

Article

Not peer-reviewed version

---

# Research on Automatic Vertical Parking Path Planning Algorithm for Narrow Parking Spaces

---

[Yanfeng Wu](#)<sup>\*</sup>, Xuan Li, [Jianping Gao](#), Xinling Yang

Posted Date: 21 August 2023

doi: 10.20944/preprints202308.1474.v1

Keywords: narrow parking spaces; automatic parking; multi-objective evaluation function; B-spline curve; gradient descent



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Article

# Research on Automatic Vertical Parking Path Planning Algorithm for Narrow Parking Spaces

Yanfeng Wu <sup>1</sup>, Xuan Li <sup>1</sup>, Jianping Gao <sup>1,\*</sup> and Xinling Yang <sup>2</sup>

<sup>1</sup> College of Vehicle and Transportation Engineering, Henan University of Science and Technology, Luoyang 471003, China; yfwu@haust.edu.cn (Y.W.); xuanli9298@163.com(X.L.); gaojp@haust.edu.cn(J.G.)

<sup>2</sup> School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471003, China; y13148059869@163.com

\* Correspondence: gaojp@haust.edu.cn

**Abstract:** The narrow parking space will result in difficulties in path planning and spot turning caused by sudden changes and discontinuities in path curvature. To address the problem, this paper investigates the performance of three path planning algorithms and proposes a path optimization algorithm. First, the narrow parking space is defined based on single-step parking of the arc-line combination parking algorithm. Second, to compare the arc-line combination algorithm, Hybrid A\* algorithm, and particle swarm optimization parking algorithm under different narrow parking spaces, a multi-objective evaluation function is proposed, including three evaluation indicators, namely, the path length, the number of positive and negative conversions of vehicle speed, and the smoothness of the path. Their performance is compared by simulation conducted in MATLAB, the results show that the Hybrid A\* algorithm is the better than others for narrow parking spaces. Third, to smooth the planning path, a path optimization algorithm based on the cubic B-spline curve and gradient descent is proposed. Finally, the simulation is conducted between the proposed algorithm and Hybrid A\* algorithm, the result shows that the proposal can effectively eliminate the curvature mutation point and constrain the path curvature to meet the requirements of a smooth path.

**Keywords:** narrow parking spaces; automatic parking; multi-objective evaluation function; B-spline curve; gradient descent

## 1. Introduction

With the continuous rise of car ownership, the availability of parking spaces is decreasing, leading to increasingly narrow parking areas, which will cause the parking difficult and inefficient for drivers, even result in collision between the vehicle and the surroundings. It can also cause traffic jams and even traffic accidents. Therefore, automatic parking technology under narrow parking spaces has gradually become the research focus.

The automatic parking system mainly uses sensors of the vehicle to measure the relative distance, speed, and angle between the vehicle and the surrounding objects. It calculates the operation process through the on-board computing or cloud computing platform. It controls the vehicle's steering, acceleration and deceleration to achieve automatic parking. The parking process can be divided into four steps, namely, environmental perception, parking space detection and recognition, parking path planning, parking path following control. Among them, path planning is to plan a path avoiding collision under vehicle dynamics constraints based on environment and location information and is executed on the vehicle [1].

At present, there are three kinds of vertical parking path planning algorithms commonly used in traditional standard parking spaces. One is the geometric algorithm. Zhao, B et al. [2] used arc lines as transition paths in local path planning to park a vehicle, including arcs and line segments based on the optimal parking starting point node, although the parking path is shortened, it is only suitable for single-step parking. Jiang, M et al. [3] proposed a vertical parking path planning algorithm based on polynomial curve optimization to improve the safety and success rate of automatic vertical

parking in complex environments, although it can solve the problem of not being able to single step parking, this algorithm requires a wider parking space. Zhang, J et al. [4] proposed a vertical parking trajectory planning algorithm based on a cycloid curve, decoupling the vertical parking trajectory planning problem into a path planning problem and a velocity planning problem, but it can only be used in slightly narrow parking spaces. The arc-line combination algorithm in the geometric algorithm has accurate characteristics, which can accurately calculate the driving trajectory of the vehicle in the parking process so that the vehicle can accurately park into the specified parking space.

The second is a random search algorithm. Yu, L et al. [5] adopted a particle swarm optimization algorithm to solve equality constraints based on parking boundary, inequality constraints based on curvature, collision avoidance, and parking space reduction. It obtained optimized, smooth, and curvature continuous programming path curve expressions by taking the weighted sum of maximum curvature and the absolute value of the horizontal coordinate of the starting position of parking as the objective function. However, due to the large number of parameters, the path search efficiency is low. Kim, M et al. [6] proposed a Target Tree RRT\* algorithm in complex environments, which uses the clothoid path to design the target tree to deal with curvature discontinuity. To further reduce the planning time, a cost function is defined to initialize an appropriate target tree considering obstacles. Combining the optimal variable RRT and searching for the shortest path, the Target Tree RRT\* algorithm obtains an approximately optimal path as the sampling time increases, although the path search time is shortened, the smoothness of the path cannot be guaranteed. Zhang, J et al. [7] proposed a parallel parking path planning algorithm based on the improved particle swarm optimization, comprehensively considered the non-holonomic kinematics constraints of the 4WS by wire vehicle, process constraints and boundary constraints of the power and steering subsystems, obstacle avoidance constraints, initial parking posture, and target parking posture constraints, and established a parallel parking path planning constraint optimization problem to minimize the total duration of the parking process, the particle swarm optimization, which can deal with equality constraints and inequality constraints, is used to solve the problem and obtain the optimal parallel parking path, although it can solve the problem of non-smooth path, it is not suitable for narrow parking spaces. The search process of the particle swarm optimization algorithm in the random search algorithm performs parallel search, the computational efficiency is high, and the parking path that meets the conditions can be found in a short time. The principle of the particle swarm optimization algorithm is simple, and the parameter adjustment is relatively simple. For different parking scenarios, only a few parameters need to be adjusted.

The third type is the graph search algorithm. Sedighi, S et al. [8] proposed an efficient computational algorithm that combines the Hybrid A\* search algorithm with visibility graph planning to find the shortest non-holonomic path in a mixed (continuous-discrete) environment for automatic parking, although the relatively short path can be searched, the search time is increased. Bai, J et al. [9] proposed a directed Hybrid A\* global path planning algorithm based on the generalized Vinor map for the path planning problem of autonomous valet parking systems, which accurately and effectively generates a collision-free path from the entrance of the parking lot to the starting point of parking. Xiong, L et al. [10] improved the Hybrid A\* by adding a penalty term for obstacle distance on the Reeds-shepp (RS) curve, which improved the problem of the RS curve being too close to obstacles and improved the effectiveness of parking. The Hybrid A\* algorithm in the graph search algorithm is a heuristic search in a continuous coordinate system, which can ensure that the generated trajectory satisfies the vehicle non-integrity constraint (kinematic constraint). It can deal with non-smooth obstacles and find the optimal global solution. By combining the discrete node network with the exploration of continuous space, the calculation amount and search time can be reduced.

These three algorithms can improve parking efficiency, reduce search time, adapt to traditional standard parking spaces. However, their applications on automatic vertical parking in narrow spaces is rare and their performance is uncertain. In addition, there is no standard for the narrow parking space and no evaluation index for automatic parking in narrow spaces.

For the reasons mentioned above, the contributions of this study are summarized as follows.

## 2. Vehicle Kinematics Model

The following differential equation can be obtained by selecting the center of the equivalent vehicle rear axle as the reference point, referring to Figure 1.

The kinematics equation of the center point of the equivalent rear axle is obtained from equation (1), which can be expressed as

Where  $\theta$  is the heading angle of the vehicle,  $\varphi$  is the Ackermann angle of the vehicle, which is the front wheel steering angle of the vehicle,  $L$  is the wheelbase of the vehicle,  $V$  is the speed at the center point of the rear axle of the vehicle.

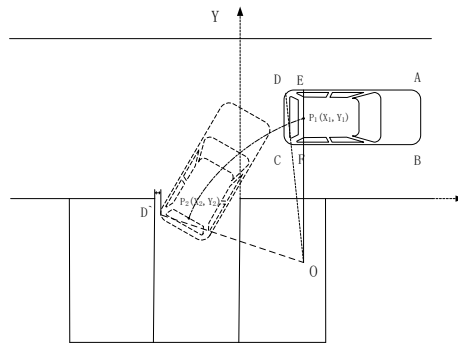


The arcs are obtained by their parameters. The angle parameter  $\varphi_0$  between the straight-line OE and the straight-line OD is obtained from Equation (3) and as shown in Figure 4.

$$\varphi_0 = \arctan\left(\frac{DE}{OE}\right) \quad (3)$$

The parameter corresponding to the first arc  $P_1P_2$  is  $\varphi_1$ , and the vehicle reverses from the initial parking position  $P_0$ . When the vehicle's left rear and left side of the parking space reach the safety threshold, the left parking space vertex coordinates are  $D'$  ( $X_{D'}$ ,  $Y_{D'}$ ). Based on the  $D'$  horizontal coordinate value and the rear axle center coordinate, the angle of the first arc can be obtained.

$$\varphi_1 = \arcsin\left(\frac{X_1 - X_{D'}}{OD}\right) - \varphi_0 \quad (4)$$



**Figure 4.** The first arc path  $P_1P_2$ .

The second arc path is  $P_2P_3$ .

The parameters corresponding to the second arc path  $P_2P_3$  are  $O_2$  and  $\varphi_2$ . The coordinates of the circle's center  $O_2$  ( $x_{o2}$ ,  $y_{o2}$ ) can be obtained from the coordinates of the center  $Q_2$  of the vehicle's rear axle and the vehicle's heading angle. The relationship between the center  $O_2$  and center  $O_3$  can be obtained from the geometric relationship between the tangent of the second and third arcs.

$$(x_{o3} - x_{o2})^2 + (y_{o3} - y_{o2})^2 = (2R_{min})^2 \quad (5)$$

The parameters corresponding to the third arc are  $O_3$  and  $\varphi_3$ . Since the arc  $P_3P_4$  is tangent to the straight line  $P_4P_5$ , the coordinates of  $O_3$  ( $x_{o3}$ ,  $y_{o3}$ ) can be obtained.

Based on the above analysis, the angle  $\varphi_2$  corresponding to the second arc path,  $P_2P_3$  can be obtained.

$$\varphi_3 = \arctan\left(\frac{y_{o2} - y_{o3}}{x_{o3} - x_{o2}}\right) \quad (6)$$

$$\varphi_2 = \frac{\pi}{2} - \varphi_3 - \varphi_1 \quad (7)$$

### 3.3. Principles of Particle Swarm Optimization Planning Algorithm

The particle swarm optimization (PSO) algorithm is a random search algorithm with high precision, fast convergence speed, few parameters, and easy implementation. Through cooperation, the population is optimized. Each potential solution of an optimization problem is considered a bird in the search space, called a "particle" [12]. Each particle has a moderate value determined by an objective function and a speed to determine the flight distance and direction. All particles follow the current optimal particle to search in the solution space, which means finding the optimal solution through multiple iterations.

Assuming that an optimization problem is defined in D-dimensional space, the number of particles in the population is  $S_k$ , and the position, velocity, and cost function values of the particles are  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ,  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ,  $f(X_i)$ , respectively. The optimal population is  $P_i$ , which refers to the particle with the smallest cost function value of the current particle in all iterations.



The global optimal is  $P_g$ , which refers to the particle with the smallest cost function value in the optimal particle.

The velocity and position update formulas of the particles are as follows.

$$v_{id}(k+1) = \eta v_{id}(k) + c_1 r_1 [P_{id} - x_{id}(k)] + c_2 r_2 [P_{gd} - x_{id}(k)] \quad (8)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1), 1 \leq k \leq k_{max}, 1 \leq i \leq N_p, 1 \leq d \leq D \quad (9)$$

$$\eta(k) = \frac{(\eta_{ini} - \eta_{end})(k_{max} - k)}{k_{max}} + \eta_{end} \quad (10)$$

Among them,  $N_p$  is the number of particles, the larger the particle swarm size, the easier it is to find the global optimal solution;  $k_{max}$  is the maximum number of iterations,  $v_{id}(k)$  is the d-dimensional velocity component of the k-th population particle  $i$ ,  $x_{id}(k)$  is the d-dimensional position component of the k-th population particle  $i$ ,  $P_{id}$  is the d-dimensional position component of the optimal particle  $i$ ;  $P_{gd}$  is the global optimal d-dimensional position component,  $r_1$  and  $r_2$  are random functions that generate random numbers between 0 and 1. The  $\eta$  is the inertia weight. Introducing inertia weight gives the particle swarm a more vital global search ability in the early stage and a more vital local optimization ability in the later stage and increasing  $\eta$  can improve global search ability, and reducing  $\eta$  can improve local search ability; reasonable  $\eta$  is the key to achieving efficient search and avoiding falling into local optima. The  $c_1$  and  $c_2$  are acceleration coefficients to determine particle and global optimization guidance weights, where  $c_1$  affects the optimal local value and  $c_2$  affects the optimal global value.

PSO stop criterion: the end of the particle swarm optimization algorithm is either the number of iterations reaches the maximum value or the optimization result reaches the error threshold.

The particle swarm optimization algorithm must find an optimal parking position and angle in vertical parking. Therefore, in the particle swarm optimization algorithm, the position and angle of the vehicle are used as parameters of the objective function; specifically, the position and angle of the vehicle are used as the state vectors of the particles, and the objective function is the path function of arc straight line planning. After determining the independent variable of the minimum value of the objective function and the starting point of parking, the parking point position information is obtained based on the vehicle's minimum turning radius and the objective function's minimum value. The objective function is used to judge its position, and the optimal solution is searched through continuous attempts and adjustments, achieving a fast and efficient vertical parking process.

### 3.4. Principles of Hybrid A\* Planning Algorithm

The Hybrid A\* algorithm is different from the traditional A\* algorithm. It is a state space heuristic algorithm. It can introduce heuristic information in the search process, the search path conforms to vehicle kinematics, the parking environment is gridded, and each search location in the gridded state space is evaluated to get the best position and then perform the following search from this position to the destination [13–15].

The Hybrid A\* algorithm is often used in various path planning, including automatic parking path planning. The Hybrid A\* extends from the parent node with various steering operations (left turn, right turn, or no turn), and combines the vehicle kinematics model to determine new nodes, thus ensuring the motion continuity requirements between the child node and the parent node. Therefore, the path generated in this way can meet the needs of vehicle driving, as shown in Figure 5.

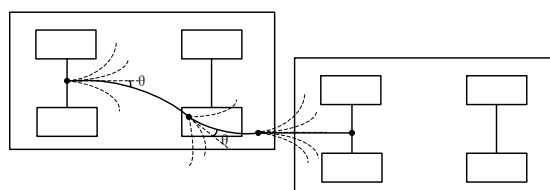


Figure 5. Hybrid A\* extended node diagram.

In path planning, three-dimensional coordinates  $(x, y, \theta)$  to describe the vehicle's posture. In the formula,  $x$  is the horizontal coordinate,  $y$  is the vertical coordinate,  $\theta$  is the heading angle of the vehicle. The node expansion process of the Hybrid A\* algorithm is shown in Figure 6, where  $N_{i-1}$  and  $N_i$  respectively represent the previous node and the current node. The basic principle is shown in Equations (11)–(15).

$$\beta = \left( \frac{L_{\text{arc}}}{W} \right) \times \tan \alpha \quad (11)$$

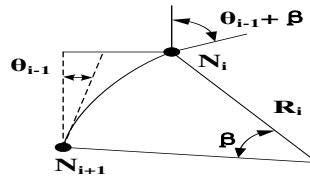
$$R_i = \frac{L_{\text{arc}}}{\beta} \quad (12)$$

$$x_i = x_{i+1} + R_i [\sin(\theta_{i+1} + \beta) - \sin(\theta_{i-1})] \quad (13)$$

$$y_i = x_{i+1} - R_i [\cos(\theta_{i+1} + \beta) + \cos(\theta_{i-1})] \quad (14)$$

$$\theta_i = \theta_{i-1} + \beta \quad (15)$$

Where,  $L_{\text{arc}}$  is the step size of each extension of the Hybrid A\* algorithm,  $\alpha$  is the steering angle of the vehicle,  $W$  is the track width, the angle of  $\beta$  corresponding to the arc between  $N_{i-1}$  and  $N_i$  nodes can also be directly set as the minimum turning radius of the vehicle during expansion.



**Figure 6.** New nodes generation principle of Hybrid A\*.

The Hybrid A\* algorithm considers the influence of the front wheel angle on the node expansion to form a three-dimensional coordinate  $(x, y, \theta)$ . At the same time, two heuristic functions are considered: unconstrained heuristic function and constrained heuristic function. Among them, the unconstrained heuristic function ignores the non-holonomic constraints of the vehicle, considers the environmental obstacles. Therefore, it is not necessary to consider the kinematic constraints of the vehicle, such as the heading angle. The constrained heuristic function ignores environmental obstacles, considers vehicle non-integrity constraints. The unconstrained heuristic function value is used to calculate the constrained heuristic function value, and the Reeds-Shepp curve is used to calculate the size of the heuristic function value.

### 3.5. Parking Path Evaluation function

Planned path length is commonly used as index to measure the performance of standard spaces parking. However, it is not suitable for evaluating the narrow spaces parking because of the frequent changes in speed and direction. In this paper, we propose a multi-objective evaluation function for narrow parking space paths to evaluate the comprehensive performance of parking algorithms.

1. The number of positive and negative transitions in vehicle speed. The number of positive and negative vehicle speed transitions  $R$  is obtained by judging the speed direction of the previous and subsequent moments.
2. The total length of the path.

$$L = \sum_{i=1}^N \left( (x(i+1) - x(i))^2 + (y(i+1) - y(i))^2 \right)^{\frac{1}{2}} \quad (16)$$

3. Path Smoothness.

The minimum curvature of the path curve can represent the smoothness of the path curve. The smaller the minimum curvature, the better the smoothness of the path curve. The average minimum curvature of the entire parking path curve can be compared.



$$\bar{\rho} = \min(\rho_{i1min}, \rho_{i2min}, \rho_{i3min}) \quad (17)$$

where  $\rho_{i1min}$  is the minimum curvature of the first path curve of the i-th path point;  $\rho_{i2min}$  is the minimum curvature of the second segment of the curve at the i-th path point;  $\rho_{i3min}$  is the minimum curvature of the third segment of the curve at the i-th path point.

The above three evaluation indicators values are calculated separately and weighting summation. The multi-objective evaluation function value  $S$  of the planning route of these three planning algorithms is obtained as follows,

$$S = \omega_l L + \omega_r R + \omega_p \bar{\rho} \quad (18)$$

where  $\omega_l$  is the weighted coefficient of the total path length;  $\omega_r$  is the weighted coefficient of the number of positive and negative conversions of vehicle speed;  $\omega_p$  is the weighted coefficient of the planned path curvature.

## 4. Path Planning Simulation

### 4.1. Parking Scene Settings

According to the "Specification for the Setting of Parking Spaces on Urban Roads," (GA/T 850—2021 of China) the size of vertical parking spaces for small cars is 2.5-2.7 \* the size is 5-6 meters. The width of the single-lane turning lane is not less than 3.5 meters, while the width of the double lane is not less than 5 meters. The turning section should meet the needs of one vehicle turning at a time. This paper selects a small sedan as the research object, with a width of 1.737m, a length of 4.579m, and a minimum turning radius of 5.6m, and the safety threshold between vehicle and parking space  $\psi$ ,  $\psi = 0.1m$ . According to the above definition of narrow parking spaces, this paper sets the parameters for narrow parking spaces, as shown in Table 1.

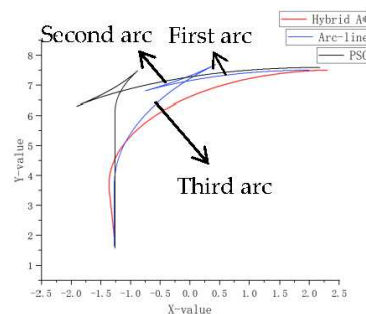
**Table 1.** Parameters of narrow parking scenarios.

Scene parameters	Parking space width Sw/m	Parking space length Sl/m	Length of driving passage L/m	Width of driving passage W/m	Vehicle's Initial heading angle $\theta_{min}/(^{\circ})$
Value	2.3~2.1	5.0	9.0	5.0	-10~10

### 4.2. Simulation

In this paper, the path planning algorithm is simulated in MATLAB. Firstly, based on the three established planning models, the arc straight line, the Hybrid A\*, and the particle swarm optimization paths for a 2.5m wide standard parking space are simulated at a given starting point. Then, the path planning simulation for narrow parking spaces is compared, and the curvature of each arc of the three is compared.

The following is a comparison of the paths planned by the three algorithms when the parking space width is a standard size of 2.5m, as shown in Figure 7.



**Figure 7.** Paths of the three algorithms.

It can be seen that the path planned by the Hybrid A\* includes two arcs and a straight line, and the path planned by arc-line combination and particle swarm optimization includes three arcs and a straight line. Moreover, the first arc of the Hybrid A\* planning is much smaller than the arc-line combination and particle swarm optimization. In this way, to complete the parking operation, the path of the Hybrid A\* planning needs to be adjusted once outside the library to enter the library, while the arc straight line and the particle swarm need to be adjusted twice outside the library. The following will simulate and compare the width of different parking spaces.

The path curve can be discretized into many path points containing  $(x,y)$  coordinate values. Then, according to the path length calculation formula in the parking path evaluation index in Section 2.3, the arc-line combination, Hybrid A\*, and particle swarm optimization path length can be calculated. Based on each coordinate point, the curvature calculation formula can be used to obtain the curvature value corresponding to each arc segment.

4.2.1. Arc-Line Combination

The path length and curvature for arc-line combination planning are simulated based on different parking space widths in MATLAB simulation, as shown below.

The above is the curvature diagram of the simulated path using the arc line planning algorithm in three narrow parking spaces with different widths. Although the parking spaces have different widths, the curvature mutation is the same. From Figure 8, the curvature of the first arc has two mutation points, the second arc has two mutation points, and the third arc has two mutation points. The comparison information of path length and reverse times simulated by MATLAB under different parking space widths can be seen in Table 2.

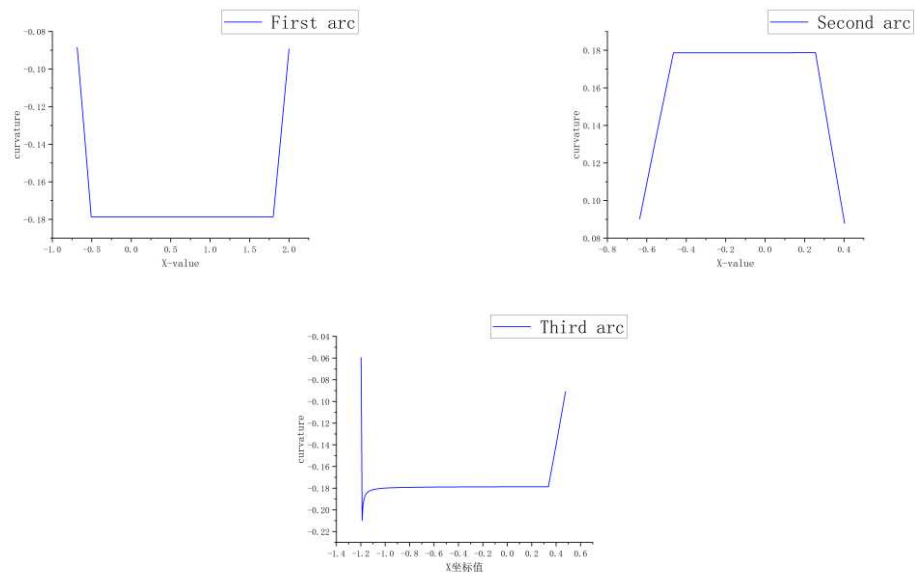
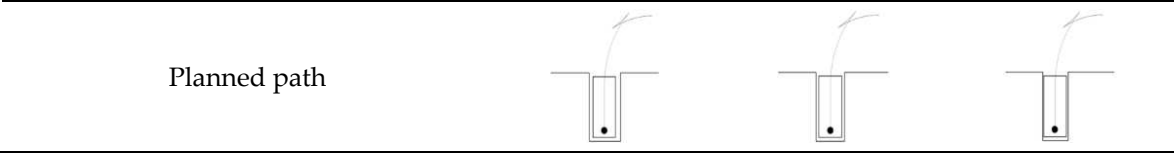


Figure 8. Three-segments arc path curvatures of arc-line combination algorithm.

Table 2. Planned paths of arc-line combination algorithm under different parking spaces.

Parking Space width/m	2.3	2.2	2.1
Total path length/m	11.39	11.37	11.21
Speed positive and negative conversion times/time	2	2	2
Security threshold/m	0.2	0.1	0.09



4.2.2. Hybrid A\*

The path length and curvature of Hybrid A\* planning is based on different parking space widths in MATLAB simulation, as shown below.

The above is the curvature map of the simulated path using the Hybrid A\* planning algorithm in three narrow parking spaces with different widths. Although the parking spaces have different widths, the curvature mutation is the same. From Figure 9, the curvature of the first arc has two mutation points, the second arc has two mutation points, and the third arc has two mutation points. The comparison information of path length and reverse times simulated by MATLAB under different parking space widths can be seen in Table 3.

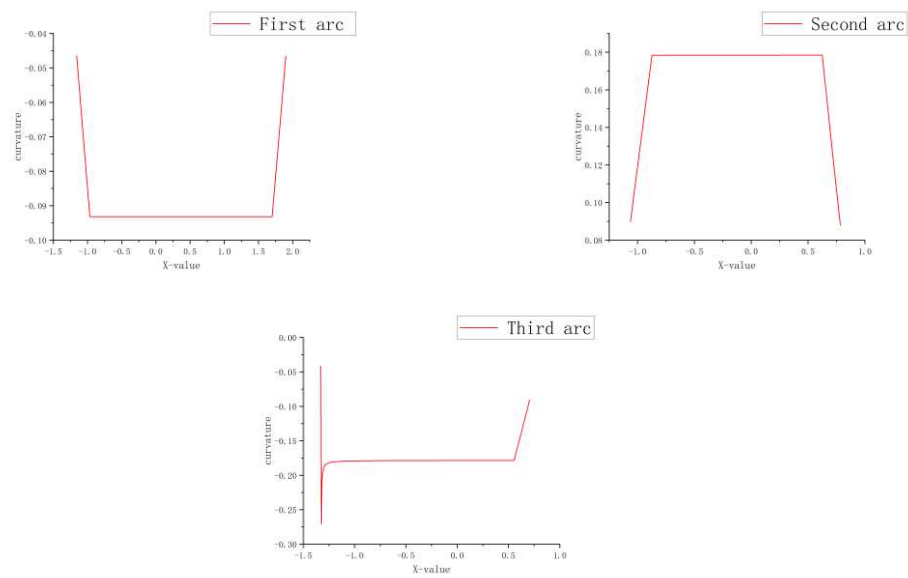
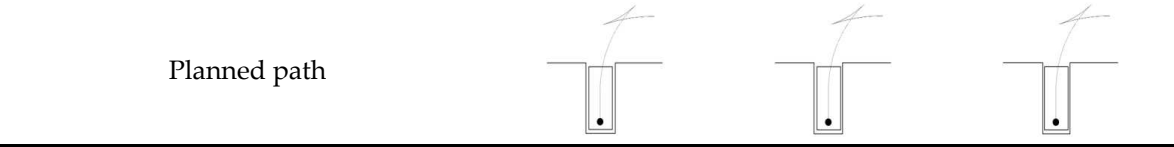


Figure 9. Three-segments arc path curvature of Hybrid A\* algorithm.

Table 3. Planned paths of Hybrid A\* algorithm under different parking spaces.

Parking Space width/m	2.3	2.2	2.1
Total path length/m	9.48	9.50	9.53
Speed positive and negative conversion times/time	2	2	2
Security threshold/m	0.2	0.1	0.09

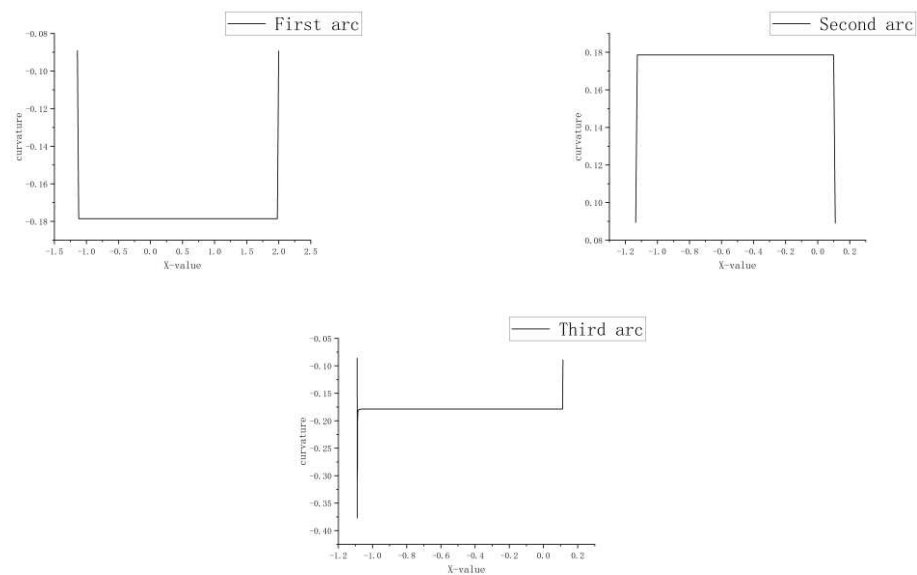


4.2.3. Particle Swarm Optimization

The particle swarm parameters are set reasonably with reference to practical engineering experience. The path length and curvature of particle swarm optimization are based on different parking space widths in MATLAB simulation, as shown below.

The above is the curvature diagram of the simulation path using the particle swarm optimization algorithm in narrow parking spaces. From Figure 10, the curvature of the first arc has two mutation

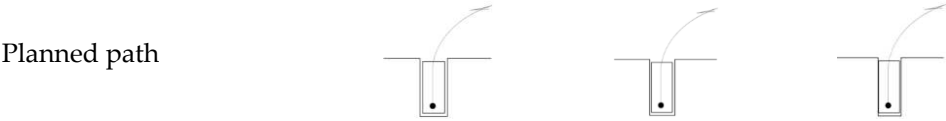
points, the second arc has two mutation points, and the third arc has two mutation points. The path curvature map planned by the particle swarm optimization algorithm is very similar to the arc line planning algorithm, but the curvature of the segmented arc in Figure 10 changes relatively faster than that of the arc line, and the curvature is less smooth; the second mutation in the third arc is relatively more severe than the mutation in the straight arc. The comparison information of path length and reverse times simulated by MATLAB under different parking space widths can be seen in Table 4.



**Figure 10.** Three-segments arc path curvature of particle swarm optimization algorithm.

**Table 4.** Planned paths of particle swarm optimization algorithm under different parking spaces.

Parking space width/m	2.3	2.2	2.1
Total path length/m	9.47	9.48	9.50
Speed positive and negative conversion times/time	2	2	2
Security threshold/m	0.1	0.09	0.08



Through the above simulation results, it can be found that the path needs to be adjusted multiple times to park smoothly due to the narrow space, whether using circle arc- straight line, Hybrid A\*, or particle swarm optimization algorithms. From the three tables, the path planned by circular arc straight lines is the longest, and the three planning algorithms have the same number of positive and negative vehicle speed conversions, that is, reverse times. By analyzing the path length, the number of positive and negative vehicle speed transitions, and the number of vehicle position adjustments outside the depot in Figure 7, the parking operation time of the path planned by the geometric algorithm is relatively long, and the efficiency is relatively low.

By analyzing the paths in Figure 7 and the tables, it can be found that the Y values of the adjusted coordinates for the arc straight line and particle swarm planning paths outside the garage are relatively large, which means that the vehicle’s position is farther away from the garage compared to Hybrid A\*, and the width of the driving lane is 5m. In this way, further away from the garage increases the risk of vehicles touching the curb, opposite vehicles, or walls, leading to parking failure.

The simulation results of selecting different narrow-width parking spaces show that only the Hybrid A\* algorithm successfully parks when the parking space width is 2.1m, the arc-line

combination and Hybrid A\* algorithm parking successfully at 2.2m. All three algorithms were successfully parked at 2.3m. This indicates that Hybrid A\* is most suitable when the parking environment is relatively narrow. Selecting a width of 2.3m parking space, the weighting coefficients adopted in this article are  $\omega_l=1.25$ ,  $\omega_r=1$ ,  $\omega_p=400$ . Based on the multi-objective evaluation function, the results of three parking planning algorithms are compared, as shown in Table 5.

**Table 5.** Performance of the three algorithms.

Algorithm	Total Path Length/m	Speed Positive and Negative Conversion Times/Time	$\bar{\rho}$	S
Arc-line combination	11.39	2	0.159	79.84
Hybrid A*	9.48	2	0.151	74.25
PSO	9.47	2	0.161	78.24

Based on comprehensive analysis, the arc-line combination planning has the longest path among the three evaluation indicators, and the particle swarm optimization has the shortest path; The number of positive and negative vehicle speed transitions is the same. The average minimum curvature of the three arcs of Hybrid A\* is the smallest, and the particle swarm is the largest. The final evaluation function S has the lowest score of Hybrid A\*, while the arc straight line has the highest score and significantly differs from the two other planning algorithms. This indicates that the arc straight line planning algorithm is unsuitable for narrow parking spaces. Although the particle swarm optimization algorithm has the same evaluation score as Hybrid A\* in a 2.3m vast parking space and the shortest path, the path is not smooth relative to Hybrid A\*, and the curvature mutation is more significant. Therefore, the Hybrid A\* planning algorithm is more suitable for narrow parking spaces.

Although the Hybrid A\* planning algorithm is better overall, it also has the problem that the curvature discontinuity of the path curve leads to the spot turn. In order to solve the problem of curvature discontinuity of the planned path, it is necessary to smooth the path optimization.

## 5. Path Smoothing Optimization of the B-spline Curve and Gradient Descent Algorithm

The Gradient descent is a numerical optimization algorithm with a small storage capacity and high stability. The gradient of the function (the tangent slope of the function at this point) can well indicate the direction in which the function value drops the fastest so that each step can effectively approach the optimal solution. It can optimize the curvature of the smoother curve and adjust the curvature of the curve to meet the kinematic requirements of vehicle driving. If the curve is not smooth in some places, there may be no gradient in these places, which makes the gradient descent algorithm unable to optimize in these places.

A series of control points, including the starting point and the end point of the curve, are selected. The B-spline curve, with multiple control points, has the characteristics of multi-order derivative continuity. Therefore, the B-spline curve can be used for path smoothing, which makes up for the shortcomings of the gradient descent algorithm that cannot be optimized in the non-smooth path. However, the parking paths are composed of multi-segment arcs and straight lines for narrow parking spaces. In this way, B-spline curves are used to optimize each arc of multi-step parking, and then the optimized smooth curve is used to limit the path curvature by the gradient descent algorithm.

### 5.1. Path Smoothing Optimization Based on the B-Spline Curve

The B-spline curve is an improved curve based on the Bézier curve [16]. It can not only change the order of the curve but also realize the local change of the curve. Assuming that there are  $n$  control vertices, the  $k$ -order B-spline curve function is expressed as follows.

$$P(u) = \sum_{i=0}^n P_i N_{i,k}(u) \quad (19)$$

In the above formula,  $k = \{0, 1, 2, \dots, n\}$ ,  $U = \{u_0, u_1, \dots, u_{(n+k+1)}\}$  is the node vector sequence,  $P_i$  is the  $i$ th curve equation corresponding to this point, and  $N_{(i,k)}(u)$  is the  $k$ -th B-spline basis function. The recurrence formula is as follows.

$$\begin{cases} \text{define } \frac{0}{0} = 0 \\ N_{i,0}(u) = \begin{cases} 1, & u_i \leq u < u_{i+1} \\ 0, & \text{other} \end{cases} \\ N_{i,k}(u) = \frac{(u-u_i)N_{i,k-1}(u)}{u_{i+k}-u_i} + \frac{(u_{i+k+1}-u)N_{i+1,k-1}(u)}{u_{i+k+1}-u_{i+1}}, & u_k \leq u < u_{n+1} \end{cases} \quad (20)$$

According to the structure of the B-spline curve, its shape is determined by the node vector and the control point of the piecewise mixed function. Then, according to the distribution of the midpoint of the node vector sequence  $U$ , the B-spline curve can be divided into four types [17]: uniform B-spline curve, quasi-uniform B-spline curve, piecewise Bézier curve, non-uniform B-spline curve.

When optimizing the parking path, the starting point and the endpoint of the parking cannot be ignored. The quasi-uniform B-spline curve meets the requirements of passing through the starting and endpoints. It is closer to the original trajectory, which can maintain the shape of the original path to the maximum extent. Therefore, this paper chooses this curve to optimize the parking path. When the value of  $k$  in Equation (21) is 3, the expression of cubic quasi-uniform B-spline can be obtained.

$$P(u) = \sum_{i=0}^3 P_i N_{i,3}(u) \quad (21)$$

Then, this paper uses the B-spline curve to fit each path arc to obtain a smooth path. The specific algorithm process is as follows: first, the path planning is carried out to obtain the path point set, and the opposite change position of the vehicle movement direction is judged; more, multiple arc path curves are obtained by segmenting the complete path, and B-spline curves are used to optimize each path curve. Finally, the final smooth path is obtained by stitching the optimized path curves.

### 5.2. Path Curvature Optimization Based on the Gradient Descent

To ensure that the curve curvature satisfies the minimum turning radius constraint of the vehicle for the optimized path  $X_i = (x_i, y_i)$ ,  $i \in [1, N]$  of the B-spline curve, the following objective function is designed for re-optimization, and the planned path curvature is controlled to be not more significant than the maximum turning curvature of the vehicle.

$$F = \omega_c \sum_{i=1}^{N-1} \sigma_c \left( \frac{\Delta \phi_i}{|\Delta X_i|} - k_{max} \right) \quad (22)$$

Where  $\Delta \phi_i$  represents the change of the tangential angle at the vertex;  $\omega_c$  represents the weight of the curvature term;  $\sigma_c$  is the penalty function,  $\Delta X_i = X_i - X_{i-1}$ , is the displacement at point  $X_i$ ;  $\Delta \phi_i$  is the angle change of the path at the point  $X_i$ ;  $k_{max}$  is the maximum curvature of the vehicle turning; among them:

$$\Delta \phi_i = \left| \tan^{-1} \left( \frac{\Delta y_{i+1}}{\Delta x_{i+1}} \right) + \tan^{-1} \left( \frac{\Delta y_i}{\Delta x_i} \right) \right| \quad (23)$$

Because the curvature is affected by the derivative of the three points  $X_i$ ,  $X_{i+1}$ ,  $X_{i-1}$ , and the curvature  $k_i = \frac{\Delta \phi_i}{|\Delta X_i|}$ , therefore:

$$\Delta \phi_i = \cos^{-1} \left( \frac{\Delta X_i^T \Delta X_{i+1}}{|\Delta X_i| |\Delta X_{i+1}|} \right) \quad (24)$$

The partial derivatives of curvature to  $X_i$ ,  $X_{i+1}$ ,  $X_{i-1}$  are calculated respectively, and the following results are obtained:



$$\frac{\partial k_i}{\partial X_i} = -\frac{1}{|\Delta X_i|} \frac{\partial \Delta \phi_i}{\partial \cos(\Delta \phi_i)} \frac{\partial \cos(\Delta \phi_i)}{\partial X_i} - \frac{\partial \Delta \phi_i}{\Delta X_i^2} \frac{\partial \Delta X_i}{\partial X_i} \quad (25)$$

$$\frac{\partial k_i}{\partial X_{i+1}} = -\frac{1}{|\Delta X_i|} \frac{\partial \Delta \phi_i}{\partial \cos(\Delta \phi_i)} \frac{\partial \cos(\Delta \phi_i)}{\partial X_{i+1}} \quad (26)$$

$$\frac{\partial k_i}{\partial X_{i-1}} = -\frac{1}{|\Delta X_i|} \frac{\partial \Delta \phi_i}{\partial \cos(\Delta \phi_i)} \frac{\partial \cos(\Delta \phi_i)}{\partial X_{i-1}} - \frac{\partial \Delta \phi_i}{\Delta X_i^2} \frac{\partial \Delta X_i}{\partial X_{i-1}} \quad (27)$$

The gradient formula of curvature is:

$$\frac{\partial F}{\partial X_i} = \omega_c \left( \omega_{i-1} \frac{\partial k_i}{\partial X_{i-1}} + \omega_i \frac{\partial k_i}{\partial X_i} + \omega_{i+1} \frac{\partial k_i}{\partial X_{i+1}} \right) \quad (28)$$

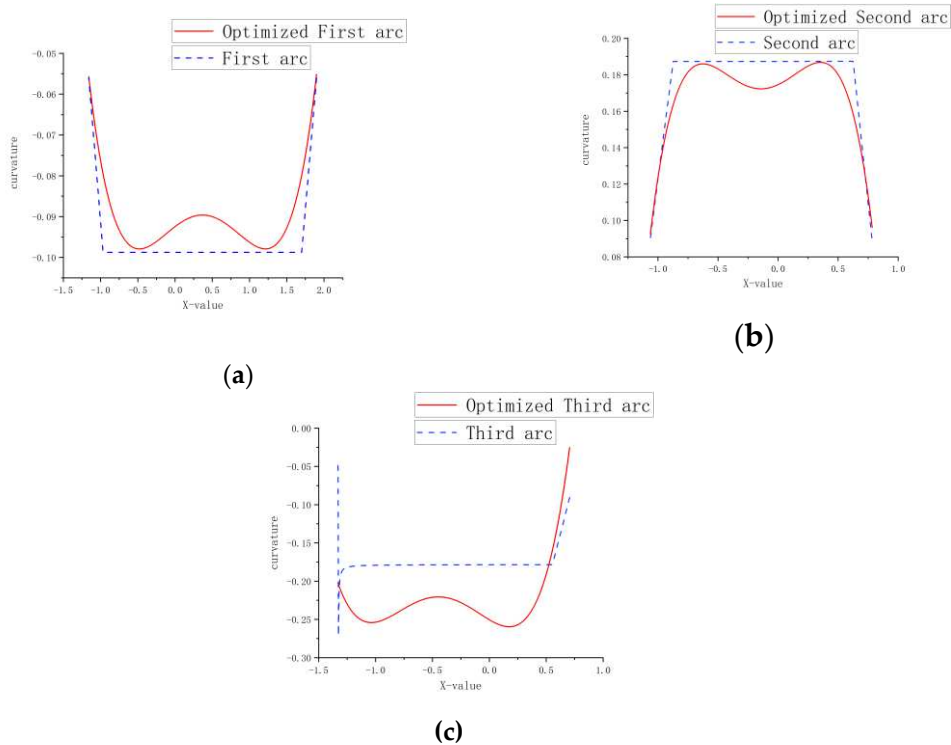
For formulas (25)–(27), according to the curvature gradient formula can be obtained:

$$\frac{\partial \Delta \phi_i}{\partial \cos} = \frac{\partial \cos^{-1}(\cos(\Delta \phi_i))}{\partial \cos(\Delta \phi_i)} = \frac{-1}{(1 - \cos^2(\Delta \phi_i))^{1/2}} \quad (29)$$

In summary, the curvature of each path point is calculated according to the current path sequence and input as a curvature penalty term, and the weighted sum is used as the overall objective function. In each iteration, the coordinate correction value of each point on the path is obtained by gradient derivation of the objective function. Then the coordinate value of each point on the original path is added with the correction value to obtain a new path optimized by one gradient descent. The algorithm will converge to the new path within the set total number of iterations M.

According to the same path information of Hybrid A\* in Section 3.2, and MATLAB simulation, the following figure is the curvature comparison after Hybrid A\* path optimization.

Figure 11a is the comparison of the curvature of the first arc parking path of the Hybrid A\* before and after the B-spline curve and the gradient descent algorithm; Figure 11b is the comparison of the curvature of the second arc parking path of the Hybrid A\* before and after the B-spline curve and the gradient descent algorithm. Figure 11c compares the curvature of the third arc parking path of the Hybrid A\* before and after the B-spline curve and the gradient descent algorithm. The mean curvature of Figure 11a,b becomes smaller. Although the curvature in Figure 11c becomes more significant, the curvature of the three path curves is continuous without mutation and meets the requirements of continuous curvature change of parking path planning and the minimum turning radius constraint of the vehicle.



**Figure 11.** Three arc path curvatures of optimized Hybrid A\*

The Hybrid A\* path curve with a width of 2.3m and the optimized path curve of 3.2 parking spaces are selected. The multi-objective evaluation function compares the multi-objective function S values before and after optimization. The simulation results can be seen in Table 6.

**Table 6.** Performance comparison between optimized Hybrid A\* and the other.

Algorithm	Total Path Length/m	Speed Positive and Negative Conversion Times/Time	$\bar{\rho}_t$	S
Hybrid A*(without optimization)	9.48	2	0.151	74.25
Hybrid A*(optimization)	11.02	2	0.146	74.18
PSO	9.47	2	0.161	78.24
Circular arc straight line	11.39	2	0.159	79.84

Based on comprehensive analysis, the curvature of the Hybrid A\* path is optimized by the B-spline curve and the gradient descent algorithm. The average value of the minimum curvature is less than the curvature before optimization, which is reduced by  $0.005 \text{ m}^{-1}$ , and less than the curvature of the arc straight line and the particle swarm optimization algorithm. Although the Hybrid A\* increases the path length by 1.54 m after optimization, the function value after optimization is less than before optimization, which is reduced by 0.07. The simulation results show that in the narrow parking environment, the path of the Hybrid A\* planning optimized by the B-spline curve and the gradient descent algorithm proposed in this paper is better overall.

## 6. Conclusions

This paper defines a narrow parking space. The single-step parking failure of the arc-line combination parking algorithm is taken as the standard. The parking space is defined as a narrow parking space, and the width of the narrow parking space is approximately 1.25-1.35 times the width of the vehicle body, namely,  $W_{p1} \approx 1.35W_p$  and  $W_{p2} \approx 1.25W_p$ . And, aiming at the narrow parking space and narrow parking environment, a multi-objective function is proposed to evaluate the applicability of three planning algorithms, namely arc straight line, hybrid A\* and particle swarm optimization, in narrow parking space.

The curvature of the parking path curve is discontinuous, and severe mutations result in spot turns according to the simulation results and the analysis of narrow parking environments. This paper proposes a algorithm based on the B-spline curve and gradient descent algorithm to optimize the path to solve the problem of discontinuous and abrupt curvature of the path curve. The optimized path curvature is smooth and satisfies the minimum turning radius constraint of the vehicle.

However, this paper does not consider the speed planning of parking, parking space detection and recognition and parking path following control. In the future, the variable speed planning and the recognition of narrow parking spaces by vehicle and the following control of vehicle will be considered.

## References

1. Xie, S.; Chen, H.; Zhang, J. Path Planning Algorithm for Parallel Parking. Proceedings of 2014 China Society of Automotive Engineering Annual Meeting, Beijing, China, 16-04-2014, pp.637-640.

2. Zhao, B.; Xin, C.; Man, J.; Liang, C.; Jing, J. A novel Path Planning Methodology for Automated Valet Parking Based on Directional Graph Search and Geometry Curve. *Robot Auton Syst*, 2020, 132, 103606.
3. Jiang, M.; Peng, Y.; Huang, W.; Xu, D. Polynomial Curve-optimized Vertical Parking Path Planning and Tracking. *J F Univ (Nat Sci Ed)*, 2023, 51(1), 76-82.
4. Zhang, J.; Zhao, J.; Shi, Z.; Yang, X. Trajectory Planning and Tracking Control for Perpendicular Parking Based on Clothoid Curve. *J S Univ (Nat Sci Ed)*, 2020, 50(1), 182-191.
5. Yu, L.; Wang, X.; Wu, B.; Hou, Z.; Wu Y. Path Planning Optimization and Tracking of Parallel Parking for Driverless Vehicles. *J J Univ (Nat Sci Ed)*, 2022, 43(5), 519-523.
6. Kim, M.; Ahn, J.; & Park, J. Target Tree-RRT\*. Continuous-Curvature Path Planning Algorithm for Autonomous Parking in Complex Environments. *IEEE T Autom Sci Eng*, 2022, 1-12.
7. Zhang, J.; Bu C.; Wang C.; Zhao J. Parallel Parking Path Planning and Tracking Control for Wire-four-wheel Steering Vehicle. *J H Univ (Nat Sci Ed)*, 2021, 48(8), 44-50.
8. Sedighi, S.; Nguyen, D.-V.; Kuhnert, K.-D. Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications. 2019 5th International Conference on Control, Automation and Robotics, Beijing, China, pp.19-04-2019.
9. Bai, J.; Xu, Y.; Ji, J.; Chen, Y.; Zhang, Y. Path Planning for Automated Valet Parking Based on Graph Search and Geometry Curve. *J C Univ of Technol (Nat Sci Ed)*, 2022, 36(6), 115-125.
10. Xiong, L.; Gao, J.; Fu, Z.; Xiao, K. Path Planning for Automatic Parking Based on Improved Hybrid A Algorithm. 2021 5th CAA International Conference on Vehicular Control and Intelligence, Tianjin, China, pp.29-10-2021.
11. Hu, J.; Zhu, L.; Chen, R.; Zhong, X.; Xu, W.; Zhang, M. Research on Parallel Parking Path Planning Algorithm for Narrow Parking Space. *AE*, 2022, 44(7), 1041-1048.
12. Van Nguyen.; L., Phung.; M. D.; Ha, Q. P. Game Theory-Based Optimal Cooperative Path Planning for Multiple UAVs. *IEEE Access*, 2022, 10, 108034-108045, [6].
13. Ren, B.; Wang, X.; Deng, W.; Nan, J.; Zhong, R.; Ding J. Path Optimization Algorithm for Automatic Parking Based on Hybrid A\* and Reeds-Shepp Curve with Variable Radius. *China J. Highw. Transp*, 2022, 35(7), 317-327.
14. Bi, J.; Kaoru, H.; Xiang, D.; Ya, P.; Zhi, Y. Path Planning Based on Improved Hybrid A\* Algorithm. *J Adv Comput Intell*. 2021, 25(1), 64-72.
15. Huang, J.; Liu, Z.; Xue, M.; Feng, H.; Hong, Y. Search-Based Path Planning Algorithm for Autonomous Parking: Multi-Heuristic Hybrid A\*. 2022 34th Chinese Control and Decision Conference, Hefei, China, pp. 15-8-2022.
16. Zhao, L.; Mao, R.; Bai, Y.; Local Path Planning for Unmanned Surface Vehicles based on Hybrid A\* and B-spline. 2022 IEEE International Conference on Unmanned Systems, Guangzhou, China, pp.28-10-2022.
17. Zeng, D.; Yu, Z.; Zhang, P.; Chen, H. Cubic B-spline Curve for Obstacle Avoidance Trajectory Planning of Unmanned Vehicle. *J T Univ (Nat Sci Ed)*, 2019, 47(z1), 159-163.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.