J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 6415--6428

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Path Search Algorithm for Parking Spaces Based on Artificial Intelligence Algorithm

Yuan Sun[1, ✉]

[1] *School of Mathematics and Information Engineering, Puyang Vocational and Technical College, Puyang, Henan, 457000, China*

ABSTRACT

In order to alleviate the problems of short supply of parking spaces and traffic congestion, intelligent driving solutions have emerged. Automatic parking has now become the first application scenario for driverless driving due to the more fixed scenario and lower traveling speed. In this study, the traditional A* algorithm is improved using the cost function, and the hybrid algorithm of parking space path search and planning is designed by combining the improved A* algorithm with the Reeds-Shepp curve, and then combined with the collision constraints to improve the algorithm's path planning performance. The results of simulation experiments and in-loop test experiments show that the maximum lateral error and heading error are low in parallel and perpendicular parking scenarios, and it is found that the average lateral error during the whole parking process is only 0.177m in the in-loop test, which is a good tracking effect for vehicles. The path search and planning algorithm designed in this paper can better realize the autonomous parking function and has high tracking accuracy and stability in the simulation scenario.

Keywords: A* algorithm, Reeds-Shepp curve, cost function, constraints, path search planning

1. Introduction

With the continuous improvement of people's living standard, every household starts to own a car. With the proliferation of automobile ownership, there is always a large amount of ineffective cruising process when parkers are looking for a vacant parking space, which consumes a lot of time and money at the same time [1-3]. Thus, reducing the cruising process of parkers and ensuring efficient parking is an important problem that needs to be solved worldwide [4].

There are two main solutions to the urban parking problem, one is to carry out intelligent upgrading of parking lots to enhance the ability of space guidance in the parking lot and improve the quality of parking service, and the other is to expand the scale of the original parking lot [5-6]. Due to the limited

✉ Corresponding author.
E-mail address: S13623936303@163.com (Y. Sun).
Received 20 March 2024; Revised 05 June 2024; Accepted 20 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-355

nature of urban land resources, expanding the scale of parking lots can only be carried out slowly, and subject to the strict constraints of the environmental conditions surrounding the parking lot, urban parking will continue to face the dilemma of "too much gruel, too little porridge", in addition to the simple expansion of the scale of the parking lot will also reduce the operational efficiency of the parking lot [7-10]. Therefore, how to carry out intelligent upgrading of the existing parking lot to meet the growing demand for parking in the city has become one of the most effective means to alleviate the urban parking problems.

With the mature application of emerging technologies, especially modern network technologies in the field of parking guidance, the parking guidance system has been upgraded to keep pace with the times [11-12]. The application of artificial intelligence technology as well as mobile Internet technology to change the release of parking information in the guidance system, parkers can use mobile terminals such as cell phones to access the parking guidance system through the Internet, and access to parking information is no longer limited to the parking guidance screen on the road [13-16]. In addition, artificial intelligence systems can also be introduced to change the one-way data release in the previous parking guidance system into a two-way interactive way, after the driver accesses the system, he can not only receive the guidance information released by the system, but also enter some predetermined instructions into the system, which in turn introduces the functions such as parking space booking, online payment, and staggered parking, etc., on the basis of the original functions of the system [17-21]. Therefore, the effective use of artificial intelligence technology will change the sense style of state information in the traditional parking lot, and is of great significance to enhance the operational efficiency and service level of the existing parking lot.

In the planning of parking paths, Liu, J. used IoT technology to model major parking lots around roads and combined particle swarm optimization algorithm and wavelet neural network model to calculate the optimal path for parkers to reach the parking lot with the constraints of the shortest path and the optimal time [22]. Shin, J. H. et al. embedded a predictive control method of neural network into an intelligent parking guidance system, which effectively solved the parking problem by evaluating multiple factors in real time in order to dynamically control vehicles to select the best parking lot [23]. Zhang, X. et al. developed a parking guidance system integrating wireless parking detectors, path optimization units and other modules that can guide parkers to the nearest parking lot and support personalized travel services for users [24]. Zhu, H. introduced a smart parking guidance system that provides real-time free parking information, which can quickly guide parkers to the target parking lot through real-time monitoring of parking space usage [25]. Chai, H. et al. developed a dynamic parking guidance system with the goal of minimizing travel costs, which supports drivers to switch between destinations and traffic paths while driving again, and plays an important role in coordinating the management of parking resources [26]. Zou, W. et al. investigated the application of data mining technology in urban parking guidance system, which can reconstruct citywide parking data at low cost by mining public information of urban parking lots to quantify their service capacity, providing data support for the development of urban parking guidance system [27]. Gao, F. et al. proposed an intelligent urban parking induction method based on ant colony algorithm to model optimal parking selection from multiple perspectives to help parkers make intelligent decisions regarding urban parking with certain effectiveness [28]. It can be found that more research on how to guide the parker to the entrance of the parking lot, in the parking lot still only rely on the parking guidance screen to provide information on available parking spaces, the parker therefore can not directly access to the information on available parking spaces, the need to stay in the parking lot on the main road for a long time to cruise to find available parking spaces. Therefore, the study of parking space path search algorithms based on artificial intelligence algorithms can greatly reduce the time and economic costs of parkers, and reduce the possibility of congestion in the aisle, and realize accurate parking space guidance.

In this paper, based on the kinematic constraints of the front and rear axles of the vehicle in the parking process and the factors such as the steering radius and deflection angle of the vehicle tires, a kinematic model of the vehicle in the parking process is proposed, according to which the trajectory of the vehicle at an arbitrary point in the parking process is deduced. Subsequently, the maximum corner constraint method and Reeds-Shepp curve are used to optimize and improve the node expansion method and cost function setting of the traditional A* algorithm to reduce the complexity of the parking space path solving. A hybrid algorithm combining the optimized A* algorithm and Reeds-Shepp curve is proposed, and this algorithm searches and plans the best parking space path. The parking collision constraints and cost function constraints are then utilized to improve the algorithm's ability to search and plan parking space paths. Finally, a parking simulation platform is designed based on CarSim software to verify the accuracy of the path search and planning algorithms through vertical and parallel parking simulation experiments. And design the automatic parking in the loop test scene to explore the effectiveness of the algorithm under the complex parking scene.

2. Vehicle kinematics model construction

In the inertial coordinate system OXY, (X_r, Y_r) and (X_f, Y_f) are the coordinates of the center points of the front and rear axles of the vehicle, φ is the vehicle swing angle (heading angle), α is the front wheel deflection angle, v_r is the velocity at the center point of the rear axle of the vehicle, v_f is the velocity at the center point of the front axle of the vehicle, and L is the wheelbase of the vehicle. R represents the radius of the vehicle's rear wheel steering, P represents the vehicle's instantaneous center of rotation, M represents the center of the vehicle's rear axle, and N represents the center of the vehicle's front axle.

Because the vehicle in the process of low-speed movement, the vehicle generally does not occur side-slip phenomenon, so the assumption of automatic parking tires without side-slip, the vehicle front, the rear axle center point of the lateral velocity is zero, so the kinematic constraints of the front, the rear axle is:

$$\begin{cases} \dot{X}_f \sin(\varphi+\alpha) - \dot{Y}_f \cos(\varphi+\alpha) = 0 \\ X_r \sin\varphi - \dot{Y}_r \cos\varphi = 0 \end{cases} \tag{1}$$

At point (X_r, Y_r) in the center of the rear axle shaft of the vehicle, the speed is:

$$v_r = \dot{X}_r \cos\varphi + \dot{Y}_r \sin\varphi \tag{2}$$

Can be obtained jointly from Eq. (3) and Eq. (4):

$$\begin{cases} \dot{X}_r = v_r \cos\varphi \\ \dot{Y}_r = v_r \sin\varphi \end{cases} \tag{3}$$

This can be obtained from the geometry of the front and rear wheels:

$$\begin{cases} X_f = X_r + L\cos\varphi \\ Y_f = Y_r + L\sin\varphi \end{cases} \tag{4}$$

Substituting Eq. (3) and Eq. (4) into Eq. (1), the angular velocity of the transverse pendulum can be solved as:

$$\omega = \frac{v_r}{L}\tan\alpha \tag{5}$$

where ω is the vehicle traverse angular velocity. The steering radius R and front wheel deflection α can be obtained from ω and vehicle speed v_r.

$$\begin{cases} R = v_r / \omega \\ \alpha = \arctan(L / R) \end{cases} \tag{6}$$

From Eq. (3) and Eq. (5), the vehicle kinematic model is obtained as:

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \tan\alpha / L \end{bmatrix} v_r \tag{7}$$

As shown in the above equation, the position of the vehicle is indicated by the center point of the rear axle and the heading angle of the vehicle. When the deflection angle of the front wheels of the vehicle and the speed are known, the motion trajectory of the center point of the rear axle of the vehicle can be obtained through the above equation, and then according to the geometrical relationship of the vehicle, the trajectory of the vehicle at any point can be deduced.

3. Parking space path search and planning methods

3.1. Path search planning based on hybrid A^* algorithm

The study of the traditional A^* algorithm can be seen, the traditional A^* algorithm [29] in the node expansion, is to the center of the surrounding eight grids, the final search to get the path there is a large corner, does not satisfy the kinematic constraints of the vehicle normal driving. Compared with the traditional A^* algorithm, the hybrid A^* algorithm is mainly different in the node expansion method and the cost function setting, the hybrid A^* algorithm consists of two parts when performing node expansion, the first one is to consider the maximum corner constraints of the vehicle, which can be expanded by a certain front wheel corner and driving step. This kind of expansion can make the nodes no longer confined to the center of the grid, and the planned paths can satisfy the kinematic constraints of the vehicle. The second is to use Reeds-Shepp curve or Dubins curve to connect the current point and the target point, if there exists a collision-free path, the two points can be connected directly, if there is a collision, continue to use the vehicle kinematics equations to calculate the next node for expansion.

The cost function [30] of the hybrid A^* algorithm is similar to that of the A^* algorithm, which consists of actual and estimated generation values, except that the actual cost $g(s)$ calculation of the hybrid A^* algorithm takes more factors into account, as shown in Equation (8):

$$g(n) = P_1 n + P_2 \left|\delta(n_i)\right| + P_3 \left|\delta(n_i) - \delta(n_p)\right| + P_4 \tag{8}$$

where $\delta(n_i)$ is the front wheel angle at the current moment and $\delta(n_p)$ is the front wheel angle at the previous moment. n is the step length of each expansion. P_1 is the weight coefficient of the cost term of the vehicle forward, and the number of times the vehicle travels forward can be reduced by reducing the weight value. P_2 is the weight coefficient of the cost term when the vehicle turns, which can reduce the curvature of the planned path. P_3 is the steering angle change rate weight, which is favorable for the vehicle to make a smooth rotation, making the path easy to track. P_4 is the change in the direction of travel of the vehicle.

From the above analysis, it can be seen that the hybrid A^* algorithm can use Reeds-Shepp curve or Dubins curve to connect the current node and the target node during the node expansion process, and if the path is safe and collision-free, the search can be ended earlier to speed up the planning time, but the Dubins curve can only stipulate the forward motion of the vehicle, whereas the Reeds-Shepp curve not only allows the vehicle to move forward, but also allows the vehicle to move backward. movement, but also allows the vehicle to move backward, this advantage is suitable for parking scenarios, because most of the parking scenarios, the vehicle to complete the parking operation requires a combination of forward plus backward movement, and Reeds-Shepp curve [31] has 48 different combinations of movement, can be completely planned to the parking space within the parking path, therefore, this paper introduces the Reeds-Shepp curve into the Hybrid $A*$ algorithm during the node expansion process. Reeds-Shepp curve for parking lot path search.

The movement of the vehicle is divided into straight line, to the left, to the right, forward and backward in six ways, expressed in letters as L^+, L^-, R^+, R^-, S^+, S^-, of which, L indicates that the vehicle is traveling to the left turn, R indicates that the vehicle is traveling to the right turn, S indicates that the vehicle is traveling forward in a straight line, the positive and negative symbols in the upper right hand corner of the letter indicates the direction of the movement, and the number in the lower right hand corner of the letter indicates the length of the path of the travel, for example, $L^+_{\frac{\pi}{2}} S^+ R^+_{\frac{\pi}{2}}$, which path indicates that the vehicle first travels $\frac{\pi}{2}$ arc lengths to the left, then travels in a straight line, and travels $\frac{\pi}{2}$ arc lengths to the right. Based on the six different motion patterns, the Reeds-Shepp curves are divided into nine different categories, where the specific form of operation for each category is shown in Equation (9):

$$\begin{Bmatrix} C|C|C, CC|C, C|CC, CSC, CC_\beta|C_\beta C, C| \\ C_\beta C_\beta|C, C|C_{\frac{\pi}{2}}SC, CSC_{\frac{\pi}{2}}|C, C|C_{\frac{\pi}{2}}SC_{\frac{\pi}{2}}|C \end{Bmatrix} \tag{9}$$

where C indicates the steering of the vehicle. The symbol "|" indicates that the vehicle brakes, as well as changes the direction of travel.

The Reeds-Sheep curve is divided into 48 curve maneuvers based on these 9 types. Although Reeds-Sheep curves have 48 types of motions, the complexity of path solving can be reduced by utilizing the time-flip, reflection and backward transformation properties of the curves.

After the above simple description of the hybrid $A*$ algorithm and the Reeds-Sheep curve, it can be seen that the path planning using the hybrid $A*$ algorithm is to consider the kinematic constraints of the vehicle, and the node expansion process can be used to connect the current point and the target point using the Reeds-Sheep line, which on the one hand, can speed up the search speed of the hybrid $A*$ algorithm, and on the other hand, the curve can satisfy the kinematic constraints of parking process, and can include both forward and backward motion, and can include both forward and backward motion. On the other hand, the curve meets the kinematic constraints during the parking process and can contain both forward and backward trajectories, which is very suitable for parking scenarios.

3.2. Path search planning constraints

3.2.1. Collision constraints. In the collision constraint study, the vehicle is simplified as a rectangle, with the length and width determined by the maximum outer contour dimensions of the vehicle, while the garage line and the road edge line are simplified as lines on the plane. In the process of collision constraint research, the main purpose is to determine whether the four sides of the outer contour of the vehicle and the lines on the ground have intersections, which is used as a basis for determining whether the vehicle is in a collision or not. The method used in this paper is to determine whether two line segments on the plane intersect through the rapid exclusion experiment and the cross-stand experiment, and the judgment process of these two methods will be analyzed in detail below.

The first step is the rapid exclusion experiment. First of all, the principle of the rapid exclusion experiment is introduced, with AB, CD as the diagonal, draw two rectangles, and then determine whether the four sides of the two rectangles intersect.

1) Assume that there is no intersection of the two rectangles, then lines AB and CD must not

intersect.
2) Assuming the two rectangles intersect, lines AB and CD may or may not intersect.

The first need to determine whether the two line segments to form a rectangle intersect, when any point to meet, the two line segments no longer intersect, there is no need to carry out the next judgment, and only the rectangle intersects the need to continue to use the cross-experiment to determine whether the line segments intersect. Fast exclusion experiment judgment conditions are shown in the following equation:

$$\begin{cases} \max\left(A_x, B_x\right) < \min\left(C_x, D_x\right) \\ \max\left(A_y, B_y\right) < \min\left(C_y, D_y\right) \\ \max\left(C_x, D_x\right) < \min\left(A_x, B_x\right) \\ \max\left(C_y, D_y\right) < \min\left(A_y, B_y\right) \end{cases} \tag{10}$$

The second step is the straddle experiment. When the conditions of the rapid exclusion experiment are not met, it is necessary to use the straddle experiment to make a judgment. To make AB, CD two line segments intersect, then one of the line segments of the two points must be located on both sides of the other line segment, this endpoints are distributed in one of the line segments at both ends of the situation is known as the straddle experiment.

In order for lines AB and CD to intersect, the two lines must satisfy the mutual straddle, and when AB and CD intersect, vector $\overrightarrow{CD}$ is between vectors $\overrightarrow{CB}$ and $\overrightarrow{CA}$, and vector $\overrightarrow{BA}$ is between vectors $\overrightarrow{BC}$ and $\overrightarrow{BD}$. To summarize, the cross-product formula can be used to determine the intersection, as shown in the following equation:

$$\begin{cases} \vec{a} \times \vec{b} = \begin{vmatrix} i & j & k \\ l & m & n \\ o & p & q \end{vmatrix} = (mq - np, no - lp, lp - mo) \\ Which \begin{cases} a = (l, m, n) \\ b = (o, p, q) \end{cases} \end{cases} \tag{11}$$

The principle of the cross product formula is to take one of the vectors as the middle vector, and the beginning endpoint of the middle vector as the starting point of the other two vectors, and determine the positional relationship between the three vectors. If you want lines AB and CD to intersect, the following equation must be satisfied:

$$\begin{cases} \left[(Bx - Cx) * (Dy - Cy) - (By - Cy) * (Dx - Cx)\right] \\ * \left[(Dx - Cx) * (Ay - Cy) - (Dy - Cy) * (Ax - Cx)\right] \geq 0 \\ \left[(Cx - Bx) * (Ay - By) - (Cy - By) * (Ax - Bx)\right] \\ * \left[(Ax - Bx) * (Dy - By) - (Ay - By) * (Dx - Bx)\right] \geq 0 \end{cases} \tag{12}$$

In this paper, based on the above assumptions, the vehicle is simplified into a rectangle, and it is these four equivalent line segments that need to be collision detection, using these four equivalent line segments formed with the obstacles to conduct a rapid exclusion experiment and straddle experiments, when the formula (10) is satisfied, the vehicle does not collide with the obstacles. When (10) is not satisfied, the straddle experiment is needed, if (12) is satisfied in the straddle experiment, the vehicle collides with the obstacle, while if (12) is not satisfied, the vehicle does not collide with the garage and the road boundary.

3.2.2. Consideration function constraints. In this paper, hybrid $A*$ algorithm is used for parking lot path search planning, if the cost function calculation is not constrained, it will generate too many steering and backward movements, which will lead to frequent steering and acceleration/deceleration operations of the vehicle, making path tracking more difficult, and also adding too many path nodes. Therefore designing constraint parameters for the cost function helps to optimize the search path.

The total cost function value is $\mathrm{Cos}\,t$, the forward cost function value $\mathrm{Cos}\,t_s$, the steering cost function value $\mathrm{Cos}\,t_c$, the steering angle change cost function value $\mathrm{Cos}\,t_{ac}$, and the heuristic function cost function value $\mathrm{Cos}\,t_{he}$. By setting the coefficients of the various cost function values, it is possible to decide on the number of nodes for various motions. The formula is:

$$\mathrm{Cos}\,t = l_1 * \mathrm{Cos}\,t_s + l_2 * \mathrm{Cos}\,t_c + l_3 * \mathrm{Cos}\,t_{ac} + l_4 * \mathrm{Cos}\,t_{he} \tag{13}$$

Where l_1, l_2, l_3, l_4 are the coefficients of each cost function, which determines the search path by controlling the magnitude of the coefficient values, where the values of each node are, respectively, $l_1 = 5$, $l_2 = 10$, $l_3 = 20$, $l_4 = 1$.

4. Parking experiment test analysis and results

4.1. Parking simulation environment construction

CarSim [32] is a platform software specially designed for dynamics simulation experiments, in which three main functional modules are designed, which mainly include the whole vehicle model parameter settings and road environment settings, setting the control quantities of vehicle inputs and outputs, and animation and plotting to display post-processing simulation results. Due to the reliable dynamics model and real vehicle data behind it, combined with the rich vehicle input and output control quantities, it is convenient for many researchers to carry out rapid control model validation and development, and is now widely used in the development of automatic driving functions. In this paper, we utilize CarSim and Simulink to build a parking simulation platform environment, and use this simulation experiment to prove the feasibility of the parking space path search and planning algorithm.

4.1.1. Vehicle Model Selection and Input/Output Settings. The CarSim software has different types of vehicle models, and the vehicle model chosen in this paper is a B-class Hatchback, with the vehicle parameters of wheelbase L (2.54m), front overhang L_{f} (0.87m), rear overhang L_{r} (0.72m), and vehicle width L_{w} (1.528m).

4.1.2. Parking Scenario Conditions. In the co-simulation of automatic parking, the first need to build the parking space environment in CarSim, because the parking space path search and planning in this paper involves parallel parking and perpendicular parking two conditions. In order to make the parking scenario coincide with the actual situation. When building the parking space in CarSim software, the left and right sides of the parking space are added to the target vehicle module in 3D Shapes Moving Object, and the vehicle position is determined by setting Station and Lateral offset, and redefining the obstacle vehicle's heading angle and speed, the vehicle obstacle can be added to the left and right sides of the parking space, and also on both sides of the road. Add the House module in 3D Shapes Repeated Object.

4.2. Simulation effect analysis of parking under different working conditions

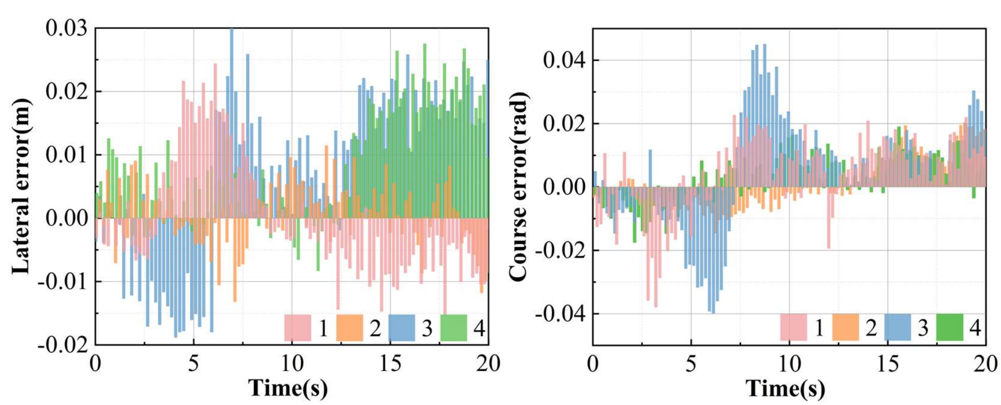
In this section, the overall validation of the automatic parking strategy containing the planning algorithm and the control algorithm is carried out by constructing the scenarios of perpendicular and parallel parking, which are common in parking lots.

4.2.1. Parallel Parking Simulation Results. In order to solve the problem that the path search and planning algorithms for parking spaces have stringent requirements on the path starting area, it is proposed to use hybrid A* path search and planning algorithms to plan a path for different starting positions, and to establish parallel parking simulation experiments according to different starting points. Parallel parking simulation experiments with different starting positions are shown in Table 1, including a total of four parallel parking conditions.

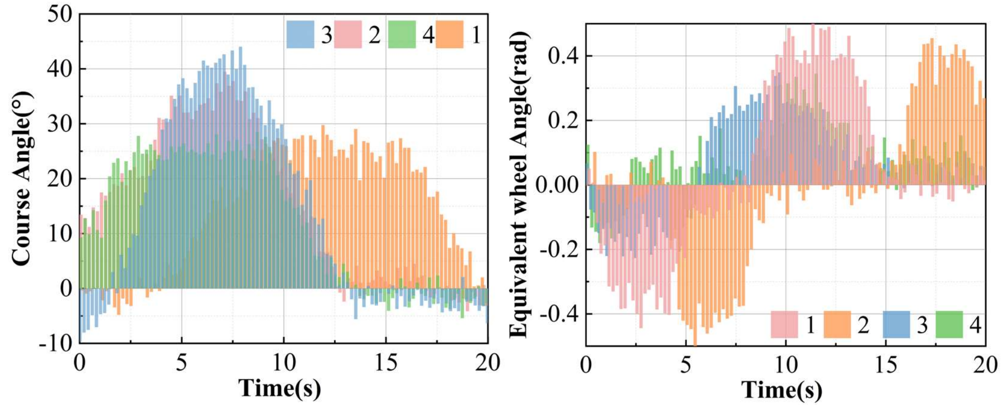
Table 1. The different starting points of the parallel parking simulation experiment

Operating condition	Initial capital (x, y, θ)	End point (x, y, θ)
Operating condition 1	(8,2,8,0)	(1.0635, -1.0263,0.042)
Operating condition 2	(13,1,7,0)	(1.0635, -1.0258,0.042)
Operating condition 3	(8,2,8, -0.25)	(1.5,0, -1.5)
Operating condition 4	(8,2,8,0.26)	(1.3, -1.0263,0)

The results of the analysis of the lateral and heading errors, the equivalent front wheel turning angle of the vehicle at each moment, and the vehicle heading angle for the four parking conditions are shown in Fig. 1, with (a)-(d) representing the results of the analysis of the lateral error of the vehicle, the heading error of the vehicle, the vehicle heading angle, and the equivalent front wheel turning angle of the vehicle, respectively. From the comparison of the lateral error and heading error for all the conditions, it can be clearly seen that the overall error of condition 3 is very large, with the maximum value of the lateral error being 0.03 m, and the maximum of the heading error being close to 0.045 m. The analyzed reason for this situation is related to the limitation of the vehicle's own physical mechanism, and the starting attitude is not conducive for the vehicle to carry out tracking and driving due to the fact that the starting position is too close to the garage, when the vehicle tilts at a negative angle, the The vehicle needs to satisfy the non-integrity constraints of the vehicle and cannot adjust the attitude of the vehicle very quickly, the vehicle will be adjusted gradually in the process of movement. Moreover, this condition is composed of two circular arcs with smooth segments, and the curvature of the path varies greatly, which leads to a larger error in the tracking process. As for the paths of the other three working conditions, path 2 is far away from the parking space due to the path, path 4 is a positive tilt angle, and there is a large proportion of straight lines in the parking space search path, and the transition stage between the circular arc and the straight line has been smoothed, and the tracking process of the paths is relatively easy, so that the tracking transverse error and heading error are both relatively small. Meanwhile, Case 2 is farther away from the parking space, and the intermediate parking process has more time for position adjustment, which makes the tracking transverse and longitudinal errors relatively small, with the maximum transverse error of 0.013 m and the maximum heading error of 0.036 m. The tracking effects of the four simulation scenarios show that the paths planned for the different starting points of the Hybrid A* algorithm are able to maintain good tracking accuracy. Therefore, it can be assumed that the improved hybrid A* algorithm has solved the problem of high requirements on starting point postures in the search and planning process of automatic parking paths in the simulation scenarios, and the hybrid A* algorithm can plan a feasible path without collision according to different starting point postures.



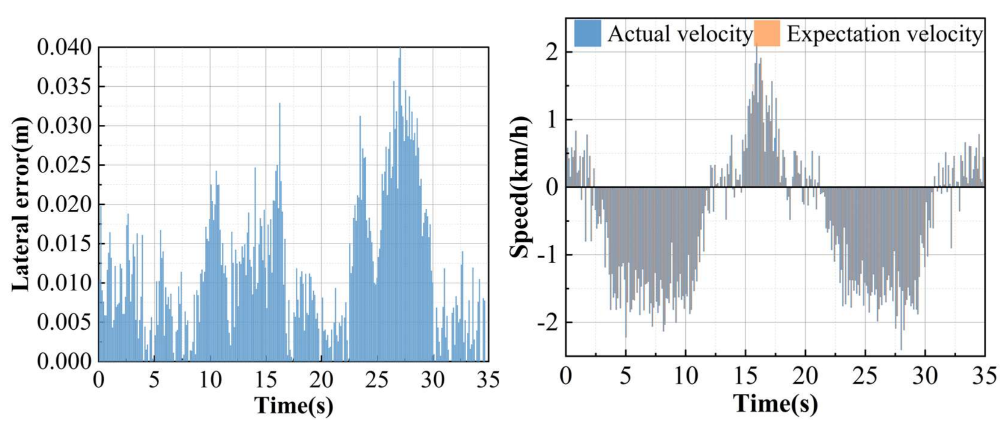
(a) Lateral error (b) Course error



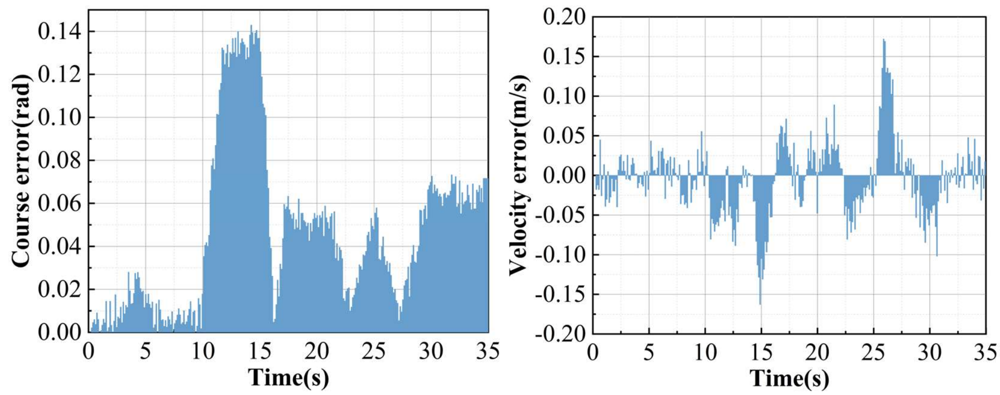
(c) Course Angle (d) Equivalent wheel Angle

Fig. 1. Parallel parking path different position path tracking simulation diagram

4.2.2. Vertical Parking Scenario Analysis. In the vertical parking scenario, the nearest distance between the boundary vehicles forming the parking space is 2.8 m. The initial position of the vehicle represented by the center of the rear axis is (14.52, 5.74), and the heading angle is 0°, and the vehicle will be parked with the center of the parking space as the target, with the target position of (12.63, 0.851) and the heading angle of 90°. Its planning path and tracking control effect are shown in Fig. 2, (a)-(d) represent the analysis results of lateral error, velocity, heading error and speed error, respectively. Due to the small width of the parking space in this scenario and the small initial lateral distance, the planning algorithm obtains a parking path with two reversing processes, and the whole journey is divided into three segments: reverse, forward and reverse. And the algorithm proposed in this paper can realize the tracking process better, and its lateral error is mostly concentrated in less than 0.025 m, and the maximum value is about 0.04 m. Due to the relative complexity of the path, its heading error reaches its maximum in the first reversing process (10.67-15.86s), with the maximum value of 0.142 rad, and the rest of the time is less than 0.07 rad. in the speed design, due to the intermediate forward section length is short, the vehicle cannot accelerate to the parking stable speed of 2km/h throughout the whole process. the overall speed tracking error is concentrated below 0.172m/s, and the fluctuation mainly occurs at the intersection of the uniform speed section and the variable speed section, but the fluctuation can be adjusted to track the desired speed very quickly after the fluctuation.



(a) Lateral error (b) Speed



(c) Course error (d) Velocity error

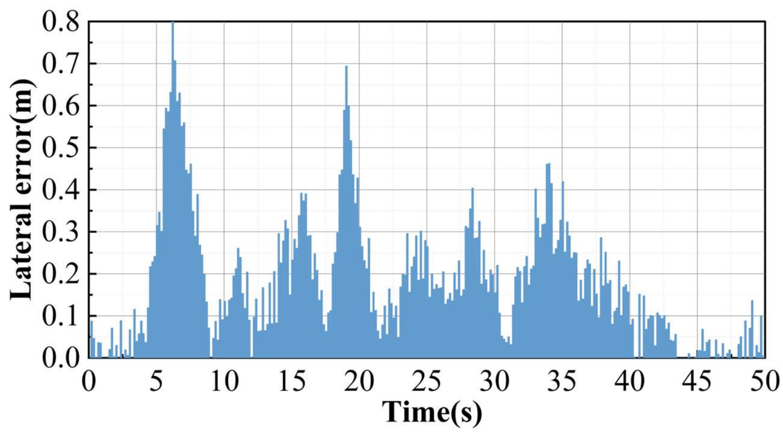
Fig. 2. Vertical parking test results

4.3. Prototype controller in-loop test validation analysis

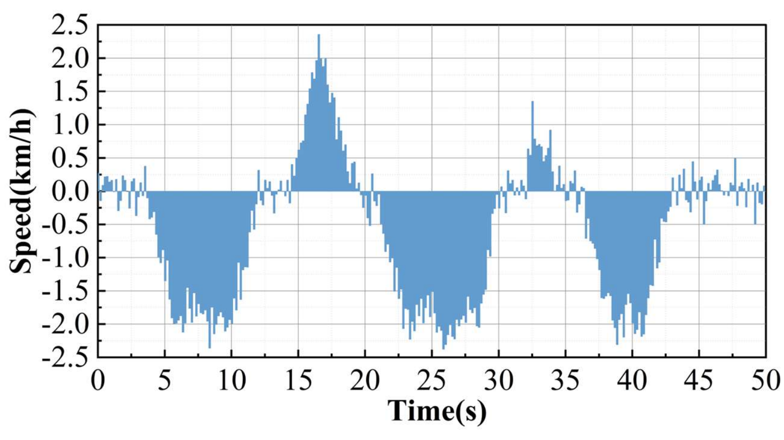
4.3.1. Test platform construction. Pure simulation test conditions are too ideal to reflect the performance of the algorithm in the actual working environment, therefore, this paper builds a prototype controller in-loop test platform, introduces hardware equipment to simulate more realistic working conditions, and verifies the performance of the algorithm. The prototype controller in-loop test platform consists of a model controller module and a vehicle dynamics module. There are two main functions realized by the model controller module. First, the automatic parking strategy model is built in the upper computer1 and downloaded and compiled into the controller to run in real time, forming the object under test. Secondly, the automatic parking strategy to be tested receives the vehicle signal output from the vehicle dynamics module, and then processes the corresponding control strategy and sends it to the vehicle dynamics module to realize the vehicle motion control.

The model controller module designed in this paper is realized based on dSPACE real-time simulation system. The MicroAutoBoxII in the standard component system is used as the hardware platform for real-time simulation, which is equipped with DS1401 quad-core processor. At the same time, the communication between the model controller module and the vehicle dynamics module is realized by the CAN bus, and the DS1513 is selected as the simulation board for CAN bus communication. The core of the vehicle dynamics module adopts the classic simulation test software Carsim in the field of vehicle dynamics simulation, which is able to configure the vehicle body model according to the demand and obtain more accurate driving behavior. The configuration process is carried out in the upper computer 2. Then the dynamics model is downloaded to the dSPACE Simulator system to run in real time, which accepts the vehicle control signals sent from the model controller module, responds to the actions in real time according to the current vehicle motion state, the designed vehicle dynamics model, etc., and jointly solves the vehicle's motion state information for the next moment and outputs it to the model controller module. As the platform introduces hardware devices, it is closer to the real working environment and can test the real-time performance of the algorithm.

4.3.2. Testing and analysis of results. The typical scene under the vertical parking test scenario above is tested, and its parking space width is the width of the vehicle plus 0.85m, which is the smallest value in the range of values taken for the width of the parking space in this paper, and the drivable area is 5.2m which is also the smallest value taken for the value in this paper, and at the same time, the lateral distance is 0.5B+1.1m, which is nearer to the obstacle of the drivable area. The key information of its final test results is shown in Fig. 3, (a) and (b) represent the analysis results of lateral error and speed, respectively. From the analysis results, it can be seen that the path search planning algorithm proposed in this paper can get the parking path containing 3 times of direction change process through the front-end path search process. Secondly, the back-end optimization process has an optimization effect but is not obvious, this is because the scene occurs all around the garage, the parking process is divided into 4 segments by the 3 times of change of direction process, each segment distance is very short, and the parking space is only wider than the width of the car by 0.85m, in order to avoid collision, the space left for the optimization is smaller, but it is found that the back-end optimization obtains the smoother path from the results. From the analysis results in the figure, it can be seen that due to the introduction of a test platform closer to the real working conditions, although the parking space path search and planning algorithm proposed in this paper can realize the vehicle path tracking control, but compared with the pure simulation test the lateral error has increased, the average lateral error during the whole parking process is 0.177 m. The fluctuation of the speed control has also increased, especially in the time period of 4.5-11.36s and 21.20-29.70s, the tracking effect decreases. However, on the whole, the effectiveness of the parking space path search and planning algorithm proposed in this paper is verified in real test scenarios. In addition, while verifying the effectiveness of the algorithm, it shows that the algorithm still needs to strengthen its robustness to the working environment.



(a) Path tracking transverse error



(b) Vehicle speed

Fig. 3. The prototype controller is tested in the ring test

5. Conclusion

This paper combines the A* algorithm and the Reeds-Shepp curve to construct a parking space path search and planning model, and verifies the effectiveness and accuracy of the algorithm through simulation experimental scenarios and prototype controller-in-the-loop test scenarios, and the results show that:

1) Work condition 2 in the parallel parking scenario is farther away from the parking space, and the intermediate parking process has more time for position adjustment, which makes the tracking transverse and longitudinal errors relatively small, and the maximum transverse error and heading error are only 0.013m and 0.036m. Most of the lateral errors in the vertical parking process are concentrated below 0.025m, and the heading error can still be kept below 0.142rad in the case of path complexity.
2) The average lateral error during the whole parking process in the ring test scenario is 0.177m, although the tracking effect has decreased, the overall error is smaller, and the effectiveness of the parking space path search and planning algorithm is verified.

In this paper, the feasibility of the parking space path search and planning algorithm and the accuracy of tracking control is verified after simulation experiments and real car test scenes, but the experiments found that the algorithm still has some defects, and in the future, a more comprehensive generation of parking guidance scenes can be obtained through the statistical analysis of objective data and other ways, so as to improve the performance of the algorithm in the parking system.

Funding

This work was supported by the Higher Vocational Education Reform Project of Henan Province (No 2024SJGLX0906).

References

[1] Dogaroglu, B., Caliskanelli, S. P., & Tanyel, S. (2021). Comparison of intelligent parking guidance system and conventional system with regard to capacity utilisation. Sustainable Cities and Society, 74, 103152.

[2] Hanzl, J. (2020). Parking information guidance systems and smart technologies application used in urban areas and multi-storey car parks. Transportation Research Procedia, 44, 361-368.

[3] Iszaidy, I., Ngadiran, R., Nazren, A. R. A., Wafi, N. M., Ramli, N., Vijayasarveswari, V., & Leow, W. Z. (2021, July). IOT Smart Guidance Parking Search System for Open Space Parking Area. In Journal of Physics: Conference Series (Vol. 1962, No. 1, p. 012060). IOP Publishing.

[4] Li, Q., & Chen, Q. (2022, August). Design of parking guidance system in intelligent parking lot. In International Symposium on Artificial Intelligence Control and Application Technology (AICAT 2022) (Vol. 12305, pp. 142-147). SPIE.

[5] Neykov, N., & Stefanova, S. (2022). Case-Based Reasoning Application for Parking Guidance Systems. IFAC-PapersOnLine, 55(11), 150-154.

[6] Gao, H., Yun, Q., Ran, R., & Ma, J. (2021). Smartphone-based parking guidance algorithm and implementation. Journal of Intelligent Transportation Systems, 25(4), 412-422.

[7] Rizvi, S. R., Zehra, S., & Olariu, S. (2018). Aspire: An agent-oriented smart parking recommendation system for smart cities. IEEE Intelligent Transportation Systems Magazine, 11(4), 48-61.

[8] Ren, C., Lee, S., Jeong, D., Chen, H., & Xiao, Y. (2022). Parking guidance system based on geomagnetic sensors and recurrent neural networks. Journal of Sensors, 2022(1), 7481064.

[9] Wang, X., Shi, H., & Zhang, C. (2020). Path planning for intelligent parking system based on improved ant colony optimization. IEEE Access, 8, 65267-65273.

[10] Han, Z., Sun, H., Huang, J., Xu, J., Tang, Y., & Liu, X. (2024). Path Planning Algorithms for Smart Parking: Review and Prospects. World Electric Vehicle Journal, 15(7), 322.

[11] Pan, F., Pan, H., Zhang, L., Ma, C., Yang, J., & Kang, G. (2021). Accurate Guidance Method and App Development for Assigning Parking Spaces Based on Indoor Wi-Fi. Promet-Traffic&Transportation, 33(3),

347-359.

[12] Manvanth, G. S., Kumar, A. P., Priyanka, V. D., Goduguluri, V., & Ravish, R. (2024, February). Smart Monitoring and Guidance System for Parking Spaces. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT) (Vol. 5, pp. 1256-1262). IEEE.

[13] Neykov, N., & Stefanova, S. (2024, February). Training data for parking guidance system. In AIP Conference Proceedings (Vol. 3063, No. 1). AIP Publishing.

[14] Mouhcine, E., Yassine, K., Mansouri, K., & Mohamed, Y. (2020, December). Smart Parking Guidance System Based on Distributed IoT Strategy. In International Conference on Advanced Intelligent Systems for Sustainable Development (pp. 74-86). Cham: Springer International Publishing.

[15] Shen, T., Hua, K., & Liu, J. (2019). Optimized public parking location modelling for green intelligent transportation system using genetic algorithms. IEEE Access, 7, 176870-176883.

[16] Gu, S. S., Sun, X. F., Wang, M., & Yu, J. Q. (2022). Research on Parking-Space Occupancy Recognition Based on MobileNet and Intelligent Parking Guidance Strategy. Journal of Highway and Transportation Research and Development (English Edition), 16(3), 78-84.

[17] Hassine, S. B., Mraihi, R., & Kooli, E. (2022). Smart parking guidance: a step towards sustainable cities. International Journal of Scientific and Research Publications, 12, 145-156.

[18] Yoo, S. E. (2019). SPGS: Smart Parking Space Guidance System based on User Preferences in a Parking Lot. Journal of the Korea Industrial Information Systems Research, 24(4), 29-36.

[19] Liu, Y., Liu, X., Wang, S., & Tian, R. (2023, February). Research on parking guidance system based on computer vision. In Journal of Physics: Conference Series (Vol. 2425, No. 1, p. 012054). IOP Publishing.

[20] Wang, H. (2021). Urban integrated intelligent parking guidance based on Dijkstra algorithm. Advances in Transportation Studies.

[21] Lin, Y., Su, C., & Zhu, L. (2015). A Novel Parking Guidance and Management System: A Case Study for Jinan, China. In CICTP 2015 (pp. 281-295).

[22] Liu, J., Wu, J., & Sun, L. (2020). Control method of urban intelligent parking guidance system based on Internet of Things. Computer Communications, 153, 279-285.

[23] Shin, J. H., Jun, H. B., & Kim, J. G. (2018). Dynamic control of intelligent parking guidance using neural network predictive control. Computers & Industrial Engineering, 120, 15-30.

[24] Zhang, X., Yu, L., Wang, Y., Xue, G., & Xu, Y. (2017, March). Intelligent travel and parking guidance system based on Internet of vehicle. In 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 2626-2629). IEEE.

[25] Zhu, H. (2024, August). Application Research of Intelligent Parking System Based on Parking Guidance Technology. In Proceedings of the 2024 4th International Conference on Internet of Things and Machine Learning (pp. 391-395).

[26] Chai, H., Ma, R., & Zhang, H. M. (2019). Search for parking: A dynamic parking and route guidance system for efficient parking and traffic management. Journal of Intelligent Transportation Systems, 23(6), 541-556.

[27] Zou, W., Sun, Y., Zhou, Y., Lu, Q., Nie, Y., Sun, T., & Peng, L. (2020). Limited sensing and deep data mining: A new exploration of developing city-wide parking guidance systems. IEEE Intelligent Transportation Systems Magazine, 14(1), 198-215.

[28] Gao, F., & Chen, A. (2022). Research on intelligent city parking guidance method based on ant colony algorithm. International Journal of Innovative Computing and Applications, 13(1), 1-10.

[29] Dang Chien Van,Ahn Heungju,Lee Doo Seok & Lee Sang C. (2022). Improved Analytic Expansions in Hybrid A-Star Path Planning for Non-Holonomic Robots. Applied Sciences(12),5999-5999.

[30] Artzai Picon,Daniel Mugica,Itziar Eguskiza,Arantza Bereciartua Perez,Javier Romero,Carlos Javier Jimenez... & Ramon Navarra Mestre. (2025). Taxonomic hierarchical loss function for enhanced crop and weed phenotyping in multi-task semantic segmentation. Smart Agricultural Technology100761-100761.

[31] Ugo Boscain & Francesco Rossi. (2010). Projective Reeds-Shepp car on S 2 with quadratic cost. ESAIM: Control, Optimisation and Calculus of Variations(2),275-297.

[32] Dong Enguo,Li Yajuan,Sun Boli & Zhang Xiaojun. (2023). Vehicle Braking Stability Analysis Considering Vehicle Structure Parameters Based on CarSim. Journal of Physics: Conference Series(1).