

Article

A Trajectory Planning Method for Autonomous Valet Parking via Solving an Optimal Control Problem

Chen Chen , Bing Wu, Liang Xuan, Jian Chen, Tianxiang Wang and Lijun Qian *

College of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China; cclwtg@163.com (C.C.); hfutwubing@126.com (B.W.); xaun865807064@126.com (L.X.); 2017110894@mail.hfut.edu.cn (J.C.); wtx417@163.com (T.W.)

* Correspondence: qianlijun66@163.com

Received: 14 September 2020; Accepted: 9 November 2020; Published: 11 November 2020



Abstract: In the last decade, research studies on parking planning mainly focused on path planning rather than trajectory planning. The results of trajectory planning are more instructive for a practical parking process. Therefore, this paper proposes a trajectory planning method in which the optimal autonomous valet parking (AVP) trajectory is obtained by solving an optimal control problem. Additionally, a vehicle kinematics model is established with the consideration of dynamic obstacle avoidance and terminal constraints. Then the parking trajectory planning problem is modeled as an optimal control problem, while the parking time and driving distance are set as the cost function. The homotopic method is used for the expansion of obstacle boundaries, and the Gauss pseudospectral method (GPM) is utilized to discretize this optimal control problem into a nonlinear programming (NLP) problem. In order to solve this NLP problem, sequential quadratic programming is applied. Considering that the GPM is insensitive to the initial guess, an online calculation method of vertical parking trajectory is proposed. In this approach, the offline vertical parking trajectory, which is calculated and stored in advance, is taken as the initial guess of the online calculation. The selection of an appropriate initial guess is based on the actual starting position of parking. A small parking lot is selected as the verification scenario of the AVP. In the validation of the algorithm, the parking trajectory planning is divided into two phases, which are simulated and analyzed. Simulation results show that the proposed algorithm is efficient in solving a parking trajectory planning problem. The online calculation time of the vertical parking trajectory is less than 2 s, which meets the real-time requirement.

Keywords: autonomous parking; trajectory planning; homotopic method; optimal control

1. Introduction

With the development of an intelligent transportation system (ITS), parking technology has gone through three stages: semi-autonomous parking, autonomous parking, and autonomous valet parking (AVP) [1]. The semi-autonomous parking system and autonomous parking system are used to help drivers park safely. Moreover, a shorter parking time can effectively reduce the negative impact of a parking process on road traffic. In addition, the last-mile transportation of autonomous driving can be achieved with an AVP system since users can send instructions through terminal devices to park or retrieve their cars [2]. In this context, parking technology is an indispensable part of the autonomous driving system.

Among the key technologies of AVP, researchers focus on the platforms, sensor setups, perception, environment model, maps, and motion planning [3]. As the basis for the realization of AVP, environmental perception is mainly realized by a fusion of ultrasonic radars and cameras [4]. On this basis, a parking guidance system (PGS) based on visual recognition is built, and environment

information is shared between the vehicle and the infrastructure system [5]. Furthermore, the tracking control is utilized to move the vehicle along the planned target path by the lateral and longitudinal motion controls [6,7].

Considering that the effects of environment perception and tracking control mainly depend on the accuracy of the sensors, path planning plays an important role in the entire system as a decision-making part. Thus, vehicle path planning is a research hotspot in the intelligent AVP technology. In the early days, curves composed of straight lines and arc joints, like Dubins curve and Reeds–Shepp curve, were used to solve planning problems by setting the shortest path as the goal [8,9]. Additionally, the geometric construction method of a straight line combined with arc joints has been widely used in the path planning of parallel parking [10]. However, the curvature of the path is discontinuous despite that the path can be easily obtained by this method. Therefore, methods such as clothoid curve, B-spline, and polynomial curve are applied to optimize the curvature of the path, but these methods are difficult to apply in path planning with a complicated constraint model [11–13].

In subsequent development, the path planning algorithms of motion planning are mainly divided into the following four categories: graph-search, sample-based, interpolation, and numerical optimization [14]. Considering the simplicity of practical application, graph-search methods are more efficient than traditional path planning strategies [15]. For example, classical graph-search algorithms like the Dijkstra and A* algorithms are both widely used to search the shortest path [16–18]. Moreover, compared with the Dijkstra algorithm, heuristic guidance is combined with the A* algorithm to reduce search work. However, they are not efficient enough to obtain the shortest path in a complicated environment. On the contrary, both the Probabilistic Roadmap Method (PRM) and Rapid-exploration Random Tree (RRT), which are based on random sampling points for planning the best path in a space, can effectively deal with a path planning problem with complicated constraints [19,20]. The above methods are all effective in global planning. However, for local path planning, the artificial potential field (APF) for obtaining the path along the direction of the field drop is more suitable [21].

Compared with the traditional algorithms mentioned above, heuristic algorithms (e.g., machine learning, neural network, fuzzy control, genetic algorithm, particle swarm optimization, and colony algorithm [22–27]) are more widely applied in solving path planning problems. In a complex dynamic environment, bio-inspired planning algorithms are effective for collision-free planning [28,29]. Additionally, in an actual urban environment, a motion planning method based on a layered strategy is more suitable for the application of autonomous driving [30]. Moreover, scholars tend to combine traditional algorithms with heuristic algorithms to obtain better planning results. For example, the model predictive control method is used for real-time path planning [31]. Furthermore, the path planning problem is described as an optimal control problem based on optimal control theory [32–34].

In general, the key to developing AVP technology is the cooperation between trajectory planning and parking management systems [35]. Problems such as complex environment models and low computational efficiency are common in existing trajectory planning methods. To solve these problems, a method that obtains the optimal trajectory of AVP by solving the optimal control problem is proposed. The vehicle kinematics model, obstacle avoidance constraints, and endpoint constraints are established. The shortest overall parking time and travel distance are set as the cost function of the optimal control problem, which are established to describe the parking trajectory planning problem. On this basis, the contribution of this paper has two main components: (i) A homotopic method is utilized to transform the obstacle avoidance constraint in each iteration, and the optimal solution is set as the initial guess of the next new problem. The purpose of this step is to increase the success rate of offline planning with less cost. (ii) An online calculation method of vertical parking trajectory is proposed, where the starting points are decided in the initial area of vertical parking to build a matrix. Additionally, the parking trajectory of each starting point is obtained by offline calculation and stored. To improve the real-time performance of online calculation, in the vertical parking phase, the appropriate trajectory data are selected as the initial guess of online calculation according to the actual starting position of parking. Then, the Gauss pseudospectral method (GPM) is used to obtain the vertical parking trajectory.

A small parking lot is selected as the scenario of AVP, and the parking trajectory planning is separated into two phases. In this context, the trajectories of the two phases are simulated and analyzed. On this basis, the vertical parking trajectory is calculated online to verify the effectiveness of the proposed algorithm.

The remainder of the paper is organized as follows. The AVP scenario and the optimal control problem for parking, including kinematics models and various constraints, are described in Section 2. The optimal control problem is discretized as the nonlinear programming (NLP) problem through the GPM, and the solution strategy of the NLP problem through the homotopic method is described in Section 3. The simulation results and discussion of parking trajectory are in Section 4, and an online planning method of vertical parking is proposed. Finally, the conclusion and future works are discussed in Section 5.

2. Trajectory Optimal Control Problem of AVP

2.1. AVP Scenario

In this paper, a small parking lot is selected as the scenario for verifying the algorithm. There is a U-shaped road in the parking lot, and the parking spaces on both sides of the road are vertical (see Figure 1a). Denoted by positions A, B, and C are the parking entrance, initial vertical parking position, and parking completion position, respectively. Based on the common parking process of a human driver, the parking process is divided into two phases: (i) The search for a vacant parking space, where the driver starts to seek a vacant parking space when entering the parking lot. Then the vehicle is moved to an initial parking area after finding a parking space. (ii) The vertical parking phase, where the vehicle moves backwards from the initial parking area until it is completely in the parking space. To facilitate parking trajectory planning, a 3D perspective of the parking space is set to top view (see Figure 1b). The initial vertical parking area and parking space are represented by the green dotted frame and the red solid frame, respectively, and the paths of the two phases are A–B and B–C.

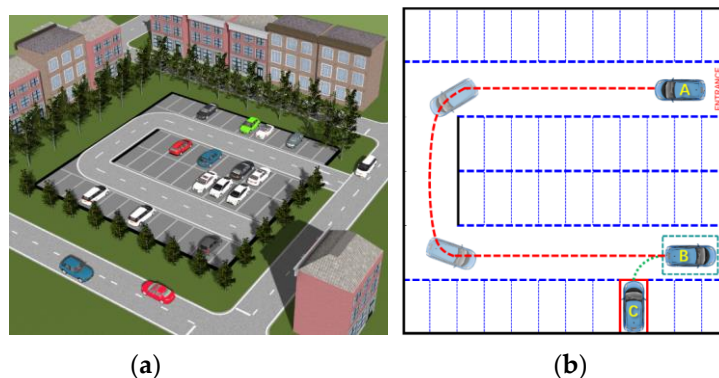


Figure 1. Parking lot schematic map: (a) parking lot in 3D view, (b) parking lot in 2D view.

Assuming that the parking lot is built with an infrastructure system, the vehicle can communicate with the system by Vehicle to Infrastructure (V2I) technology [36]. Moreover, vacant parking spaces can be detected by the system and allocated to vehicles entering the parking lot. In this context, the vehicle can be guided to the initial vertical parking area according to the system information. However, the guidance information provided by the system is not complete. Vehicle sensors, such as ultrasonic sensors, are needed to detect the environment for the accuracy of information around the vacant parking space. Besides, the parking trajectory in the vertical parking phase is obtained according to the initial parking position and the position relative to the vacant parking space. The trajectories of these two phases can be obtained in the following two means. One is that the trajectory is calculated by the server of the infrastructure system and sent to the vehicle, which is controlled to track the trajectory.

Another is that the vehicle obtains part of the guidance information from the facility system, and then combines it with its own perception information for calculation. Also, the vehicle is controlled to track the obtained trajectory.

2.2. Kinematics Model

According to Ackerman steering, all wheels are in a state of rolling, and the vehicle is not affected by lateral force during the parking process. Thus, the slip of the wheel is neglected, and the steering axis of the front wheel intersects the axis of the rear wheel. On this basis, a kinematics model of a vehicle in autonomous parking is built as shown in Figure 2, where $R(x, y)$ denotes the center of the rear axle; v refers to the speed of the center of the rear axle; O stands for the center of the instantaneous turning of the vehicle; f and r denote the front suspension and rear suspension of the vehicle, respectively; d stands for the width of the vehicle; L is the wheelbase of the vehicle; φ stands for the equivalent front wheel swing angle; and θ refers to heading angle. Furthermore, A, B, C, and D are the four vertices of the vehicle body.

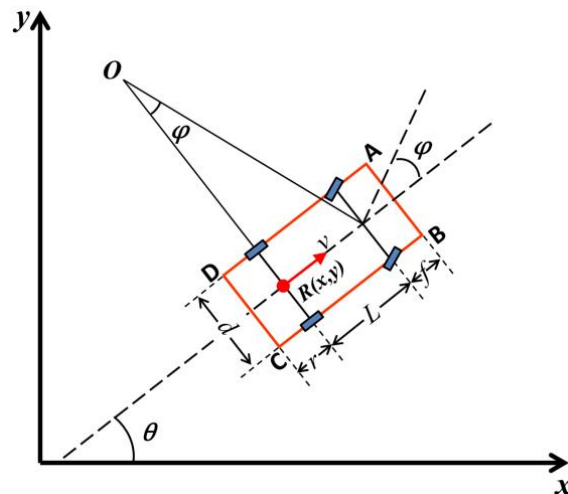


Figure 2. Kinematics model of vehicle.

According to the vehicle kinematics model, kinematics differential equations can be obtained as follows:

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

where a and j refer to vehicle acceleration and jerk of the vehicle, ω denotes the angular velocity of the front wheel pendulum, and t_0 and t_f stand for initial time and final time, respectively.

According to the geometric relationship, the coordinate values of A, B, C, and D can be obtained by the center point of the rear axle $R(x, y)$ and are as follows:

$$\begin{cases} [Ax(t), Ay(t)] = \left(x(t) + (L + f) \cos \theta(t) - \frac{d}{2} \sin \theta(t), y(t) + (L + f) \sin \theta(t) + \frac{d}{2} \cos \theta(t) \right) \\ [Bx(t), By(t)] = \left(x(t) + (L + f) \cos \theta(t) + \frac{d}{2} \sin \theta(t), y(t) + (L + f) \sin \theta(t) - \frac{d}{2} \cos \theta(t) \right) \\ [Cx(t), Cy(t)] = \left(x(t) - r \cos \theta(t) + \frac{d}{2} \sin \theta(t), y - r \sin \theta(t) - \frac{d}{2} \cos \theta(t) \right) \\ [Dx(t), Dy(t)] = \left(x(t) - r \cos \theta(t) - \frac{d}{2} \sin \theta(t), y - r \sin \theta(t) + \frac{d}{2} \cos \theta(t) \right) \end{cases} \quad (2)$$

2.3. Physical Constraints

Considering safety and crew comfort, some state variables and control variables of the system are constrained as follows:

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

where v_{max} stands for the maximum speed with a_{max} for the maximum acceleration, $jerk_{max}$ denotes maximum acceleration change rate, φ_{max} denotes the maximum front wheel swing angle, and ω_{max} is the maximum front wheel swing angle speed. Another purpose of constraining these variables is to make the planned trajectory as smooth as possible so that it will be relatively easier for each vehicle execution system (such as steering system, power system, braking system) to implement trajectory tracking.

2.4. Avoid Collision Constraints

There is no collision allowed while the vehicle is parked. To this end, collision between vehicles and dynamic or static obstacles (e.g., other vehicles, pedestrians, road boundaries) must be considered. The vehicle contour is assumed to be rectangular. Considering the detection accuracy of on-board sensors and the detection blind area in practice, the four-circle model is selected to describe the vehicle contour in this paper, which is shown in Figure 3, where $O_1, O_2, O_3,$ and O_4 denote the center of four vehicles. The geometric relationship is described as follows:

$$\begin{cases} L_1 = L_5 = \frac{1}{8}(r + L + f) \\ L_2 = L_3 = L_4 = \frac{1}{4}(r + L + f) \\ R_v = \sqrt{L_1^2 + (d/2)^2} \end{cases} \quad (4)$$

where L_1 stands for the distance from circle center O_1 to the rear of the vehicle, and L_5 stands for the distance from circle center O_4 to the front of the vehicle. Furthermore, $L_2, L_3,$ and L_4 refer to the distance between $O_1O_2, O_2O_3,$ and O_3O_4 , respectively. R_v denotes the radius of the outer circle.

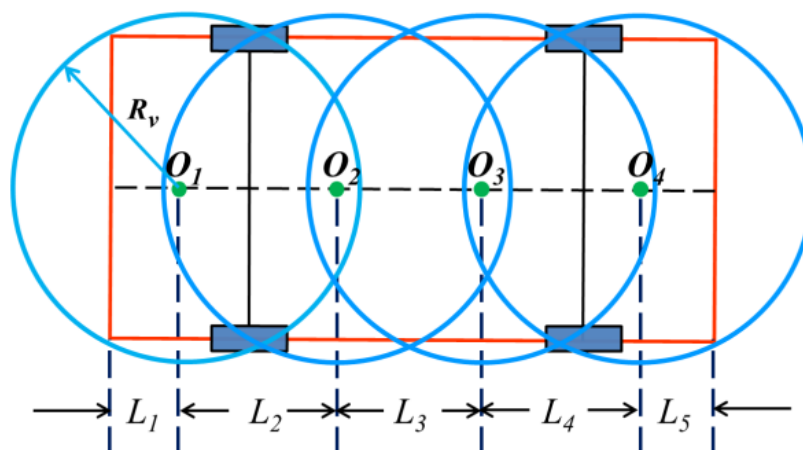


Figure 3. Circle position diagram of vehicle outside contour.

From the above geometric relationships and the center point of the rear axle $R(x, y)$, the positions of four centers are described as follows:

$$\begin{cases} [O_1x(t), O_1y(t)] = (x(t) - (r - L_1) \cos \theta(t), y(t) - (r - L_1) \sin \theta(t)) \\ [O_2x(t), O_2y(t)] = (x(t) + (L_1 + L_2 - r) \cos \theta(t), y(t) + (L_1 + L_2 - r) \sin \theta(t)) \\ [O_3x(t), O_3y(t)] = (x(t) + (L + f - 3L_5) \cos \theta(t), y(t) + (L + f - 3L_5) \sin \theta(t)) \\ [O_4x(t), O_4y(t)] = (x(t) + (L + f - L_5) \cos \theta(t), y(t) + (L + f - L_5) \sin \theta(t)) \end{cases}, t \in [t_0, t_f]$$
 (5)

Similarly, the contours of obstacles are described by multiple circumferential circles (see Figure 3). As shown in Figure 4, if the shapes of the obstacles are similar to a small square, such as pedestrians and trash cans, they can be described by a single circle for convenience. Therefore, the collision-free conditions between vehicles and obstacles can be described by the avoidance of the overlap between the circle of the vehicle contour and that of the obstacle contour. The strict mathematical description of security is shown in Equation (6):

$$(O_ix(t) - Ox(t))^2 + (O_iy(t) - Oy(t))^2 \geq (R_v + R_o)^2$$
 (6)

where O and R_o denote the center and radius of the circumferential circle of the contour of the obstacle, respectively.

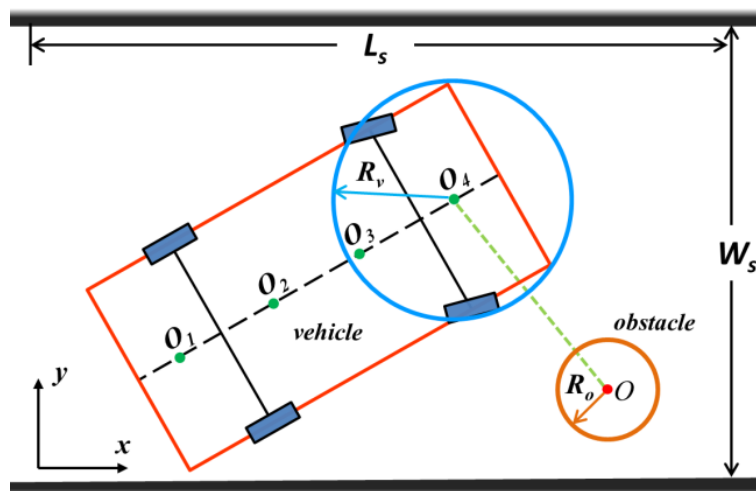


Figure 4. Schematic of avoidance between vehicle and obstacle.

Otherwise, collision between a vehicle and road boundaries has to be considered. The movement of the vehicle is in an area bordered by a road boundary, where L_s and W_s denote the length and width of the road, respectively (see Figure 4). Hence, in the course of the movement, the four vertices of the vehicle, A, B, C, and D, all fall in the area expressed as follows:

$$\begin{cases} [Ax(t), Bx(t), Cx(t), Dx(t)] \in (0, L_s) \\ [Ay(t), By(t), Cy(t), Dy(t)] \in (0, W_s) \end{cases}$$
 (7)

2.5. Terminal Conditions

The satisfaction of the above constraints ensures a collision-free motion of the vehicle from the starting moment t_0 to the ending moment t_f . Moreover, the terminal constraint of the vehicle at t_0 and t_f is supposed to be taken into consideration. In this paper, the initial state and the end state of the vehicle are stationary, that is, $v(t_0) = 0, v(t_f) = 0$. Other states like acceleration at t_0 are selected depending on different scenarios. The four vertices of the vehicle must be inside the parking space at t_f . In addition, it is stipulated that the body of the vehicle should be parallel to the parking space as

shown in Figure 5, where $P(x, y)$ is the upper-left vertex of the vertical parking space. The terminal constraints at t_f are as follows:

$$\begin{cases} Px < [Ax(t_f), Bx(t_f), Cx(t_f), Dx(t_f)] < Px + L_p \\ -W_p < [Ay(t_f), By(t_f), Cy(t_f), Dy(t_f)] < Py \\ \theta(t_f) = \pi/2 \end{cases} \quad (8)$$

where L_p and W_p denote the length and width of the parking space, respectively.

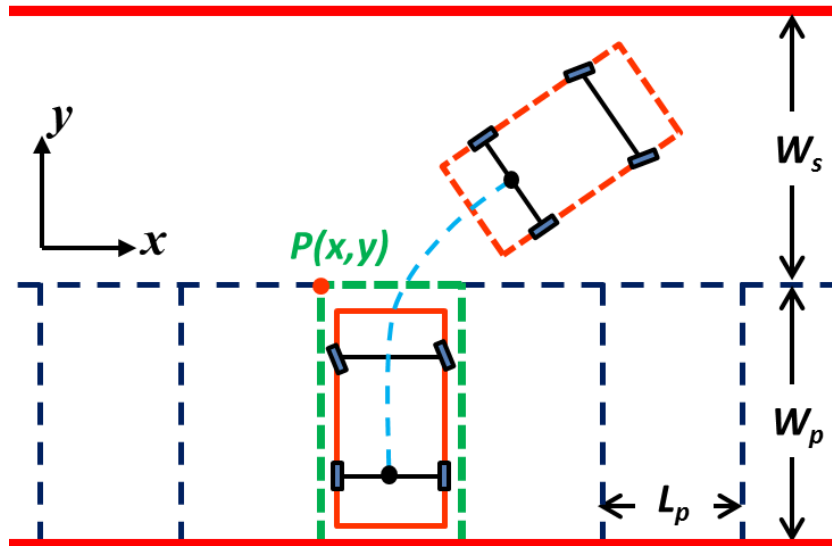


Figure 5. Schematic of the process of vertical parking.

2.6. Trajectory Planning Optimal Control Problem

In this work, the minimum sum of t_f and the driving distance are set as the cost function. Combined with the kinematics model, the collision avoidance constraints, the above-mentioned endpoint constraints, and the optimal control problem of autonomous parking trajectory planning are depicted as follows:

$$J = \min \left(\omega_1 \cdot t_f + \omega_2 \cdot \int_{t_0}^{t_f} v(t) dt \right)$$

$$\text{s.t.} \begin{cases} \text{Differential equations of vehicle dynamics in Eq.(1)} \\ \text{Physical Constraints in Eq.(3)} \\ \text{Avoid collision constraints in Eqs.(4)(5)(6)(7)} \\ \text{Terminal constraints in Eq.(8)} \end{cases} \quad (9)$$

where ω_1 and ω_2 are the weight of t_f and the driving distance, respectively. When $\omega_1 = 1$ and $\omega_2 = 0$, only the optimal t_f is considered in the cost function. In contrast, when $\omega_1 = 0$ and $\omega_2 = 1$, only the shortest driving distance is considered in the cost function. Furthermore, ω_1 and ω_2 are set to be 0.5.

3. Homotopic Method for Solving Optimal Control Problem

In Section 3, a unified trajectory optimization framework for the parking problem is presented. The GPM is utilized to discretize the above optimal control problem into a nonlinear programming problem for solving [37]. Here, the location of the collocation point is determined at first, and the state variables and control variables are discretized into unknown parameters at the collocation points. Then, Lagrange interpolation polynomials are constructed from these collocation points to determine the continuous state variables and control variables. By deriving the above

Lagrange interpolation polynomials, the parking kinematics equation is transformed into algebraic equations at the collocation points, and the integral part of the cost function is discretized by the Gauss integral. Compared with the traditional collocation method and direct shooting method, the GPM, a kind of global collocation method, has better solving accuracy and convergence speed [38,39]. However, for nonsmooth problems, the convergence speed of the GPM is slower. To this end, a homotopic method is used to change the boundary of the obstacle avoidance constraint in the optimal control problem, and the solution of the last optimal control problem is used as the initial value of the next solution. The detailed framework is described as follows:

Step 1. Time domain transformation

The time domain of parking trajectory planning $t \in [t_0, t_f]$ is transformed into a time domain of the GPM in $[-1, 1]$:

$$\tau = \frac{2t - t_f - t_0}{t_f - t_0}, \quad \tau \in [-1, 1] \quad (10)$$

Step 2. Discretization of variables

Lagrange interpolation and Gauss quadrature equations are used to discretize the optimal control problem at a series of Legendre–Gauss (LG) points. The discrete points of the GPM include $N+1$ points composed of NLG points and $\tau_0 = -1$. State variables and control variables are approximated at each configuration point as follows:

$$x(\tau) \approx X(\tau) = \sum_{i=0}^N L_i(\tau) \cdot X(\tau_i) \quad (11)$$

$$\mu(\tau) \approx U(\tau) = \sum_{i=0}^N L_i(\tau) \cdot U(\tau_i) \quad (12)$$

where x and μ stand for state variables and control variables, respectively. X and U refer to discrete variables, and $L_i(\tau)$ denotes Lagrange interpolation basis functions.

Step 3. Transformation of state equation and constraint equation

After the derivation of Equation (11), the obtained derivative of a state variable at collocation point $\tau = \tau_k$ can be approximated as follows:

$$\dot{x}(\tau_k) \approx \dot{X}(\tau_k) = \sum_{i=0}^N \dot{L}_i(\tau_k) \cdot X(\tau_i) = \sum_{i=0}^N D_{k,i} \cdot X(\tau_i) \quad (13)$$

where D denotes a differential state matrix, which represents the differential value of the Lagrange interpolation basis function at each collocation point. Through Equation (13), a state equation can be transformed into equality constraints, as follows:

$$\sum_{i=0}^N D_{k,i} \cdot X(\tau_i) - \frac{t_f - t_0}{2} f(X(\tau_k), U(\tau_k), \tau_k) = 0 \quad (14)$$

The obstacle avoidance constraints and endpoint constraints in parking trajectories are also discretized at the collocation points. The obtained equality constraints are as follows:

$$C(X(\tau_k), U(\tau_k), \tau_k) \leq 0 \quad (15)$$

$$E(X(\tau_0), \tau_0, X(\tau_f), \tau_f) = 0 \quad (16)$$

where C and E denote the obstacle avoidance constraint and endpoint constraint, respectively.

Step 4. Transformation of cost function

The integral term in the cost function can be approximated by the Gauss integral method. In this way, the cost function is transformed as follows:

$$J = \Phi(X(t_0), t_0, X(t_f), t_f) + \frac{t_f - t_0}{2} \sum_{i=0}^N \omega_i G(X(\tau_i), U(\tau_i), \tau_i) \quad (17)$$

where ω_i stands for the integral weight.

Step 5. The homotopic transformation of constraint equation

The optimal control problem of parking trajectory planning becomes unsmooth with the increase of environment complexity, which makes it difficult to solve the original problem by directly using the GPM. Thus, this paper proposes a homotopic method to improve the smoothness of the original problem. In the proposed method, the bounds of obstacle avoidance constraints in the original problem are taken as homotopic parameters to make the original problem of parking trajectory planning smoother. In this way, the unsmooth original problem is directly solved without changing the form of the original problem. In addition, the optimal solution is used as the initial value of the next calculation. To this end, Equation (15) is transformed as follows:

$$C(X(\tau_k), U(\tau_k), \tau_k) \leq H_c \quad (18)$$

where H_c denotes the homotopic parameter vector. To improve the convergence of the generated new problem, H_c is added, releasing the constraint boundaries. Therefore, when H_c evolves into 0, the problem becomes the original problem. Equation (18) can also be transformed as follows:

$$C(X(\tau_k), U(\tau_k), \tau_k) - H_c \leq 0 \quad (19)$$

It is worth noting that Equation (19) has the same form as Equation (15). Therefore, the solution of the original problem can be obtained by constant iteration with a homotopic method. Moreover, the result of the last optimal control problem is taken as a better initial value, which reduces the difficulty of the next solution. As a result, the convergence speed of the solution is effectively improved.

After the five steps above, the optimal control problem of parking trajectory planning is transformed into a discrete NLP problem by the GPM, and SNOPT is adopted as the software package in this paper. The overall flowchart of the proposed GPM-based methodology is demonstrated in Figure 6.

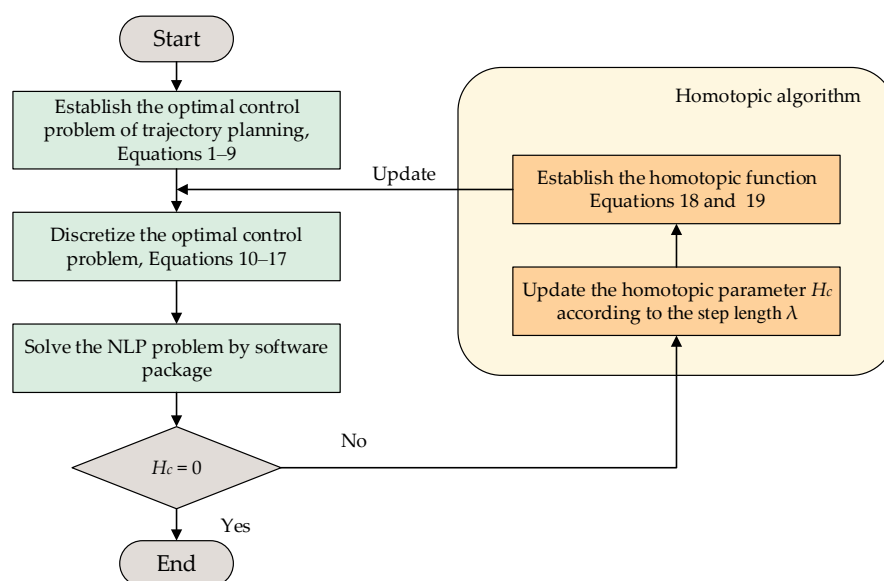


Figure 6. Flowchart of Gauss pseudospectral method (GPM)-based methodology for parking trajectory planning.

4. Simulation Results and Discussion

The trajectory planning of AVP is separated into two phases. Two obtained optimal trajectories of corresponding phases are spliced into a complete trajectory of the whole AVP. The code is executed on a laptop running Windows 7 with an Intel(R) Core(TM) i5-3210M 2.50 GHz processor and 8 GB RAM, written in MATLAB. The vehicle parameters and physical constraints used in the simulation are shown in Table 1.

Table 1. Vehicle parameters and physical constraints.

Parameter	Description	Value
L	Wheelbase	2.560 m
f	Front overhang	0.902 m
r	Rear overhang	0.883 m
d	Width	1.765 m
v_{max}	Maximum speed	3.0 m/s
a_{max}	Maximum acceleration	0.75 m/s ²
$jerk_{max}$	Maximum acceleration rate of change	0.3 m/s ³
φ_{max}	Maximum front wheel swing angle	0.56 rad
ω_{max}	Maximum front wheel swing angular velocity	0.56 rad/s

4.1. Trajectory of Parking Space Searching Phase

In the first phase of the parking process, the vehicle moves from the entrance of the parking lot to the initial area of vertical parking. In this phase, it is assumed that there are no other dynamic obstacles, such as vehicles and pedestrians, but only some constraints of road boundary and parking space. The position of the vehicle at the entrance and the initial area of vertical parking are defined as follows:

$$\begin{cases} [Rx(t_0), Ry(t_0)] = (25.5, 17.5) \\ 24 \leq Ax(t_f), Bx(t_f), Cx(t_f), Dx(t_f) \leq 29 \\ 0.5 \leq Ay(t_f), By(t_f), Cy(t_f), Dy(t_f) \leq 4 \end{cases} \quad (20)$$

According to the above-mentioned algorithm, the trajectory planning problem of the first phase is solved without constraints, and the trajectory of the vehicle (see Figure 7a) is a circular arc. The termination condition is that all the vertices of the vehicle fall in the initial parking area. In this context, the vehicle reaches the top, rather than the middle, of the initial parking area to optimize the cost function, which is shown in Figure 7a. It is also found out that the trajectory intersects with the parking spaces in the middle of the parking lot, instead of those at the bottom or top of the parking lot. Therefore, by using the above-mentioned homotopic algorithm, the constraints of the middle parking spaces are gradually approximated to the constraints of the original optimal control problem. Then vehicle trajectories from the entrance to the initial area of parking are obtained by iterative solution, which are shown in Figure 7b,c.

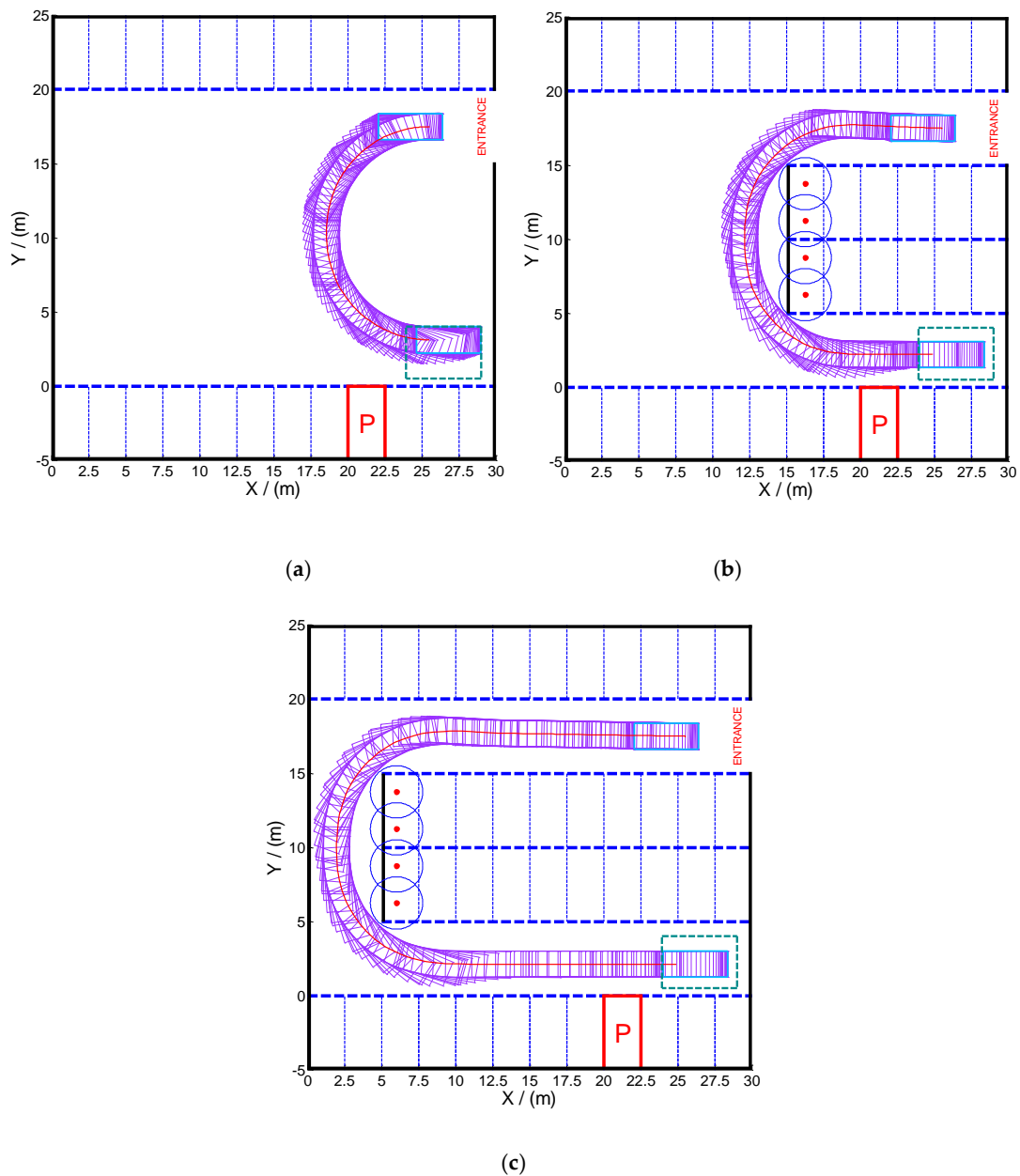


Figure 7. Trajectory planning in the parking space searching phase: (a) depicts the vehicle trajectory from the entrance of the parking lot to the initial area of vertical parking; (b,c) depict the vehicle trajectory under partial and complete constraints, respectively.

4.2. Trajectory in Vertical Parking Phase

In the second phase of the parking process, the convenience of passengers in getting on and off has to be considered. To this end, the minimum width of the space on both sides of the vehicle when the parking ends is set to 0.3. Therefore, the minimum parking length L_p is set to 2.3, and the parking width W_p is set to 5.0. Additionally, based on the range of the initial parking area, the road width W_s is set to 4.5 (see Figure 5). For the convenience of calculation, the upper-left vertex of the vertical parking space is set as the origin of the coordinate system, that is, $[Px, Py] = (0, 0)$. Furthermore, the coordinate of the central point of the rear axle at the starting position is $[Rx(t_0), Ry(t_0)] = (5, 1.5)$. In addition, the termination condition of parking completion is required to meet the constraints in Equation (8), and the equation of $Ry(t_f) = L_p/2$ to ensure that the vehicle is in the middle of the parking space at the ending moment.

According to the above-mentioned homotopic algorithm, the parking space L_p gradually decreases from 3.5 to the minimum value. In other words, the homotopic parameter $H_c = [3.5 : \lambda : 2.3]$, where the step length $\lambda = 0.02$. Hence, the problem is solved with 61 times of iteration, and there is only a difference of L_p between each iteration process. Then, four vertical parking processes with different L_p values are selected for analysis and discussion. The values of L_p in Cases 1–4 are set as 3.50, 3.12, 2.72, and 2.30, respectively. The optimized trajectory, control variables, and state variables are illustrated in Figures 8–15.

First of all, all vehicles can complete vertical parking and meet the above-mentioned termination constraints in Cases 1–4, which means that the vehicle is in the middle of the parking space when the parking ends. The control variables all present the feature of the bang–bang control (see Figure 9, Figure 11, Figure 13, and Figure 15) because time is required to be optimal in cost function. That is to say that the state variables are continuous and smooth despite that the control variables are oscillated.

As shown in Figures 8 and 10, the trajectories in Cases 1 and 2 are similar. The adjustment of the vehicle occurs outside the parking space rather than inside, even if the parking space is relatively larger in Cases 1 and 2. This phenomenon is similar to our daily manual vertical parking process because the narrow parking space is not conducive to the adjustment of the vehicle posture.

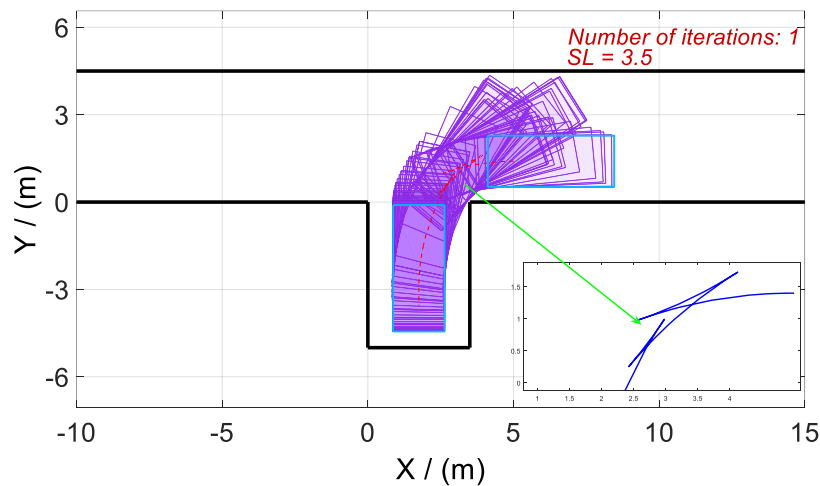


Figure 8. Optimized trajectories for Case 1 and $t_f = 23.9753$ s.

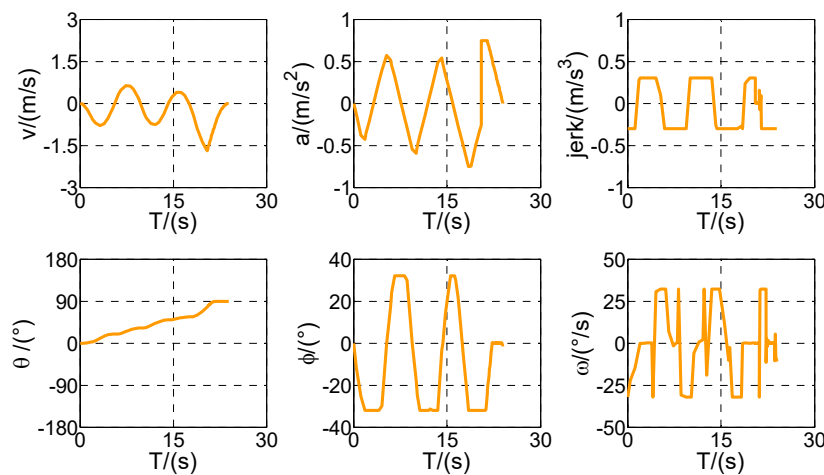


Figure 9. Optimized control and state variables in Case 1.

As shown in Figures 10–15, parking time t_f shows a growing trend with the gradual decrease of L_p . Furthermore, the t_f of Case 4 is the longest, and the reason is obvious. It is because of the narrow

parking space resulting in six times of direction changes in Case 4, compared with only four times in Cases 1–3.

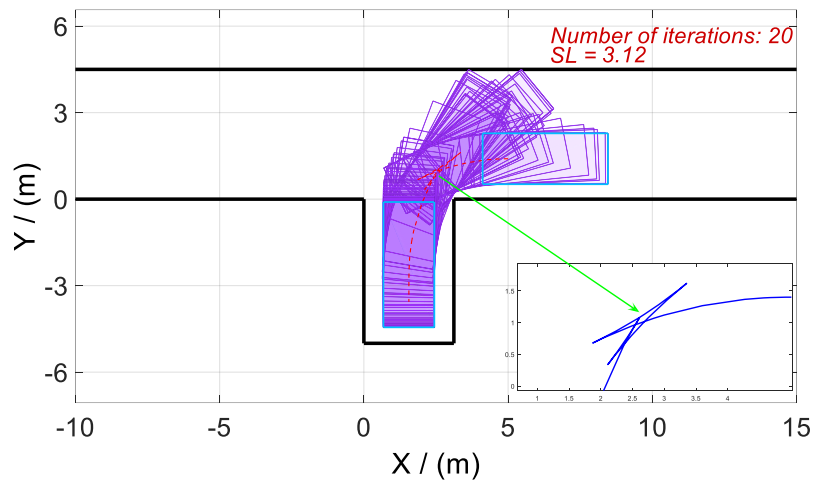


Figure 10. Optimized trajectories for Case 2 and $t_f = 24.1324$ s.

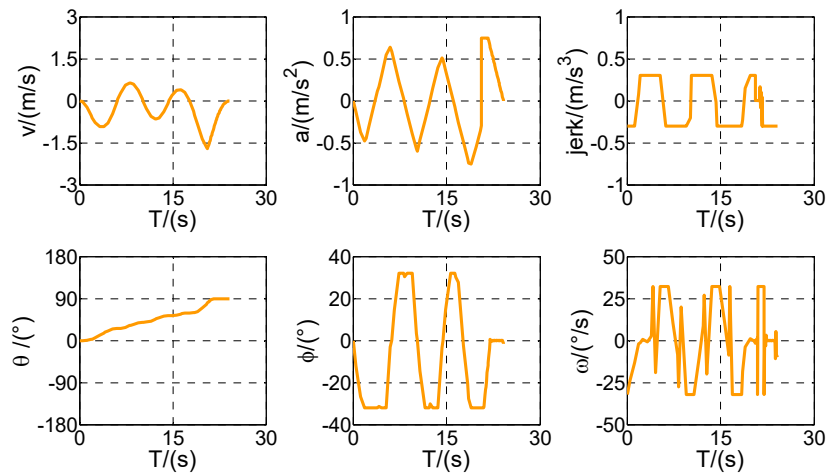


Figure 11. Optimized control and state variables in Case 2.

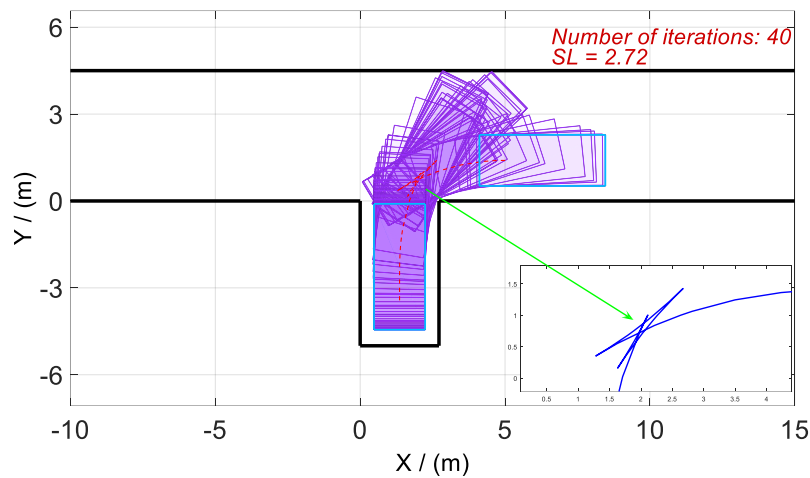


Figure 12. Optimized trajectories for Case 3 and $t_f = 24.4173$ s.

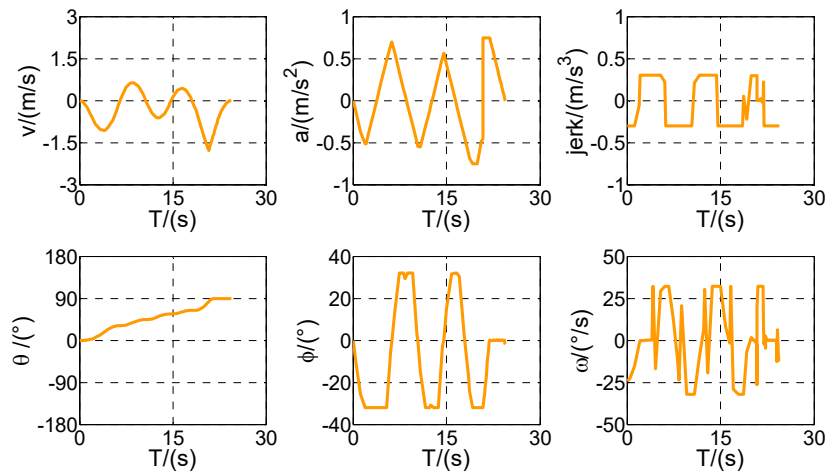


Figure 13. Optimized control and state variables in Case 3.

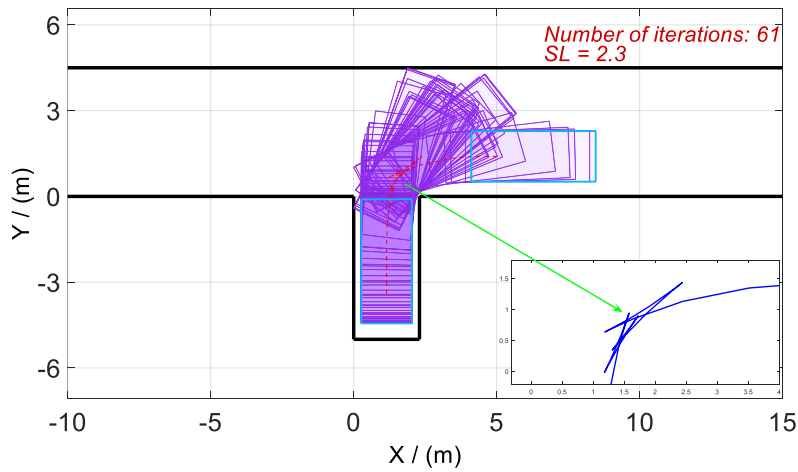


Figure 14. Optimized trajectories for Case 4 and $t_f = 31.1608$ s.

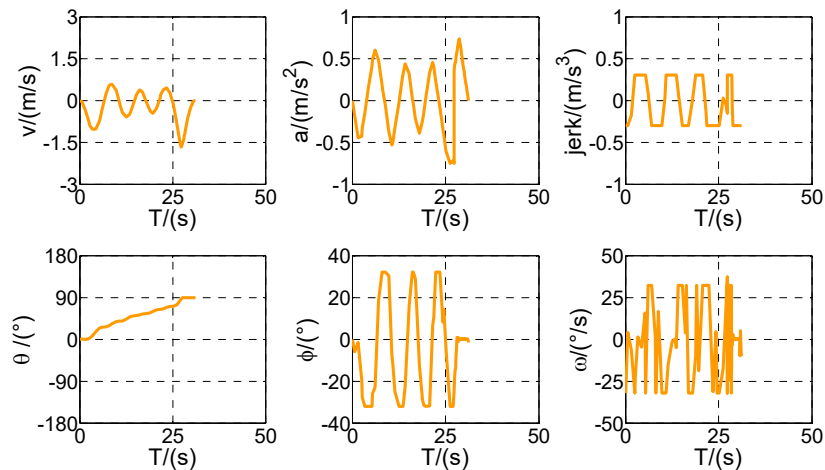


Figure 15. Optimized control and state variables in Case 4.

The traditional GPM takes a long time to directly solve the optimal parking problem, and even ends with nonconvergence when the parking space is narrow. Thus, this paper adopts a homotopic method to select a rational initial guess, which can solve this problem effectively. As depicted in Figure 16, the iteration time is less than 1 s in more than 90% of the cases. And the longest calculation time does not exceed 2 s. Longer iteration times occur at the beginning and end of the calculation for

two reasons: (i) In the first iteration, the constraint is set to the calculation for the first time, and the increase of the constraint condition leads to the extension of the calculation time. (ii) At the end of the iteration, the rear space is narrow, so the process of trajectory optimization becomes difficult.

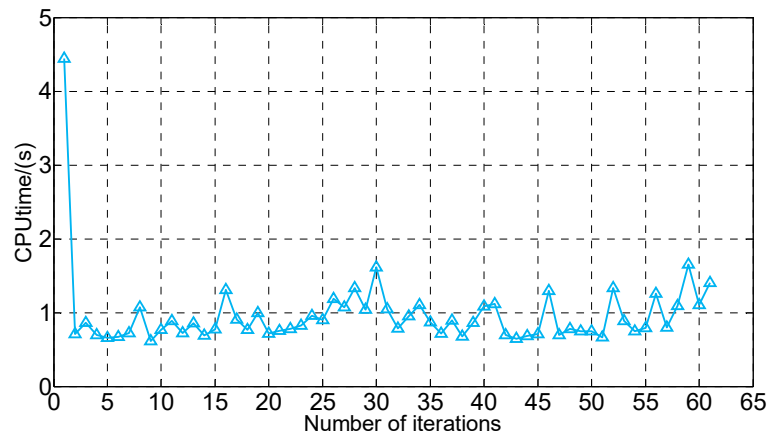


Figure 16. Computation time for each iteration with L_p from 3.5 to 2.3 m.

4.3. Online Computing Method for Vertical Parking Trajectory Planning

The offline calculation method of vertical parking trajectory planning is illustrated above. In the process of approaching the original optimal control problem by homotopic parameters, the overall calculation time is long even if the time cost of each iteration is short. At this time, real-time requirements will not be met. Thus, an online planning method of parking trajectory is proposed. Let us assume that the trajectories calculated by each iteration in the second phase are stored as a table. If the starting position and parking space size in practice are the same as in any case stored, the trajectory can be obtained directly from the stored trajectories by looking up the tables without any calculation. However, these ideal situations barely happen in practical applications. Considering this problem, the stored trajectory is used as an initial guess rather than an actual trajectory. The complete process is as follows: First, a parking space is found, and the vehicle is parked in its initial area. In this context, an optimal control problem for parking trajectory planning is formed. Then, the most similar set of trajectories is found from the stored data based on actual parking space size and the starting position, and used as the initial guess of the actual optimal control problem, which is solved by the GPM. The advantage is that the solving speed can be effectively improved by the proposed method. This is because the corresponding original problem of the found trajectory is similar to the current problem, which makes this trajectory similar to the current solution, being a good initial guess.

To verify the above method, a working condition with $L_p = 2.5$, $[Rx(t_0), Ry(t_0)] = (5, 1.5)$ is assumed, and its parking trajectory, used as the initial guess of the subsequent optimal control problem, is stored in the data table. Cases 5 and 6 are set up according to the distance from the actual parking starting position to point R. In each case, there are four vehicles whose start locations $R'(x, y)$ are in the circles that take $R(x, y)$ of the initial guess as the center and 0.2 and 0.5 as radii (as shown in Table 2). The GPM is also taken to solve the new optimal control problem for vertical parking trajectory planning.

Table 2. Actual parking start position.

ID	Case 5		Case 6	
	$R'x(t_0)$	$R'y(t_0)$	$R'x(t_0)$	$R'y(t_0)$
Vehicle 1	5.0	1.7	5.0	2.0
Vehicle 2	5.2	1.5	5.5	1.5
Vehicle 3	5.0	1.3	5.0	1.0
Vehicle 4	4.8	1.5	4.5	1.5

As shown in Figure 17a,b, the new optimal control problems in Cases 5–6 can be solved quickly, and the solving speed of Vehicle 4 is the quickest, followed by that of Vehicle 2. For Vehicles 2 and 4, the vertical positions are the same as those in the initial guess, that is, $R'y(t_0) = Ry(t_0)$, despite differences in the horizontal position. Therefore, their trajectories at the beginning phase of the parking are closer to the optimal ones. In contrast, the vertical positions of Vehicles 1 and 3 are different from those in the initial guess despite the same horizontal positions, which leads to a large deviation of the trajectories at the beginning phase. Therefore, the calculation time increases.

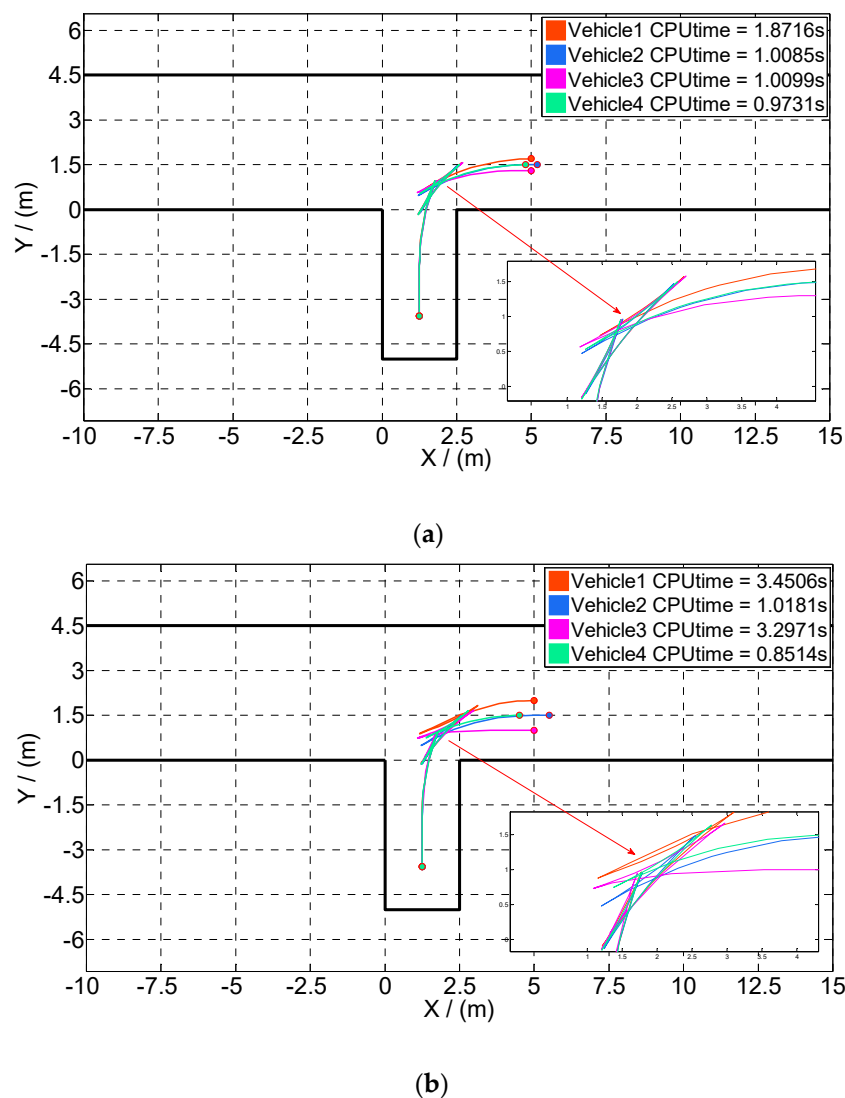


Figure 17. Optimal trajectory and calculation time obtained by online calculation of vertical parking. As depicted in Figure 17 (a) and (b), the starting points of vertical parking are in circles, taking the starting point of the initial guess as the center and 0.2 and 0.5 as radii.

In Case 5, the maximal calculation time of all the vehicles is 1.87 s, which is acceptable in real-time parking applications. However, in Case 6, the calculation time of Vehicles 1 and 3 is more than 3 s. Comparing Case 5 and Case 6, it can be noted that the large deviation, especially horizontal deviation, of vehicle trajectory from the initial guess leads to a greater impact on calculation time. Therefore, the calculation time of the optimal control problem increases. In contrast, the deviation of the horizontal position has less effect on the calculation speed.

To verify the effectiveness and real-time performance of the proposed homotopic strategy, the calculation results of the two initialization methods are selected for comparison. The CPU time and

t_f of Cases 1–4 are recorded, and fail means nonconvergence in the calculation process. As presented in Table 3, the simulation costs of the proposed homotopic strategy are much lower than those of the other two strategies. Especially in conditions with small space (Cases 3 and 4), the calculation costs of the incremental strategy and the spatiotemporal decomposition strategy increase significantly, and the calculation may fail. We cite the following reasons to explain the excellent real-time performance of the proposed strategy: In the iterative solution process, only the width of the parking space changes according to the calculation step. Thus, optimal control problems of two adjacent calculations are very similar. In the homotopic strategy, the optimal solution of the previous calculation is used as the initial guess for each solution. And the calculation is guided by a high-quality initial guess. It is worth noting that since there is no initial guess stored when solving the problem of Case 1, the calculation time is slightly longer than that of other cases. This result is consistent with the conclusion in Figure 16.

Table 3. Comparison of the performances of distinct strategies.

	Incremental Strategy [40]		Spatiotemporal Decomposition [41]		Homotopic Strategy	
	CPU Time	t_f	CPU Time	t_f	CPU Time	t_f
Case 1	47.2232	22.4859	41.9217	25.5481	4.4415	23.9753
Case 2	42.9378	37.6651	65.5836	27.6475	0.7151	24.1324
Case 3	87.6384	fail	89.3382	28.8263	1.0877	24.4173
Case 4	115.4827	fail	92.7271	31.0379	1.4082	31.1608

As a summary, the processes of the online calculation of vertical parking trajectory are as follows: (i) A matrix, whose size is $R_{m \times n}$, is formed with starting points selected in the initial area of vertical parking. The optimal parking trajectories of these starting points are obtained by an offline method and stored. (ii) The starting point with the minimum deviation from the actual location is obtained by looking up the table of stored data. Note that the vertical deviation minimum takes higher priority than the horizontal one. Finally, the trajectory with the minimum deviation is taken as the initial guess of the optimal control problem. This optimal control problem is solved by the GPM, and an optimal trajectory of the actual parking is obtained.

5. Conclusions

In this paper, a method that combines the GPM and homotopic method to solve the optimal control problem is proposed for the trajectory planning of AVP. An optimal control problem is determined to describe the optimization problem of parking trajectory. Then the GPM is used to transform this optimal control problem into a nonlinear programming problem, which can be solved by sequence quadratic programming. The difficulty of solving is handled by a homotopic method, which adjusts the constraint boundary. The original problem is approached by iteration where the last solution result is taken as the initial value of the next solution. AVP trajectory planning is divided into two phases, and an online calculation method for the vertical parking phase is proposed. Furthermore, simulation results indicate that the proposed method can effectively improve the calculation speed and convergence of the solution. In the vertical parking phase, the online calculation time of trajectory planning is less than 2 s, which meets the real-time requirement.

In our future works, various sizes of parking scenarios will be taken into account. Furthermore, we will study the effective solving of the multivehicle coordinated trajectory of connected and automated vehicles in a parking lot.

Author Contributions: Conceptualization, C.C. and B.W.; methodology, B.W.; software, L.X.; data curation, J.C.; writing—original draft preparation, B.W.; writing—review and editing, C.C. and T.W.; funding acquisition, L.Q. The first two authors, C.C. and B.W., contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Fund of China (Grant No. 51875149).

Acknowledgments: The authors acknowledge the support of the ICVE Laboratory of Hefei, Anhui Province, China.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lin, T.; Rivano, H.; Le Mouël, F. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [[CrossRef](#)]
- Dinh, T.; Kim, Y. A novel location-centric IoT-cloud based on-street car parking violation management system in smart cities. *Sensors* **2016**, *16*, 810. [[CrossRef](#)] [[PubMed](#)]
- Banzhaf, H.; Nienhüser, D.; Knoop, S.; Zöllner, J.M. The future of parking: A survey on automated valet parking with an outlook on high density parking. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Long Beach, LA, USA, 11–14 June 2017; pp. 1827–1834.
- Chirca, M.; Roland, C.; Roland, L. Autonomous valet parking system architecture. In Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 2619–2624.
- Huang, K.Y.; Chang, S.B.; Tsai, P.R. The advantage of the arduino sensing system on parking guidance information systems. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, China, 10–13 December 2017; pp. 2078–2082.
- Chen, X.; Wang, S.; Shi, C.; Wu, H.; Zhao, J.; Fu, J. Robust ship tracking via multi-view learning and sparse representation. *J. Navigation.* **2019**, *72*, 176–192. [[CrossRef](#)]
- Han, S.J.; Choi, J. Parking space recognition for autonomous valet parking using height and salient-line probability maps. *Etri. J.* **2015**, *37*, 1220–1230. [[CrossRef](#)]
- Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [[CrossRef](#)]
- Reed, J.; Shepp, L. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [[CrossRef](#)]
- Sungwoo, C.; Boussard, C.; d’Andréa-Novel, B. Easy path planning and robust control for automatic parallel parking. *IFAC Pro. Vol.* **2011**, *44*, 656–661.
- Vorobieva, H.; Glaser, S.; Minoiu-Enache, N.; Mammar, S. Automatic parallel parking in tiny spots: Path planning and control. *IEEE Trans. Intell. Transp.* **2014**, *16*, 396–410. [[CrossRef](#)]
- Maekawa, T.; Noda, T.; Tamura, S.; Ozaki, T.; Machida, K.I. Curvature continuous path generation for autonomous vehicle using B-spline curves. *Comput. Aided. Design.* **2010**, *42*, 350–359. [[CrossRef](#)]
- Zhang, S.; Simkani, M.; Zadeh, M.H. Automatic vehicle parallel parking design using degree polynomial path planning. In Proceedings of the IEEE Vehicular Technology Conference (VTC Fall), San Francisco, CA, USA, 5–8 September 2011; pp. 1–4.
- González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1135–1145. [[CrossRef](#)]
- Vega-Brown, W.; Roy, N. Asymptotically optimal planning under piecewise-analytic constraints. In *Algorithmic Foundations of Robotics XII*; Goldberg, K., Abbeel, P., Bekris, K., Miller, L., Eds.; Springer: New York, NY, USA, 2020; pp. 528–543.
- Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
- Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A*. *J. ACM* **1985**, *32*, 505–536. [[CrossRef](#)]
- Kim, C.; Suh, J.; Han, J.H. Development of a hybrid path planning algorithm and a bio-inspired control for an omni-wheel mobile robot. *Sensors* **2020**, *20*, 4258. [[CrossRef](#)] [[PubMed](#)]
- Siméon, T.; Laumond, J.P.; Nissoux, C. Visibility-based probabilistic roadmaps for motion planning. *Adv. Robot.* **2000**, *14*, 477–493. [[CrossRef](#)]
- Yang, K.; Moon, S.; Yoo, S.; Kang, J.; Doh, N.L.; Kim, H.B.; Joo, S. Spline-based RRT path planner for non-holonomic robots. *J. Intell. Robot. Syst.* **2014**, *73*, 763–782. [[CrossRef](#)]
- Zhang, B.; Chen, W.; Fei, M. An optimized method for path planning based on artificial potential field. In Proceedings of the sixth International Conference on Intelligent System Design and Applications ISDA, Jinan, China, 16–18 October 2006; pp. 35–39.

22. Huy, Q.; Mita, S.; Yoneda, K. A practical and optimal path planning for autonomous parking using fast marching algorithm and support vector machine. *IEICE. Trans. Inf. Syst.* **2013**, *96*, 2795–2804.
23. Syed, A.; Kunwar, F.; Iqbal, M. Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots. *Robot. Auton. Syst.* **2014**, *62*, 474–486. [[CrossRef](#)]
24. Chen, W.J.; Jhong, B.G.; Chen, M.Y. Design of path planning and obstacle avoidance for a wheeled mobile robot. *Int. J. Fuzzy. Syst.* **2016**, *18*, 1080–1091. [[CrossRef](#)]
25. Ahn, C.W.; Ramakrishna, R.S. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE. Trans. Evolut. Comput.* **2002**, *6*, 566–579.
26. Zhang, Y.; Gong, D.W.; Zhang, J.H. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing* **2013**, *103*, 172–185. [[CrossRef](#)]
27. Bakdi, A.; Hentout, A.; Boutami, H.; Maoudj, A.; Hachour, O.; Bouzouia, B. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robot. Auton. Syst.* **2017**, *89*, 95–109. [[CrossRef](#)]
28. Chen, Y.; Liang, J.; Wang, Y.; Pan, Q.; Tan, J.; Mao, J. Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics. *Soft Comput.* **2020**, *24*, 1–17. [[CrossRef](#)]
29. Jafari, M.; Xu, H.; Carrillo, L.R.G. Brain emotional learning-based path planning and intelligent control co-design for unmanned aerial vehicle in presence of system uncertainties and dynamic environment. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1435–1440.
30. Lim, W.; Lee, S.; Sunwoo, M.; Jo, K. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 613–626. [[CrossRef](#)]
31. Bai, G.; Liu, L.; Meng, Y.; Luo, W.; Gu, Q.; Ma, B. Path tracking of mining vehicles based on nonlinear model predictive control. *Appl. Sci.* **2019**, *9*, 1372. [[CrossRef](#)]
32. Zips, P.; Böck, M.; Kugi, A. Optimisation based path planning for car parking in narrow environments. *Robot. Auton. Syst.* **2016**, *79*, 1–11. [[CrossRef](#)]
33. Li, B.; Shao, Z. A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowl-Based. Syst.* **2015**, *86*, 11–20. [[CrossRef](#)]
34. Li, B.; Shao, Z. Precise trajectory optimization for articulated wheeled vehicles in cluttered environments. *Adv. Eng. Softw.* **2016**, *92*, 40–47. [[CrossRef](#)]
35. Dariani, I.R.; Lobig, T.; Lauermann, J.; Rieck, J.; Schindler, J. Trajectory planning for cooperative autonomous valet parking. In Proceedings of the 14th Magdeburger Maschinenbautage, Magdeburg, Germany, 24–25 September 2019.
36. Chien, C.F.; Chen, H.T.; Lin, C.Y. A low-cost on-street parking management system based on bluetooth beacons. *Sensors* **2020**, *20*, 4559. [[CrossRef](#)]
37. Garg, D.; Hager, W.W.; Rao, A.V. Pseudospectral methods for solving infinite-horizon optimal control problems. *Automatica* **2011**, *47*, 829–837. [[CrossRef](#)]
38. Huntington, G.T.; Rao, A.V. Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method. *J. Guid. Control. Dynam.* **2008**, *31*, 689–698. [[CrossRef](#)]
39. De Groote, F.; Kinney, A.L.; Rao, A.V.; Fregly, B.J. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. *Ann. Biomed. Eng.* **2016**, *44*, 2922–2936. [[CrossRef](#)]
40. Li, B.; Shao, Z. An incremental strategy for tractor-trailer vehicle global trajectory optimization in the presence of obstacles. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 1447–1452.
41. Li, B.; Zhang, Y.; Shao, Z. Spatio-temporal decomposition: A knowledge-based initialization strategy for parallel parking motion optimization. *Knowl-Based Syst.* **2016**, *107*, 179–196. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).