parking space detection diagram.

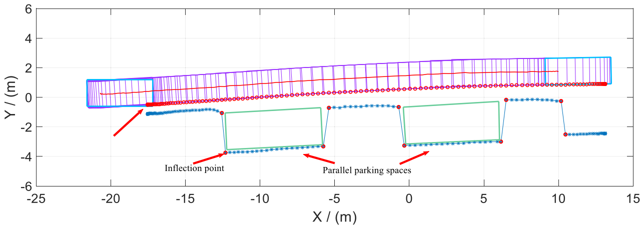


FIGURE 16. Real vehicle detection results of parking spaces.

The length and width of the selected parallel parking spaces were 6.2 m and 2.5 m respectively, which represented the typical medium-size parking space. The vehicle traveled in a straight line from stationary. After finding multiple effective parking spaces, the vehicle stopped and maintained idle conditions. The driving duration was 23.831s, the maximum driving velocity was 2.564m/s, and the maximum deviation angle of body heading angle was 3.893°. Fig.16 shows the result of real vehicle parking detection. The ultrasonic radar of the parking system detected multiple inflection points in the detection process. A total of three free areas were detected according to the positions of the radar inflection points. The first two idle areas are determined as effective parking spaces. The first detected parking space is 6.390 m in length and 2.493 m in width. The space length of the second detection parking space was 6.388 m, and the width was 2.508 m. Although there was a certain space in the third idle area, the space size did not meet the parking space standard, and it was judged as a non-parking space.

The length of the above detection parking space is more than 6.2 m, mainly considering the blind area of ultrasonic radar detection. In order to avoid vehicle collision during real vehicle test, the effective motion range of vehicle in parking space should be less than the actual size of the parking space. Consequently, the effective motion range is

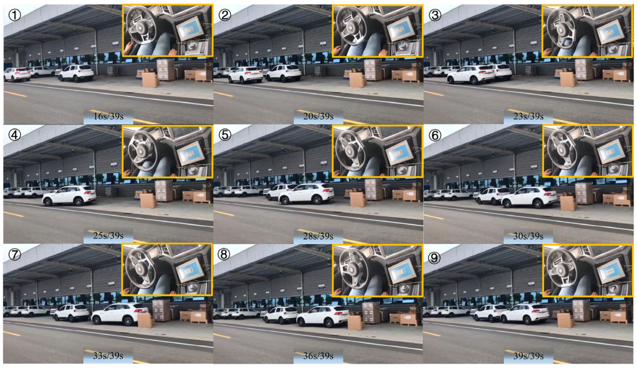


FIGURE 17. Real vehicle test of autonomous parking under parallel working conditions.

TABLE 3. Real vehicle test results of autonomous parallel parking.

Test No.	Parking space length /m		Longitudinal position of contour vertices /m		Heading angle /°	Whether collision or not
	Target	Measured	Vertex A	Vertex D		
1	6.4	6.390	-0.657	-0. 679	0.48	NIL
	6.4	6.388	-0.662	-0.688	0.55	NIL
2	6.4	6.380	-0.679	-0.672	-0.10	NIL
	6.4	6.389	-0.694	-0.723	0.63	NIL
3	6.4	6.383	-0.688	-0.684	-0.05	NIL
	6.4	6.379	-0.681	-0.701	0.42	NIL

used as the input parameter in the trajectory planning layer. Because the detection blind area of the ultrasonic radar in short range mode is $0 \sim 0.2\ m$. Hence, the effective length of the parking space is $0.2\ m$ shorter than the measured length. In the opposite sense, it means that the length of the actual parking space should be increased by $0.2\ m$ compared with the simulation environment, and other parameter settings are the same. The vehicle was tested three times from left to right, and two valid parking spaces can be detected by the radar in each test. In this way, the reliability of test results can be guaranteed. The vehicle drove to the initial parking position and kept the vehicle static, and then the autonomous parking system takes over the vehicle. The reference trajectory was planned according to the actual vehicle pose and the size of the detected parking space. The trajectory information further guided the vehicle to park. The real vehicle test process is shown in Fig.17.

At present, there is no unified evaluation standard for the performance of the parking system. Some parking parameters are recorded to determine the completion of the parking, such as the measured length of the parking space, whether the vehicle collides with the obstacle in the parking process, the longitudinal positions of the contour vertices A and D, and the body attitude angle after the completion of the parking. The test results are shown in Table 3.

The experimental results show that although the measured length of the parking space was smaller than the actual length due to the physical characteristics of the beam angle of the ultrasonic sensor, taking the actual measured length as the input parameter of the trajectory planning algorithm also ensured the safety of the parking process. During the three tests, the vehicle did not collide with any obstacle and was safely parked into the target parking space, i.e., the four contour vertices were all in the parking space, and the body heading angle was within the range of $\pm 1^\circ$. The final parking effect was acceptable.

VII. CONCLUSION

The innovation of this paper is to propose a hierarchical control method for autonomous parallel parking trajectory planning and trajectory tracking. The GPM algorithm is adopted in the planning layer to realize the rapid planning of autonomous parallel parking trajectory. The collision avoidance constraint is considered in the algorithm to effectively improve the safety of parking, which is characterized by fast convergence, good robustness. The planning result is trajectory, which is easier to track for the actuator. The tracking layer uses MPC algorithm to achieve accurate and stable trajectory tracking. The algorithm considers various constraints of spatial state variables, and takes the minimum error of vehicle position and heading angle as the objective function, which improves the trajectory tracking accuracy and adaptability, so that its tracking performance is better than that of traditional PID. In the practical application process, under the condition of fully obtaining the parking environment information, the double-layer algorithm can simultaneously solve the decision planning technology and control execution technology in the key technologies of autonomous parking.

1) According to the development architecture of autonomous parking system with hierarchical control of trajectory planning and tracking, this paper proposes an autonomous parking trajectory planning and tracking control method based on GPM-MPC. In the planning layer, by establishing the vehicle kinematics model, physical system constraints, boundary constraints and obstacle avoidance constraints, the parking trajectory planning problem is transformed into the optimal control problem, and the shortest parking completion time is taken as the optimal objective function, and GPM is applied to solve the optimal control problem. Three parallel autonomous parking conditions with different lengths of parking spaces are selected for trajectory planning simulation. The results show that GPM can converge to obtain the parking trajectory without collision in three cases. The algorithm can realize parking trajectory planning in a narrow space, and the obtained trajectory is smooth, which is conducive to trajectory tracking control and can improve the success rate of parking.

2) In the trajectory tracking control layer, a parking trajectory tracking controller based on the MPC algorithm is designed. Velocity and steering angle of the front wheel sequence is obtained as the control signals for trajectory tracking by MPC. The effectiveness of the tracking control method based on MPC was verified by the co-simulation of CarSim and Simulink. Compared with the simulation results based on PID control method, the simulation results show that the tracking errors of longitudinal and lateral displacement are less than 0.15m and the errors of heading angle are less than 2° in three cases. MPC based trajectory tracking control has higher accuracy and better robustness.

3) Real vehicle test is implemented to further verify the feasibility and effectiveness of the proposed hierarchical control algorithm for trajectory planning and tracking based on GPM and MPC. Through three tests, a total of six effective parking spaces were detected by vehicle-borne ultrasonic radar, and the vehicle can park into the parking spaces safely and quickly without collision. The errors of vehicle contour vertex and heading angle are small. The test results show that the proposed method is suitable for different parking scenarios. However, this method still has some errors and requires a large amount of computation. Reducing the error and improving the computational efficiency would be further research.

REFERENCES

[1] Z. Qin, X. Chen, M. Hu, L. Chen, and J. Fan, "A novel path planning methodology for automated valet parking based on directional graph search and geometry curve," *Robot. Auton. Syst.*, vol. 132, pp. 1–12, Oct. 2020.

[2] W. Yao, N. Qi, C. Yue, and N. Wan, "Curvature-bounded lengthening and shortening for restricted vehicle path planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 15–28, Jan. 2020.

[3] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 396–410, Feb. 2015.

[4] R. F. Zhou, X. F. Liu, and G. P. Cai, "A new geometry-based secondary path planning for automatic parking," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, pp. 1–17, May 2020.

[5] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961–1976, Aug. 2015.

[6] X. Jin, Z. Yan, G. Yin, S. Li, and C. Wei, "An adaptive motion planning technique for on-road autonomous driving," *IEEE Access*, vol. 9, pp. 2655–2664, 2021.

[7] K. Mi, J. Zheng, Y. Wang, and J. Hu, "A multi-heuristic A* algorithm based on stagnation detection for path planning of manipulators in cluttered environments," *IEEE Access*, vol. 7, pp. 135870–135881, Sep. 2019.

[8] P. Shen, X. Zhang, and Y. Fang, "Tree-search-based any-time time-optimal path-constrained trajectory planning with inadmissible island constraints," *IEEE Access*, vol. 7, pp. 1040–1051, Dec. 2019.

[9] W. Zhu, W.-C. Yeh, N. N. Xiong, and B. Sun, "A new node-based concept for solving the minimal path problem in general networks," *IEEE Access*, vol. 7, pp. 173310–173319, Nov. 2019.

[10] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path planning of industrial robot based on improved RRT algorithm in complex environments," *IEEE Access*, vol. 6, pp. 53296–53306, Sep. 2018.

[11] A. T. Khan, S. Li, S. Kadry, and Y. Nam, "Control framework for trajectory planning of soft manipulator using optimized RRT algorithm," *IEEE Access*, vol. 8, pp. 171730–171743, Jan. 2020.

[12] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 7, pp. 993–1004, Jul. 2016.

[13] H. Li, Y. Luo, and J. Wu, "Collision-free path planning for intelligent vehicles based on Bézier curve," *IEEE Access*, vol. 7, pp. 123334–123340, Aug. 2019.

[14] Y. Chen, Y. Cai, J. Zheng, and D. Thalmann, "Accurate and efficient approximation of clothoids using Bézier curves for path planning," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1242–1247, Oct. 2017.

[15] Y. L. Kuo, C. C. Lin, and Z. T. Lin, "Dual-optimization trajectory planning based on parametric curves for a robot manipulator," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 3, pp. 1–14, May 2020.

[16] J. Song, W. Zhang, X. Wu, H. Cao, Q. Gao, and S. Luo, "Laser-based SLAM automatic parallel parking path planning and tracking for passenger vehicle," *IET Intell. Transp. Syst.*, vol. 13, no. 10, pp. 1557–1568, Oct. 2019.

[17] Y. Liu, C. Guo, and Y. Weng, "Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance," *IEEE Access*, vol. 7, pp. 146301–146308, Oct. 2019.

[18] J. Zhang, H. Chen, S. Song, and F. Hu, "Reinforcement learning-based motion planning for automatic parking system," *IEEE Access*, vol. 8, pp. 154485–154501, Aug. 2020.

[19] W. Liu, Z. Li, L. Li, and F.-Y. Wang, "Parking like a human: A direct trajectory planning solution," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3388–3397, Dec. 2017.

[20] B. Woosley and P. Dasgupta, "Integrated real-time task and motion planning for multiple robots under path and communication uncertainties," *Robotica*, vol. 36, no. 3, pp. 353–373, Mar. 2018.

[21] M. Mehndiratta, E. Kayacan, M. Reyhanoglu, and E. Kayacan, "Robust tracking control of aerial robots via a simple learning strategy-based feedback linearization," *IEEE Access*, vol. 8, pp. 1653–1669, Dec. 2020.

[22] N. N. Samani, M. Danesh, and J. Ghaisari, "Parallel parking of a car-like mobile robot based on the P-domain path tracking controllers," *IET Control Theory Appl.*, vol. 10, no. 5, pp. 564–572, Mar. 2016.

[23] D. Xu, Y. Shi, and Z. Ji, "Model-free adaptive discrete-time integral sliding-mode-constrained-control for autonomous 4WMV parking systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 834–843, Jan. 2018.

[24] Z. Fan and H. Chen, "Study on path following control method for automatic parking system based on LQR," *SAE Int. J. Passenger Cars-Electron. Electr. Syst.*, vol. 10, no. 1, pp. 41–49, Sep. 2016.

[25] Z. Liu, W. Zhang, and Z. Lv, "Design and experiment of path tracking of agricultural vehicle," *Int. Agricult. Eng. J.*, vol. 27, pp. 44–52, Dec. 2018.

[26] J. Backman, T. Oksanen, and A. Visala, "Nonlinear model predictive trajectory control in tractor-trailer system for parallel guidance in agricultural field operations," *IFAC Proc. Volumes*, vol. 43, no. 26, pp. 133–138, 2010.

[27] X. Song, Y. Shao, and Z. Qu, "A vehicle trajectory tracking method with a time-varying model based on the model predictive control," *IEEE Access*, vol. 8, pp. 16573–16583, Dec. 2020.

[28] Y. Wang, S. Shi, S. Gao, Y. Xu, and P. Wang, "Active steering and driving/braking coupled control based on flatness theory and a novel reference calculation method," *IEEE Access*, vol. 7, pp. 180661–180670, Dec. 2019.

[29] X. Jin, J. Yang, Y. Li, B. Zhu, J. Wang, and G. Yin, "Online estimation of inertial parameter for lightweight electric vehicle using dual unscented Kalman filter approach," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 1–10, 2020.

[30] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proc. ASME Dyn. Syst. Control Conf.*, Cambridge, MA, USA, Jan. 2010, pp. 265–272.

DUOYANG QIU was born in Shouxian, Anhui, China, in 1986. He received the M.S. and Ph.D. degrees in automotive engineering from the School of Automotive and Traffic Engineering, Hefei University of Technology, Hefei, China, in 2012 and 2019, respectively. Since 2019, he has been working with Hefei University, engaged in scientific research. His research interests include control technology of hybrid electric vehicle, motion planning, and tracking methods for automated vehicles.

DUOLI QIU was born in Huainan, Anhui, China, in 1981. He received the M.S. degree in computer software and theory from Huaibei Normal University, Huaibei, China, in 2012, where he is currently pursuing the Ph.D. degree. He is also an Associate Professor with the School of Economic and Management, Huaibei Normal University. His research interests include artificial intelligence, machine learning, software theory, and information processing.

BING WU was born in Tongcheng, Anhui, China, in 1986. He received the M.S. degree in automotive engineering from the School of Automotive and Traffic Engineering, Hefei University of Technology, China, in 2012, where he is currently pursuing the Ph.D. degree. His research interests include vehicle automation, motion planning, real-time trajectory generation, and advanced driver assistance systems.

MAN GU was born in Hefei, Anhui, China, in 1970. She received the Ph.D. degree in materials science from the School of Materials Science and Engineering, Hefei University of Technology, Hefei. She worked as a Professor with Hefei University, engaged in scientific research and teaching. Her research interests include intelligent manufacturing equipment and advanced manufacturing technology.

MAOFEI ZHU was born in Hefei, Anhui, China, in 1983. He received the Ph.D. degree in automotive engineering from the School of Automotive and Traffic Engineering, Hefei University of Technology, China, in 2011. His research interests include vehicle system dynamics and control, intelligent vehicle control by wire, and advanced driver assistance systems.