Robotica (2026), **44**, pp. 883–907
doi:10.1017/S0263574726103324

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Two-step path planning for vertical parking in narrow non-ideal scenarios based on predefined path pattern and numerical optimization

Xiaoxiao Lv[1,2], Wenrui Jin[2,3], Jiaxue Li[2], Xiangping Qiu[4,5], Fan Mo[6,7], Min Fang[2], Qingyang Yu[8], Ying Yu[2] and Jiangtao Yu[9]

[1]Postdoctoral Station of Mechanical Engineering, Tongji University, China
[2]School of Mechanical Engineering, Tongji University, China
[3]Sino-German College of Applied Sciences (CDHAW), Tongji University, China
[4]First Institute of Telecommunications Technology, Shanghai, China
[5]Yangtze Communications Industry Group Co., Ltd., Wuhan, China
[6]Department of Computer Science, University of Oxford, UK
[7]Department of Computer Science and Technology, University of Cambridge, UK
[8]College of Electronic and Information Engineering, Tongji University, China
[9]College of Civil Engineering, Tongji University, China
Corresponding author: Wenrui Jin; Email: wrjin@tongji.edu.cn

Received: 9 May 2025; **Revised:** 18 February 2026; **Accepted:** 20 February 2026; **First published online:** 5 May 2026

Keywords: autonomous parking; path planning; non-ideal parking scenarios; numerical optimization; clothoid curve

Abstract

Parking in narrow spaces presents significant challenges, often resulting in deviations between vehicle's final parking state (FPS) and normal FPS. These deviations can cause partial intrusion into the target parking spots, increasing the risk of collisions in the parking process and potentially making the parking spots unusable. To address these issues, this paper proposes an optimization method for both the parking path and FPS in narrow and non-ideal vertical parking scenarios. Initially, a partition-based calculation method for the minimum distance between the vehicle and obstacles (DBVO) was developed to quantitatively assess the impact of intrusion on parking safety. Following this, a predefined geometric set of clothoids is used to smooth curvature discontinuities in the parking path, and a four-phase parking path pattern is devised. Subsequently, a preferred method for parking manners is proposed by analyzing the effects of parking direction and maneuvers on spatial requirements. Finally, a two-step parking path optimization method is presented with the framework integrating both online and offline calculations, using the minimum DBVO field and a predefined path pattern. Comparative experiments demonstrate that this method could enhance the proportion of available intrusion scenarios, increase the success rate of effective path acquisition, and improve the overall quality of parking paths.

1. Introduction

With the continuous increase in vehicle ownership, parking lots are in short supply, leading to the parking lots and the space near the parking spots (SNPS) becoming increasingly narrow, thus rendering manual parking operations more challenging [1]. Unskilled parking operations may disturb the surrounding road users and potentially cause traffic jams [2]. Besides that, inexperienced drivers who are reluctant to try narrow parking lots would have to drive for longer periods of time and distance, thereby contributing to additional air pollution, fuel consumption, and congestion [3]. Therefore, automatic parking technology for narrow parking spaces has gradually become a research focus [4].

The existing literature review centers on four techniques for automatic parking path planning. In specific parking situations, geometric approaches are applied [5], leveraging analytical expressions to rapidly pinpoint key trajectory points. For instance, Han [6] put forward a geometric collision-free path

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

planning method. Additionally, Clothoid, Bezier, and spline curves are utilized for path smoothing [7]. Search methods discretize the continuous space into grids and search for a feasible parking path. Dolgov et al. [8] introduced the Hybrid A* algorithm with Reeds-Shepp curve, which takes vehicle kinematics into account and employs more suitable heuristic functions. Subsequently, enhancement strategies are devised to boost path security, adaptability, and planning efficiency [9, 10]. Sampling methods generate sample points in continuous space and aim to find a kinematically feasible and obstacle-avoiding point sequence as the planning outcome. Kuwata et al. [11] considered vehicle kinematic constraints, assessed sampling-point feasibility via Dubins path length, and expanded nodes accordingly. To improve computational efficiency, Han et al. [12] proposed the Bidirectional Rapidly Random Tree (RRT) algorithm. Numerical optimization methods formulate trajectory planning as optimal control problems [13]. Proper description and handling of obstacle-avoidance constraints are vital for parking trajectory planning using these methods [14]. Algorithms like Particle Swarm Optimization and Genetic Algorithm are employed to solve path optimization problems. Wang et al. [15] proposed a hierarchical algorithm combining nonlinear optimization and an improved RRT with Reeds-Shepp curves to address inefficiencies and discontinuous curvature in parking trajectory planning, enabling continuous-curvature paths across diverse scenarios. Yang et al. [16] proposed a Sobol-RRT algorithm with Reeds-Shepp curves and optimal control to improve parking efficiency and dynamic obstacle handling in four-wheel steering vehicles. Li et al. [17] presented a Kinodynamic-Informed-Bidirectional Rapidly-exploring Random Tree Star algorithm to reduce unnecessary exploration near concave obstacles and generate kinematically admissible paths.

Based on the above-mentioned path planning methods, a general automatic parking system (APS) was developed in conjunction with the vehicle control system [18]. In recent years, with the rapid advancement of artificial intelligence, end-to-end APS [19], which integrates the path planning module and path tracking control module, has emerged. It directly outputs control variables to control the vehicle based on environmental information. Current implementation methods for end-to-end APS are deep learning and deep reinforcement learning (DRL) [20]. It offers significant advantages not only in the end-to-end parking system but also in the parking lot allocation system [21]. Many researchers have conducted numerous studies on this. Zhang et al. [22] proposed a DRL-based APS and compared its parking performance with that of a general APS. The results indicated that the DRL-based APS could park more easily with fewer deviations. Chai et al. [23] proposed an improved DNN-based motion planning algorithm and a DRL-based online collision avoidance approach in uncertain environments. Song et al. [24] developed a parking model using actual parking data, addressing the limitation of model-free RL that requires extensive interactive data and enhancing training efficiency. Bernhard et al. [25] applied experience-based heuristic DRL to automatic parking, using human expertise to evaluate the explored data. Zhang et al. [26] regarded parking as a multi-objective optimization process considering safety, comfort, parking efficiency, and the FPS. Chai et al. [27] designed a multilayer control scheme combining optimal trajectory planning and deep learning to guide the parking maneuver of autonomous ground vehicles (AGVs), and in [28], they developed a novel RDNN-based motion planning approach for real-time parking movement planning of different AGVs. The end-to-end APS can solve path planning problems in uncertain dynamic environments. Nevertheless, it has the following drawbacks: limited generalization ability, a sharp decline in performance as the training task difficulty increases, and overly long training times for the system model [29].

The above methods can enhance parking efficiency, reduce search time, and adapt to conventional parking spots. However, due to the constraints of narrow parking spaces and the limitations of drivers' skills, the final parking state (FPS) of the vehicle often deviates, leading to partial encroachment into target parking spots and increased risk of collision during parking [2, 30]. For narrow spaces, Lian et al. [31] designed an enhanced hybrid A* algorithm to address the search problem in narrow and complex environments. Li and Wang [32] proposed a parallel parking path planning method in narrow spaces based on a three-stage curve interpolation approach. Vorobieva et al. [7] designed a parallel parking path within small parking spaces using a continuous curvature curve improved by a clothoid curve. Li and Shao [33] put forward a unified motion planning method for autonomous vehicle parking, which

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

can effectively achieve path planning in the presence of irregularly placed obstacles. However, due to the comprehensive consideration of various factors, its real-time performance is slightly weak. Regarding non-standard scenarios, Tang et al. [29] proposed a segmented parking training framework based on soft actor-critic to reduce the impact of vehicle intrusion on safety. Nevertheless, this method is not suitable for unfamiliar scenarios because of the inherent characteristics of reinforcement learning.

Current methods are rarely applied to narrow non-ideal scenarios for vertical parking, and their performance remains uncertain. Furthermore, there is a lack of characterization for non-ideal parking scenarios resulting from intrusion and a corresponding evaluation index for these scenarios.

In light of these issues, the contributions of this study are summarized as follows: (1) Definition and Characterization of Non-Ideal Parking Scenarios: The study defines and characterizes non-ideal parking scenarios using the deviation angle and offset distance relative to the actual parking spot boundaries. (2) Safety Index and Fast Calculation for Non-ideal Parking scenarios: A safety index based on the minimum distance between vehicles and obstacles (DBVO) is introduced for non-ideal parking. A partition-based method for calculating the minimum DBVO is proposed to quickly update these scenarios, enabling adaptability to various intrusion scenarios. (3) Four-Phase Parking Path Pattern: A four-phase parking path pattern is developed for non-ideal scenarios, incorporating straight lines, arcs, and clothoids. This pattern provides greater flexibility for the entry path to enhance safety distance while ensuring curvature continuity. (4) Preferred method for Parking Manners: A preferred rapid method for parking manners, including parking direction and number of maneuvers, is proposed based on the spatial demands of each parking manner. (5) Two-Step Optimization Framework: A two-step optimization framework combining online and offline methods is established for path planning in non-ideal parking scenarios. The off-line computation of preliminary data improves the efficiency of the online calculations, facilitating effective real-time path planning. The first step focuses on optimizing the FPS and entry path for improved safety, while the second step aims to improve parking efficiency by optimizing the path beyond the parking spot, thus enhancing overall path quality.

2. Quantitative characterization of the safety of intrusion parking scenarios

2.1. Parking scenarios

This study investigates two types of parking scenarios: ideal and actual ones. The actual scenario takes into account the impact of the FPS deviation of adjacent vehicles. As shown in Figure 1, adjacent vehicles 1–4 are within the normal FPS, while vehicles 5-10 deviate from it. Thus, parking spots 1 and 2 represent ideal scenarios without intrusion, and parking spots 3 and 4 are actual scenarios with one-sided and two-sided intrusions, respectively.

2.2. Safety index indicators for parking scenario with intrusion

The actual parking spot is determined by the edges of adjacent vehicles, as shown in Figure 2. For modeling purposes, a coordinate system O_PXY is established with the geometric center of the original parking spot as the origin. The offset distance and angle of the left and right edges (Δx_L, Δx_R, $\Delta\beta_L$ and $\Delta\beta_R$) are used to describe the parking scenario. Then, the coordinates of the four vertices of the actual parking spot are

$$\begin{cases} x_{\mathrm{LU}} = -\dfrac{W_{\mathrm{P}}}{2} + \frac{L_{\mathrm{P}}}{2} \cdot \cot\beta_{\mathrm{L}} + \Delta x_{\mathrm{L}}, \quad y_{\mathrm{LU}} = \dfrac{L_{\mathrm{P}}}{2} \\ x_{\mathrm{LB}} = -\dfrac{W_{\mathrm{P}}}{2} - \dfrac{L_{\mathrm{P}}}{2} \cdot \cot\beta_{\mathrm{L}} + \Delta x_{\mathrm{L}}, \quad y_{\mathrm{LB}} = -\dfrac{L_{\mathrm{P}}}{2} \\ x_{\mathrm{RU}} = \dfrac{W_{\mathrm{P}}}{2} + \dfrac{L_{\mathrm{P}}}{2} \cdot \cot\beta_{\mathrm{R}} - \Delta x_{\mathrm{R}}, \quad y_{\mathrm{RU}} = \dfrac{L_{\mathrm{P}}}{2} \\ x_{\mathrm{RB}} = \dfrac{W_{\mathrm{P}}}{2} - \dfrac{L_{\mathrm{P}}}{2} \cdot \cot\beta_{\mathrm{R}} - \Delta x_{\mathrm{R}}, \quad y_{\mathrm{RB}} = -\dfrac{L_{\mathrm{P}}}{2} \end{cases} \tag{1}$$

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

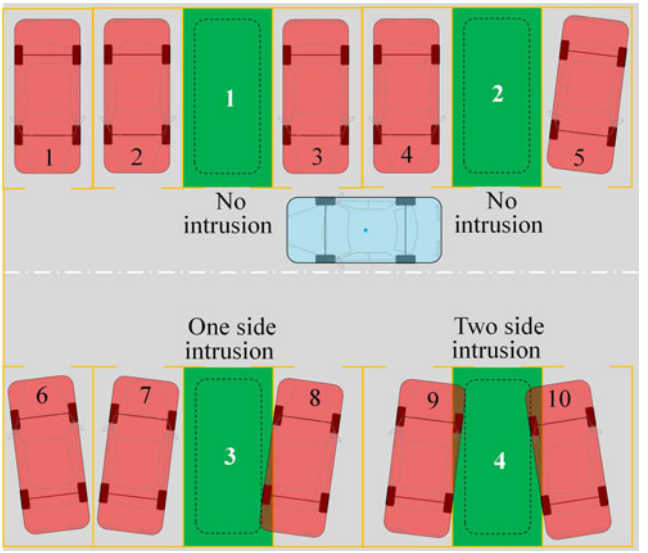


Figure 1. *Ideal parking scenarios and actual parking scenarios.*

where Δx_L and Δx_R are the offset distances of the mid-points of the left and right edges, respectively. β_L and β_R are the inclination angles of the left and right edges ($\beta_L = \Delta\beta_L + \frac{\pi}{2}$, $\beta_R = \Delta\beta_R + \frac{\pi}{2}$), respectively. L_P and W_P are the length and width of the original parking lot, respectively.

The shape of the actual parking spot is only determined by the total offset distance Δx_O ($\Delta x_O = \Delta x_L + \Delta x_R$) and the offset angles $\Delta\beta_L$ and $\Delta\beta_R$. Thus, the vectors $[\Delta x_O, \Delta\beta_L, \Delta\beta_R]$ are used to characterize the actual parking scenarios.

Generally, SNPS, including the width of the corridor (W_C) and the front space (S_F), is limited, resulting in a higher risk of collision during parking. The intrusion of adjacent vehicles further increases this risk. As shown in 2, to ensure safety, a rectangular expansion is applied to the vehicle boundary, the expansion distance indicated as d_E, which is set at 0.1 m in this study. During the parking process, the ego vehicle is subject to constraints from the left, right, and upper sides, as well as bottom constraints defined by the boundaries of the actual parking spot. The minimum DBVO is used for safety evaluation.

The vehicle state is generally characterized by the coordinates of the rear axle center point and heading angle, represented as $\boldsymbol{S} = [x_E, y_E, \theta]$. To take advantage of the vehicle's symmetry, the state S_{OV} ($S_{OV} = [x_{OV}, y_{OV}, \theta]$), based on the geometric center of the vehicle, is adopted for safety analysis. The conversion formula between these two states is

$$\begin{cases} x_{\mathrm{OV}} = x_{\mathrm{E}} + \left(\dfrac{L_{\mathrm{V}}}{2} - l_{\mathrm{R}}\right)\cos\theta \\ y_{\mathrm{OV}} = y_{\mathrm{E}} + \left(\dfrac{L_{\mathrm{V}}}{2} - l_{\mathrm{R}}\right)\sin\theta \end{cases} \tag{2}$$

where (x_{OV}, y_{OV}) are the coordinates of the geometric center of the vehicle. L_V is vehicle length. l_R is length of vehicle rear overhang.

The process for calculating the minimum distance between the vehicle and each obstacle is as follows. For the left, right, and upper constraints, the coordinates of the four expanded vertices of the vehicle are first computed for each heading angle, assuming the vehicle's center is at the origin.

$$\begin{bmatrix} \mathbf{x}_{\mathrm{V}}(\theta) \\ \mathbf{y}_{\mathrm{V}}(\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathrm{V}}(0) \\ \mathbf{y}_{\mathrm{V}}(0) \end{bmatrix} \tag{3}$$

where $x_V(\theta)$ and $y_V(\theta)$ are the abscissa and ordinate vectors of the vehicle's expanded vertexes (points A, B, C, D in Figure 2) for a heading angle of θ.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

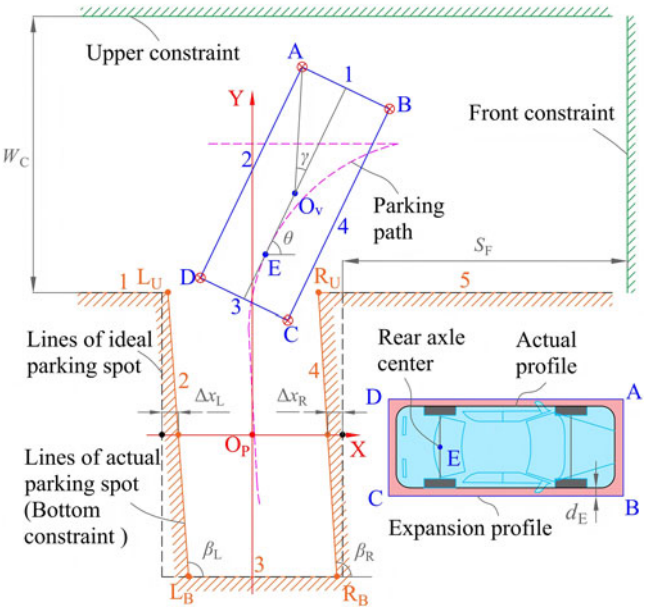


Figure 2. *Actual parking scenario model and safety characterization.*

Then, the minimum distances between the vehicle and the upper, left, and right obstacles in state of S_{OV} are calculated as follows.

$$\begin{cases} d_{\mathrm{U}} = W_{\mathrm{C}} + \dfrac{L_{\mathrm{P}}}{2} - \left[\max\left(\mathbf{y}_{\mathrm{V}}(\theta)\right) + y_{\mathrm{OV}}\right] \\ d_{\mathrm{R}} = \dfrac{W_{\mathrm{P}}}{2} + S_{\mathrm{F}} - \left[\max\left(\mathbf{x}_{\mathrm{V}}(\theta)\right) + x_{\mathrm{OV}}\right] \\ d_{\mathrm{L}} = \min\left(\mathbf{x}_{\mathrm{V}}(\theta)\right) + x_{\mathrm{OV}} - \dfrac{W_{\mathrm{P}}}{2} - S_{\mathrm{F}} \end{cases} \tag{4}$$

where d_U, d_R, and d_L are the minimum distances between the vehicle and the upper, left, and right obstacles, respectively, with negative values indicating a collision.

As illustrated in Figure 2, the bottom constraint consists of five line segments. Collision detection should be performed first, which involves checking for any interference between the four line segments of the vehicle's expanded profile and the five line segments of the obstacles in the bottom constraint. The specific method is as follows.

$$f_{\mathrm{IN}ij} = \begin{cases} 0 & \text{no intersection point} \\ 1 & \text{else} \end{cases}, i = 1, 2, \cdots, 4;\ j = 1, 2, \cdots, 5 \tag{5}$$

where $f_{IN_{ij}}$ is a logical variable indicating whether there is an intersection point between line segment i of the vehicle's expansion profile and the obstacle line segment j. A sum of $\sum f_{IN} = 0$ indicates no collision with the bottom constraint; otherwise, a collision occurs.

When there is no collision, the minimum distance between the vehicle and the bottom constraint is determined by the smallest distance from the four expanded vertices of the ego vehicle to the five obstacle line segments, as well as the distances from the two convex points, L_U and R_U, of the actual parking lot to the four expanded edges of the vehicle. The distance from point L_U to line segment AB is calculated as follows.

$$d_{\mathrm{PL}i} = \begin{cases} \left|\mathrm{L_U L_{U_\perp}}\right| & \angle \mathrm{L_U AB} < \dfrac{\pi}{2} \cap \angle \mathrm{L_U BA} < \frac{\pi}{2} \\ \min\left(\left|\mathrm{L_U A}\right|, \left|\mathrm{L_U B}\right|\right) & \text{else} \end{cases}, \quad i = 1, \cdots, 28 \tag{6}$$

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

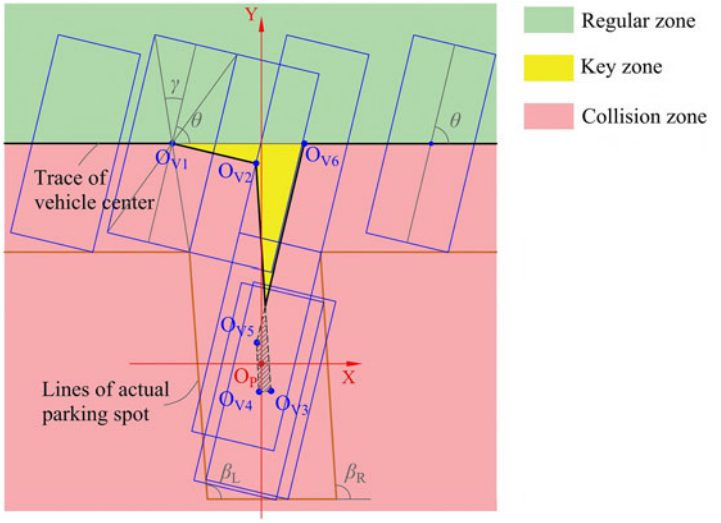


Figure 3. *Partition of SNPS.*

Thus, the minimum distance between the vehicle and the bottom constraint (d_B) is

$$d_{\mathrm{B}} = \begin{cases} \min\{\mathbf{d}_{\mathrm{PL}}\} & \sum f_{\mathrm{IN}} = 0 \\ -0.1 & \sum f_{\mathrm{IN}} \neq 0 \end{cases} \tag{7}$$

2.3. Partition-based calculation method for d_B field

Calculating the minimum DBVO for safety evaluation during the path planning phase can significantly reduce planning efficiency, as the calculation of d_B involves numerous interference detection and distance computations from points to line segments for each node in every path during the search process. To achieve real-time parking path planning, a method based on partition for calculating the d_B field is proposed.

As shown in Figure 3, under a constant heading angle, the trace of the vehicle's geometric center – formed as the vehicle bottom slides along the bottom constraint line – divides the SNPS into three zones: the collision zone, the regular zone, and the key zone. The d_B along the trace is zero. In the regular zone, the increment in d_B equals the increment in the ordinate, as follows:

$$d_{\mathrm{B}}(x, y, \theta) = d_{\mathrm{B}}(x, y_{\mathrm{OV1}}, \theta) + y - y_{\mathrm{OV1}} \tag{8}$$

where (x, y) are the coordinates of the geometric center of the vehicle. y_{OV1} is the vertical coordinate of the vehicle geometric center when its lower-right vertex of the expansion boundary is at the top-left vertex position of the parking space.

Due to the symmetry of the vehicle,

$$d_{\mathrm{B}}(x, y, \theta) = d_{\mathrm{B}}(x, y, \pi + \theta) \tag{9}$$

Therefore, only the d_B in the key zones within the heading angle range of $[0, \pi]$ needs to be calculated to obtain the d_B field of the entire SNPS. As illustrated in Figure 3, the area of the key zones is substantially smaller than that of the entire SNPS, which leads to a marked improvement in computational efficiency.

According to the relationship between the heading angle θ and the three angles of $\pi/2$, β_L, and β_R, there are eight distinct partition situations. As illustrated in Figure 3, which represents the case where $\theta < \pi/2$, $\theta < \beta_L$, and $\theta < \beta_R$, the coordinates of the six critical points (x_{OVi}, y_{OVi}) essential for the partitioning process are derived through geometric analysis.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

$$\begin{cases} x_{\mathrm{OV1}} = x_{\mathrm{LU}} + L\cos(\gamma+\theta), \quad y_{\mathrm{OV1}} = \dfrac{L_{\mathrm{P}}}{2} + L\sin(\gamma+\theta) \\ x_{\mathrm{OV2}} = x_{\mathrm{LU}} + L\cos(\gamma-\theta), \quad y_{\mathrm{OV2}} = \dfrac{L_{\mathrm{P}}}{2} - L\sin(\gamma-\theta) \\ x_{\mathrm{OV3}} = (y_{\mathrm{OV3}} - y_{\mathrm{OV2}})\cot\beta_{\mathrm{L}} + x_{\mathrm{OV2}}, \quad y_{\mathrm{OV3}} = -\dfrac{L_{\mathrm{P}}}{2} + L\sin(\gamma+\theta) \\ x_{\mathrm{OV4}} = x_{\mathrm{RB}} - L_{\mathrm{VE}}\cos\theta - L_{\mathrm{VE}}\sin\theta\cot\beta_{\mathrm{R}} + L\cos(\gamma+\theta), \quad y_{\mathrm{OV4}} = y_{\mathrm{OV3}} \\ x_{\mathrm{OV5}} = (y_{\mathrm{OV5}} - y_{\mathrm{OV4}})\cot\beta_{\mathrm{R}} + x_{\mathrm{OV4}}, \quad y_{\mathrm{OV5}} = \dfrac{L_{\mathrm{P}}}{2} + L\sin(\gamma-\theta) \\ x_{\mathrm{OV6}} = (y_{\mathrm{OV5}} - y_{\mathrm{OV5}})\cot\theta + x_{\mathrm{OV5}}, \quad y_{\mathrm{OV6}} = \dfrac{L_{\mathrm{P}}}{2} + L\sin(\gamma+\theta) \end{cases} \tag{10}$$

where γ is the angle between the diagonal of the expanding vehicle and the vehicle's side. $L = |\mathrm{AC}|/2$.

If the trace formed while sliding along the left side ($OV_1 - OV_2 - OV_3$) is positioned to the right of the trace formed while sliding along the right side ($OV_4 - OV_5 - OV_6$), the enclosed area constitutes the collision zone, as illustrated by the shaded region in Figure 3. The remaining enclosed area is designated as the key zone. Similarly, key zones for the other seven situations can be determined.

In conclusion, the total minimum DBVO is given by:

$$d_{\mathrm{A}} = \min(d_{\mathrm{U}}, d_{\mathrm{B}}, d_{\mathrm{L}} \text{ or } d_{\mathrm{R}}) \tag{11}$$

The calculations for d_U, d_L, and d_R are relatively straightforward. Thus, computing the d_B field prior to path optimization and utilizing it during path search for safety detection can significantly enhance the computational efficiency of path planning.

3. Parking path design for intrusion scenarios based on predefined geometric set

3.1. Design of parking path curve

In traditional path planning, abrupt changes in curvature occur between straight lines and arcs (transitioning from 0 to $\frac{1}{R}$), which complicates the parking process. To address this issue, some researchers have employed clothoids to ensure curvature continuity in the optimal paths derived from combinations of straight lines and arcs, thereby enhancing the quality of parking paths. However, this approach increases the spatial requirements for parking, which poses significant safety challenges in narrow spaces [34]. To overcome these challenges, a Clothoid-Aca-Clothoid (CAC) curve, characterized by curvature values of 0 at both the beginning and end, is proposed as an alternative to circular arcs. This CAC curve is directly applied in path optimization to ensure safer parking in tiny spaces.

Figure 4 presents a schematic illustration of the CAC curve, where δ_{Cm} is the maximum central angle of the clothoid, and δ_T is the central angle of the CAC curve. As depicted in Figure 4 (a), when the center angle of CAC $\delta_T > 2\delta_{Cm}$, the CAC curve comprises two clothoids and an arc segment. Referring to Figure 4(c), it is evident that the curvature of this curve remains continuous, peaking at $\frac{1}{R_{min}}$. However, when $\delta_T \leq 2\delta_{Cm}$, as illustrated in Figure 4 (b), the CAC curve simplifies to just two clothoids. In this case, the maximum curvature of the curve stands at $\sqrt{\delta_T \sigma}$. The detailed design of CAC curve can be found in the appendix.

As demonstrated in Figure 4 (d), the CAC curve, upon undergoing mirroring, translation, and rotation, transforms into the essential components of the parking path. Here, there exist four distinct CAC paths that originate from the initial state $S_0 = [x_0, y_0, \theta_0]$ and share a center angle δ_T. These paths can be unambiguously determined based on the rotation angle ($\delta_R = \theta_0$ or $\theta_0 + \pi$) and the mirror logic variable ($f_M = 1$ or -1), which are determined according to the path function. The parameters of CAC path (x_c, y_c and θ_c) can be derived as follows.

$$\begin{cases} x_{\mathrm{c}} = x_{\mathrm{c0}} \cdot \cos\delta_{\mathrm{R}} - f_{\mathrm{M}} \cdot y_{\mathrm{c0}} \cdot \sin\delta_{\mathrm{R}} + x_0 \\ y_{\mathrm{c}} = x_{\mathrm{c0}} \cdot \sin\delta_{\mathrm{R}} + f_{\mathrm{M}} \cdot y_{\mathrm{c0}} \cdot \cos\delta_{\mathrm{R}} + y_0 \\ \theta_{\mathrm{c}} = f_{\mathrm{M}} \cdot \theta_{\mathrm{c0}} + \theta_0 \end{cases} \tag{12}$$

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

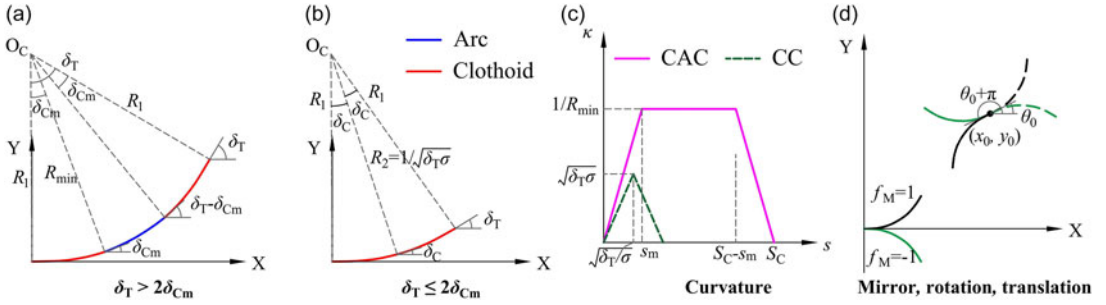


Figure 4. *CAC for parking path.*

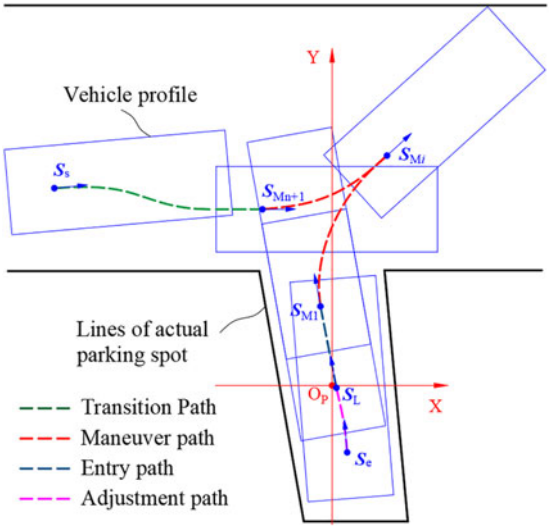


Figure 5. *Four-phase parking path pattern.*

where x_{c0}, y_{c0} and θ_{c0} are the coordinates and heading angle of the original CAC curve, respectively. f_M is a logical variable signifying whether a path needs to be mirrored, with a value of -1 indicating mirroring.

For simplicity in our discussion, the CAC curve obtained through the processes of mirroring, rotation, and translation is denoted as:

$$\text{path_CAC} = f_{\text{CAC}}(\mathbf{S}_0, \delta_{\text{T}}, f_{\text{M}}, \delta_{\text{R}}) \tag{13}$$

3.2. Design of parking path in intrusion scenarios

Due to the intrusion into the parking spot, the current vertical straight parking path presents a significant risk of collision. To address this, a four-phase parking path pattern is created, as illustrated in Figure 5. The entire parking path, extending from the starting state S_s near the target parking spot to FPS S_e, is segmented into four distinct phases: the transition path, maneuver path, entry path, and adjustment path. This path is composed of CAC curves and straight lines. The path can be determined using a reverse design approach, with the specific process detailed as follows.

3.2.1. Adjustment path and entry path

The entry path is a sloping straight line, while the adjustment path is a CAC curve, as depicted in Figure 6. If the FPS of the vehicle is $S_e = [x_e, y_e, \theta_e]$ and the inclination angle of the entry path is θ_L, then the total

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

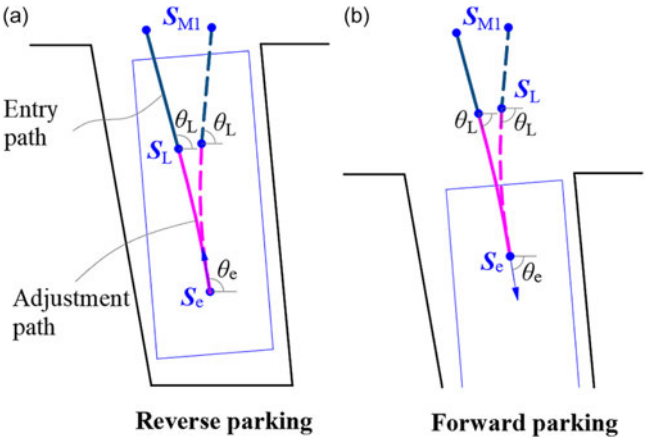


Figure 6. *Four-phase parking path pattern.*

center angle of the CAC curve for adjustment path is $\delta_T = |\theta_L - \theta_e|$. Furthermore, if $\theta_L - \theta_e < 0$, the CAC curve must be mirrored for both forward and reverse parking scenarios.

For reverse parking, the original CAC curve needs to be rotated by θ_e. Consequently, the adjustment path is given by:

$$\text{path_A} = f_{\text{CAC}}\left(\mathbf{S}_{\text{e}}, |\theta_{\text{L}} - \theta_{\text{e}}|, \text{sign}(\theta_{\text{L}} - \theta_{\text{e}}), \theta_{\text{e}}\right) \tag{14}$$

For forward parking, the original CAC curve needs to be rotated by $\theta_e + \pi$, thus:

$$\text{path_A} = f_{\text{CAC}}\left(\mathbf{S}_{\text{e}}, |\theta_{\text{L}} - \theta_{\text{e}}|, \text{sign}(\theta_{\text{L}} - \theta_{\text{e}}), \theta_{\text{e}} + \pi\right) \tag{15}$$

After determining the state adjustment path, the lower node state S_L of the entry path (Path_E) can be calculated. Hence, with the FPS known, the entry path and state adjustment path can be defined by the ordinate y_{M1} of the upper node of the entry path and θ_L.

3.2.2. Maneuver path

The maneuver path consists of CAC curves, with the number of curves determined by the required maneuvers, as shown in Figure 7. The vector representing the center angles of these CAC curves is denoted as $\boldsymbol{\delta} = [\delta_1, \delta_2, \cdots, \delta_n]$. As illustrated in Figure 7 (a) and (c), when the vehicle approaches from the left, the heading angle changes from 0 to θ_L. Conversely, when approaching from the right, the heading angle changes from π to θ_L, as shown in Figure 7 (b) and (d). Thus, the sum of the total center angles of all CAC curves in the maneuver path is

$$\delta_{\text{A}} = \sum_{i=1}^{n} \delta_i = \begin{cases} |\theta_{\text{L}}| & f_{\text{L}} = 1 \\ \pi - |\theta_{\text{L}}| & f_{\text{L}} = -1 \end{cases} \tag{16}$$

where f_L is a logical variable indicating whether the vehicle approaches from the left. In state S_{M1}, the heading angle $\theta_{M1} = \theta_L$. The heading angles of other node states $S_{M2} \sim S_{Mn}$ are given by:

$$\begin{cases} \theta_{\text{M}i} = \theta_{\text{L}} - \sum_{j=1}^{i-1} \delta_j, \quad i = 2, \cdots, n \end{cases} \tag{17}$$

where n is number of maneuvers.

As shown in Figure 7(a) and (b), in the situations of reverse parking, the first maneuver path ($S_{M1} \sim S_{M2}$) when vehicles approach from the left is obtained by rotating the mirrored original CAC by θ_{M1} and then translating it to (x_{M1}, y_{M1}). The second maneuver path ($S_{A2} \sim S_{A3}$) is obtained by rotating the mirrored original CAC by $\theta_{A2} + \pi$ and then translating it to (x_{M2}, y_{M2}). The third section follows the pattern of the first, and the fourth section is similar to the second one. When the vehicle approaches from the right, the rotation angle of the original CAC curve remains the same as when approaching from the

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

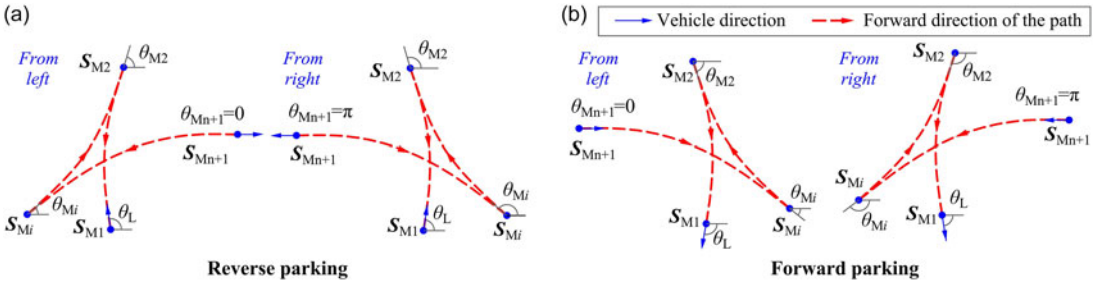


Figure 7. *Design of maneuver path.*

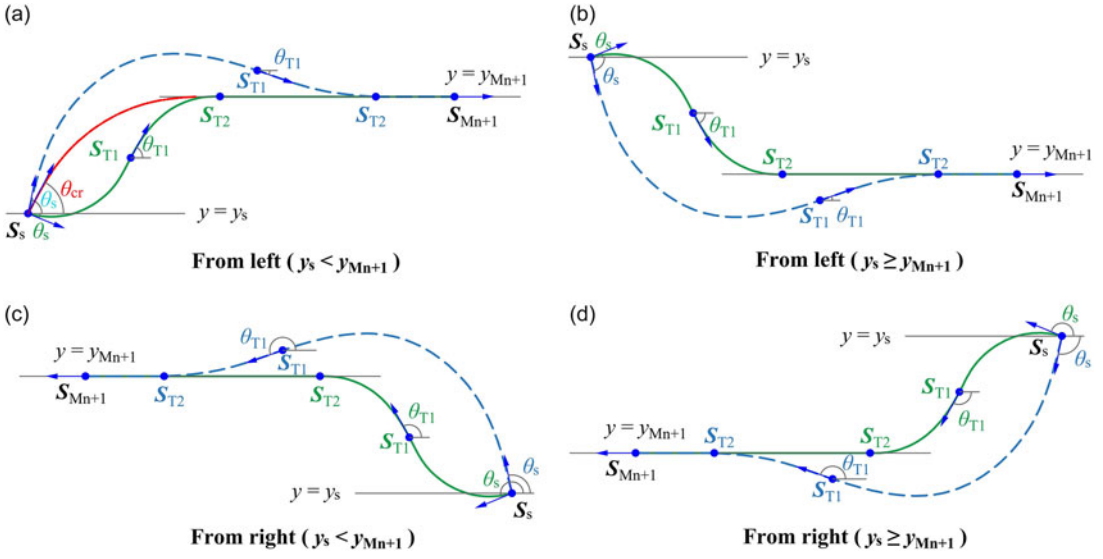


Figure 8. *Design of state path.*

left, but the CVC curves do not require mirroring. In summary, the i-th ($i = 1, \ldots, n$) section of the maneuver path for reverse parking is

$$\text{path_}M_i = f_{\text{CAC}}\left(\mathbf{S}_{M_i}, \delta_i, -f_{\text{L}}, \theta_{M_i} + \left[1 + (-1)^i\right] \cdot \frac{\pi}{2}\right) \tag{18}$$

Similarly, the i-th ($i = 1, \ldots, n$) section of the maneuver path for forward parking can be derived.

$$\text{path_}M_i = f_{\text{CAC}}\left(\mathbf{S}_{M_i}, \delta_i, f_{\text{L}}, \theta_{M_i} + \left[1 + (-1)^{i+1}\right] \cdot \frac{\pi}{2}\right) \tag{19}$$

In the following, we present algorithms for offline planning with lead times in the order of tens of seconds to several minutes, depending on the complexity of the scenario and the desired quality of the solutions, and for online planning with computation times in the order of fractions of a second up to very few seconds. The algorithm for offline planning is based on a global optimization approach and ensures high-quality solutions. The algorithm for online planning uses a sequential optimization procedure that provides flight paths with slightly decreasing quality.

3.2.3. Transition path

As shown in Figure 8, the transition path consists of two CAC curves and a horizontal straight line segment. Its purpose is to move the vehicle from the starting state (S_S) to the starting state of the maneuver path ($S_{M_{n+1}}$). Based on the orientation of the vehicle in S_S and the vertical distance between S_S and $S_{M_{n+1}}$, the transition path can be categorized into four scenarios.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

To demonstrate the design process of the transition path, consider the first scenario depicted in Figure 8(a). If the vertical displacement of the CAC curve, initiated with a heading angle of θ_s and terminated at a heading angle of 0, precisely matches the value of $y_{M_{n+1}} - y_s$, the transition path can be efficiently constructed using a single CAC curve in conjunction with one linear segment. This optimal configuration is illustrated by the red curve in 8(a). In this case, the starting heading angle θ_s is referred to as θ_{cr}:

$$\theta_{\mathrm{cr}} = H_{\mathrm{C0}}^{-1}(y_{\mathrm{M}n+1} - y_{\mathrm{s}}) \tag{20}$$

where $H_{C0}^{-1}(y)$ is the inverse function of $H_C(\delta_T, \delta_R, f_M)$ with $\delta_R = 0$ and $f_M = 1$. $H_C(\delta_T, \delta_R, f_M)$ represents the vertical distance between the two endpoints of the original or mirrored CAC curve, with a center angle of δ_T rotated by δ_R. $H_{C0}(\delta_T) = H(\delta_T, 0, 1)$. In this case, the CAC curve of the transition path is

$$\mathrm{path_}T_{\mathrm{C}} = f_{\mathrm{CAC}}(\mathbf{S}_{\mathrm{s}}, \theta_{\mathrm{s}}, -1, \theta_{\mathrm{s}}) \tag{21}$$

When $\theta_s \neq \theta_{cr}$, the transition path requires two CAC curves ($S_s \rightarrow S_{T1}$ and $S_{T1} \rightarrow S_{T2}$). The key to determining these curves is to find the heading angle θ_{T1} in S_{T1}. If $\theta_s > \theta_{cr}$, the transition path will produce an overshoot as indicated by the blue dashed line in Figure 8(a), and θ_{T1} must satisfy:

$$H_{\mathrm{C}}\left(\left|\theta_{\mathrm{T1}} - \theta_{\mathrm{s}}\right|, \theta_{\mathrm{s}}, -1\right) - H_{\mathrm{C}}\left(-\theta_{\mathrm{T1}}, 0, 1\right) = y_{\mathrm{M}n+1} - y_{\mathrm{s}} \tag{22}$$

If $\theta_s > \theta_{cr}$, the transition path does not overshoot, and θ_{T1} must satisfy the condition specified in Eq. (23). The corresponding transition path is represented by the solid green curve in Figure 8(a).

$$H_{\mathrm{C}}\left(\left|\theta_{\mathrm{T1}} - \theta_{\mathrm{s}}\right|, \theta_{\mathrm{s}}, 1\right) + H_{\mathrm{C}}\left(\theta_{\mathrm{T1}}, 0, 1\right) = y_{\mathrm{M}n+1} - y_{\mathrm{s}} \tag{23}$$

Using Eq. (22) or Eq. (23), θ_{T1} can be determined. Consequently, the two CAC curves for the transition path are

$$\begin{cases} \mathrm{path_}T_{\mathrm{C1}} = f_{\mathrm{CAC}}\left(\mathbf{S}_{\mathrm{s}}, \left|\theta_{\mathrm{T1}} - \theta_{\mathrm{s}}\right|, -1, \theta_{\mathrm{s}}\right) \\ \mathrm{path_}T_{\mathrm{C2}} = f_{\mathrm{CAC}}\left(\mathbf{S}_{\mathrm{T1}}, \theta_{\mathrm{T1}}, 1, \theta_{\mathrm{T1}}\right) \end{cases} \tag{24}$$

For the remaining three scenarios shown in Figure 8(b), (c), and (d), the CAC curves of the transition path can be derived through symmetry. These CAC curves and the horizontal line segment connecting S_{T1} to $S_{M_{n+1}}$ together form the complete transition path (path_T). In summary, given the starting and FPS of the vehicle, the parking path designed using the presented method can be determined by δ, θ_L and y_{M1}.

4. Two-step optimization method of FPS and parking path

During the entry and state adjustment phases, the risk of collision is at its highest, making safety the primary concern. Conversely, in the transition and adjustment phases, efficiency takes precedence. Therefore, this paper proposes a two-step parking path optimization method that optimizes for safety and efficiency as objectives for different optimization steps.

4.1. Framework for two-step parking path optimization method

To enhance the computational efficiency of the two-step parking path optimization method, a framework that integrates both online and offline calculations has been developed, as illustrated in Figure 9. The offline calculations encompass the basic parameters of the CAC curve, the construction of the CAC curve set and its associated functions, the d_B field in an ideal parking scenario, and the critical SNPS required for each parking manner.

The online calculation process begins after parking allocation. If the parking spot is invaded, the dB field is updated accordingly. Subsequently, the first step of optimization is performed. If a solution is found, the process advances to the second step. In the second step, the parking manner is determined

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

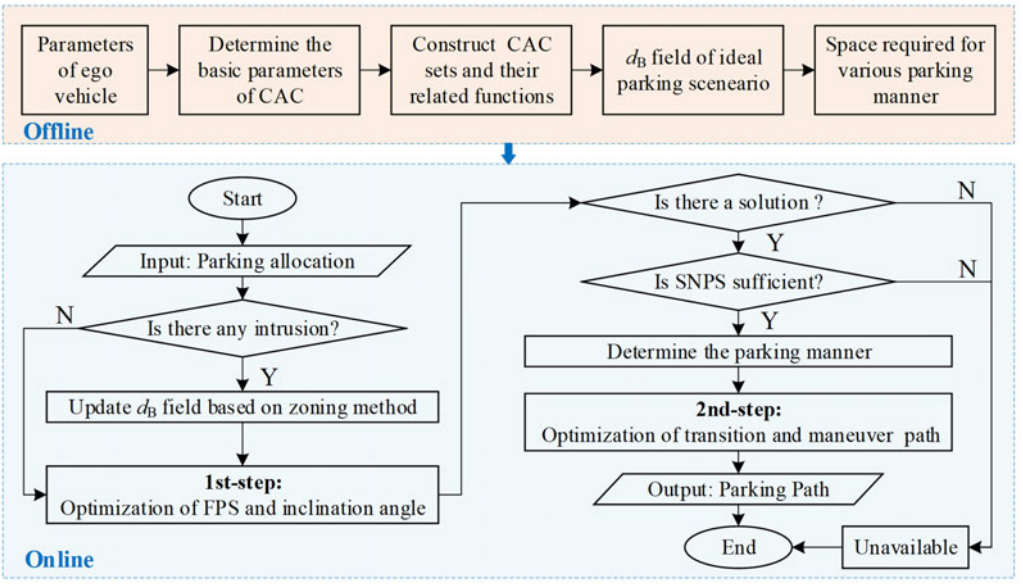


Figure 9. *Framework of the two-step parking path optimization method.*

based on the actual SNPS, followed by the optimization of the transition and maneuver paths. Ant colony algorithms [35] are employed to solve the optimization models.

4.2. Optimization of FPS and inclination angle of entry path

For safety, the minimum DBVO under bottom constraints (d_B) is taken as the index. The first-step optimization is to maximize d_B during the entry and adjustment phases. Accordingly, the objective function is defined as follows:

$$d_{\min 1} = \min \left[d_{\mathrm{B}}(\mathrm{path_E}, \mathrm{path_A}) \right] \tag{25}$$

The optimization parameters include the vehicle FPS and the inclination angle of the entry path. To simplify calculations, the state based on the vehicle's geometric center is adopted. The parameter y_{M1} is preset to determine the entry path for calculating the objective function.

$$\mathbf{X}_1 = [x_{\mathrm{OVe}}, y_{\mathrm{OVe}}, \theta_{\mathrm{e}}, \theta_{\mathrm{L}}] \tag{26}$$

where (x_{OVe}, y_{OVe}) are coordinates of vehicle's geometric center in FPS.

The constraints are outlined as follows. The range for the geometric center of the vehicle in FPS is given by

$$g_1^1(\mathbf{X}_1): \begin{cases} -0.5 \le x_{\mathrm{OVe}} \le 0.5 \\ -0.5 \le y_{\mathrm{OVe}} \le 0.5 \end{cases} \tag{27}$$

This equation expands the allowable range for the geometric center, which may lead to the vehicle not being entirely within the parking spot; therefore, additional constraints are necessary:

$$g_2^1(\mathbf{X}_1): \begin{cases} \max\left(\mathbf{y}_{\mathrm{V}}(\theta_{\mathrm{e}}) + y_{\mathrm{OVe}}\right) \le \frac{L_P}{2} \\ d_{\min 1} \ge 0 \end{cases} \tag{28}$$

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

The directions of the final heading angle and the inclination angle should be between the directions of the two side lines of the actual parking lot. When parking forward, the constraints are defined as

$$g_3^1(\mathbf{X}_1): \begin{cases} -\max(\beta_\mathrm{L}, \beta_\mathrm{R}) \le \theta_\mathrm{e} \le -\min(\beta_\mathrm{L}, \beta_\mathrm{R}) \\ -\max(\beta_\mathrm{L}, \beta_\mathrm{R}) \le \theta_\mathrm{L} \le -\min(\beta_\mathrm{L}, \beta_\mathrm{R}) \end{cases} \tag{29}$$

When parking in reverse, the conditions are

$$g_3^1(\mathbf{X}_1): \begin{cases} \min(\beta_\mathrm{L}, \beta_\mathrm{R}) \le \theta_\mathrm{e} \le \max(\beta_\mathrm{L}, \beta_\mathrm{R}) \\ \min(\beta_\mathrm{L}, \beta_\mathrm{R}) \le \theta_\mathrm{L} \le \max(\beta_\mathrm{L}, \beta_\mathrm{R}) \end{cases} \tag{30}$$

Thus, the optimization model for both FPS and entry path inclination angle is expressed as

$$\begin{aligned} &\max \quad d_{\min 1} \\ &\text{s.t.} \quad g_i^1(\mathbf{X}_1)\ i = 1, 2, 3 \end{aligned} \tag{31}$$

For the convenience of discussion, the optimal FPS based on the rear axle center is denoted as S_{eO}, and the optimal inclination angle of the entry path is denoted as θ_{LO}.

4.3. Optimization of the maneuver and transition path

The maneuver and transition paths are defined by parameter δ and y_{M1}. Therefore, the second step of optimization focuses on finding a set of maneuver path parameters that maximize parking efficiency. The parking efficiency is influenced by both the length of the parking path and the parking manner.

4.3.1. Optimization model of path length

The optimization function is based on the total length of the maneuver and transition path (l_{TM}), the calculation formula is

$$l_\mathrm{TM} = L_\mathrm{C}(|\theta_\mathrm{T1} - \theta_\mathrm{s}|) + L_\mathrm{C}(|\theta_\mathrm{T1}|) + \sum_{i=1}^{n} L_\mathrm{C}(\delta_i) + |x_\mathrm{T2} - x_{\mathrm{M}n+1}| \tag{32}$$

where $L_C(\delta)$ is the length of the CAC curve with a central angle of δ.

The optimization parameters include the ordinate of the starting point of the entry path and the center angle vector of the CAC curves of the maneuver path.

$$\mathbf{X}_2 = [y_\mathrm{M1}, \delta] \tag{33}$$

The range of values for the ordinate y_{M1} is defined as

$$g_1^2(\mathbf{X}_2): \begin{cases} y_\mathrm{LO} \le y_\mathrm{M1} \le W_\mathrm{C} + \frac{L_\mathrm{P}}{2} - l_\mathrm{A} - l_\mathrm{F} - d_\mathrm{E} & \text{Reverse} \\ y_\mathrm{LO} \le y_\mathrm{M1} \le W_\mathrm{C} + \frac{L_\mathrm{P}}{2} - l_\mathrm{R} - d_\mathrm{E} & \text{Forward} \end{cases} \tag{34}$$

where y_{LO} is the starting point ordinate of the adjustment path obtained from the first-step optimization. l_F is length of vehicle front overhang.

The constraint for the central angle vector δ is given by

$$g_2^2(\mathbf{X}_2): \begin{cases} \sum_{i=1}^{n} \delta_i = \delta_\mathrm{A} \\ 0 < \delta_i \le \delta_\mathrm{A}\ i = 1, 2, \cdots, n \end{cases} \tag{35}$$

The vehicle's FPS and the entry path inclination angle are set based on the results obtained from the first step of optimization.

$$g_3^2(\mathbf{X}_2): \begin{cases} \mathbf{S}_\mathrm{e} = \mathbf{S}_\mathrm{eO} \\ \theta_\mathrm{L} = \theta_\mathrm{LO} \end{cases} \tag{36}$$

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

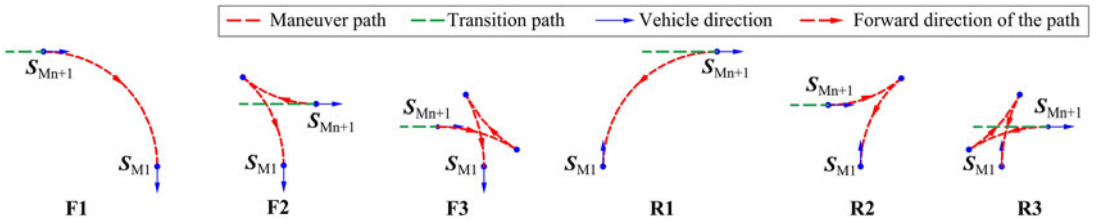


Figure 10. *Maneuver path under different maneuvering times.*

To ensure non-collision, the minimum DBVO from the starting state to state S_{M1} must be at least 0. Thus

$$g_4^2(\mathbf{X}_2): d_{\min 2} = \min\left[d_{\mathrm{A}}(\text{path_T}, \text{path_M})\right] \geq 0 \tag{37}$$

Therefore, the optimization model for the path length is expressed as

$$\begin{aligned} &\min \quad l_{\mathrm{TM}} \\ &\text{s.t.} \quad g_i^2(\mathbf{X}_2)\ i = 1, 2, 3, 4 \end{aligned} \tag{38}$$

4.3.2. Optimization of parking manners

Parking manners are influenced by the parking direction and the number of maneuvers required. As shown in Figure 10, forward parking with one maneuver (F1) does not involve any reversing, while forward parking with two or three maneuvers (F2 or F3) necessitates reversing the vehicle direction twice. In contrast, reverse parking with one or two maneuvers (R1 or R2) requires reversing only once, while reverse parking with three maneuvers (R3) requires reversing three times. Each reversal requires deceleration and acceleration, which not only decreases parking efficiency but also reduces passenger comfort. Additionally, compared to R1 and F2, R2 and F3 each involve one additional directional change operation, making vehicle control more challenging. Therefore, to minimize the number of reversals and directional changes, the priority order of parking manners, assuming no constraints on SNPS, is as follows: F1, R1, R2, F2, F3, and R3.

In actual parking scenarios, SNPS is limited, so parking manners must be optimized within SNPS constraints. Thus, it is necessary to analyze the SNPS requirements for various parking manners in intrusion scenarios. A multi-objective optimization model is established with the aim of minimizing both corridor width and front space, as represented by Eq. (39). By solving the Pareto optimal solution for this model, the critical SNPS required for each parking manner can be determined. Combining this with the priority order allows for the identification of the optimal parking manner.

$$\begin{aligned} &\min \quad f_{\mathrm{ob}}(\mathbf{X}_2) = [W_{\mathrm{C}}, S_{\mathrm{F}}] \\ &\text{s.t.} \quad \begin{cases} g_i^2(\mathbf{X}_2)\ i = 1, 2, 3, 4 \\ \mathbf{S}_{\mathrm{e}} = \mathbf{S}_{\mathrm{eO}}, \theta_{\mathrm{L}} = \theta_{\mathrm{LO}} \\ \mathbf{S}_{\mathrm{s}} = [x_{\mathrm{s}}, y_{\mathrm{s}}, 0] \end{cases} \end{aligned} \tag{39}$$

where

$$\begin{cases} W_{\mathrm{C}} = \max\left(\mathbf{y}_{\mathrm{V}}\right) - \dfrac{L_{\mathrm{P}}}{2} \\ S_{\mathrm{F}} = \begin{cases} \max\left(\mathbf{x}_{\mathrm{V}}\right) - \dfrac{W_{\mathrm{P}}}{2} & \text{From left} \\ -\dfrac{W_{\mathrm{P}}}{2} - \min\left(\mathbf{x}_{\mathrm{V}}\right) & \text{From right} \end{cases} \\ x_{\mathrm{s}} = \begin{cases} 10 & \text{From left} \\ -10 & \text{From right} \end{cases} \\ y_{\mathrm{s}} = 5 \end{cases} \tag{40}$$

where (x_s, y_s) is the starting position.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

Table I. *Geometric structure parameters of vehicles and parking lot parameters.*

Parameters	Values
length of vehicle front overhang, l_F (m)	0.96
vehicle wheelbase, l_A (m)	2.84
length of vehicle rear overhang, l_R (m)	0.96
width of vehicle, W_V (m)	1.76
length of vehicle, L_V (m)	4.76
maximum steering angle of front wheel, φ_{max} (°)	35
maximum speed of front wheel steering, ω_{max} (rad/s)	0.25
length of parking spot, L_P (m)	5.5
width of parking spot, W_P (m)	2.5

Table II. *The range and discrete interval of the intrusion scene representation parameters.*

Parameters	Min	Max	Interval
Δx_O(m)	−0.6	0.6	0.2
$\Delta\beta_L$ (°)	−10	10	2
$\Delta\beta_R$ (°)	−10	10	2

5. Results and discussion

Using the dimensions of typical vehicles and parking spaces as a reference, the parameters are detailed in Table I.

$$\sigma = \frac{\omega}{v \cdot \varphi_{\max} \cdot R_{\min}} \tag{41}$$

According to Eq. (46), the minimum turning radius R_{min} for the given vehicle is 4.06m. Additionally, a clothoid curve is generated when the vehicle's speed remains constant while the steering angle of the front wheels changes uniformly. The rate of curvature change for the clothoid is given by Eq. (41), where v is velocity of parking. Setting $\omega = \omega_{\max}$, the value of σ varies from 1 to 0.05 as v ranges from 0.1m/s to 2m/s, based on the given vehicle parameters. Considering the required space and parking efficiency, a value of $\sigma = 0.32$ is selected.

5.1. Result of d_B field

According to the statistical data from actual parking lots, the distribution range of parameters $[\Delta x_O, \Delta\beta_L, \Delta\beta_R]$ that characterize parking scenarios is shown in Table II. Through discretization for dB field calculation and analysis, 847 parking scenarios were generated. The simulation experiments were conducted on a computer with the following configuration: Intel (R) Core (TM) i9-10940X CPU @ 3.30 GHz, 16.0 GB RAM, and 12-thread parallel computing capabilities.

Figure 11 displays the calculation time for the d_B field, with an ideal parking scenario taking 8.73s. Additionally, there are 84 parking scenarios without intrusion, for which the dB field of the ideal scenario can be used for path optimization. The remaining scenarios involve intrusions, and the average calculation time for their dB fields is 9.08s, with the maximum time not exceeding 25s. Given that the time for a vehicle to travel from the parking space entrance to a nearby point in the target parking spot is longer than the dB field calculation time, the d_B field can be computed and updated online during this period.

Figure 12 illustrates the dB field for both ideal and intrusive scenarios ($[\Delta x_O, \Delta\beta_L, \Delta\beta_R] = [0\ 10\ 6]$). In the entry and adjustment phases (when θ is around 90°), the non-collision zone in the invaded scenario

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

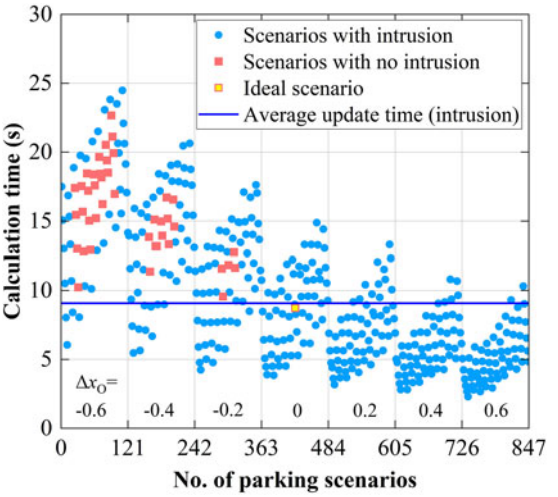


Figure 11. *Calculation time for dB field.*

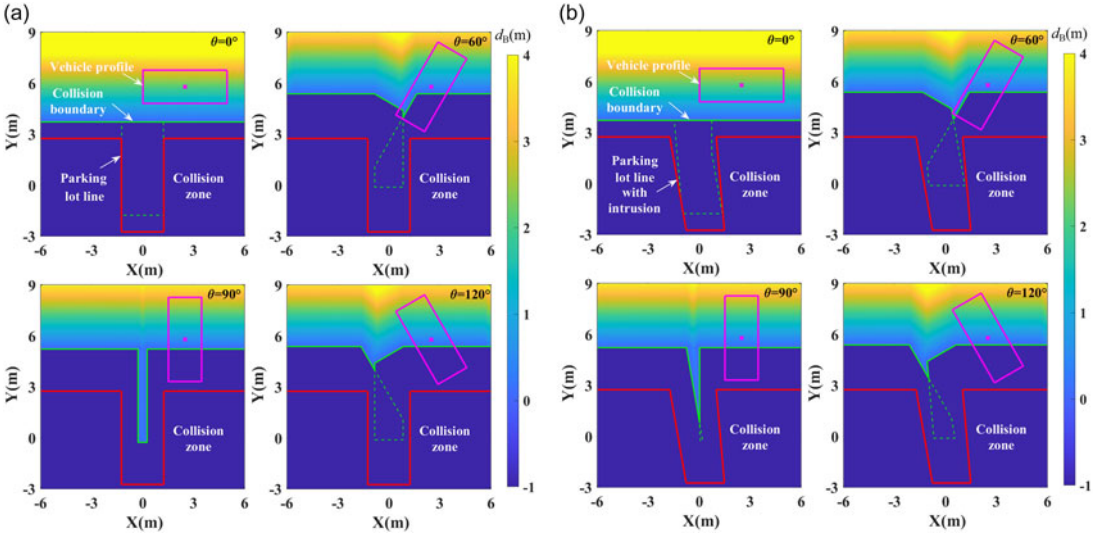


Figure 12. *Result of d_B field. (a) Ideal parking scenarios; (b) Invaded parking scenarios.*

is significantly reduced and distorted compared to the ideal scenario, which severely limits parking path options. Although the impact of intrusion on the non-collision zone is relatively minor during the transition and maneuver phases (when θ outside the 90° neighborhood), the constraints on the entry path substantially affect the optimization of the transition and maneuver paths.

5.2. Optimization results of FPS

Figure 13 illustrates the minimum DBVO for both optimal and normal FPS across 847 parking scenarios. In the figure, a minimum DBVO value below zero indicates the occurrence of a collision event. As the offset distance Δx_O increases, the number of scenarios meeting the non-collision condition decreases for both FPS, and no scenarios meet the condition when Δx_O reaches 0.6m. Additionally, only 36.2% of the scenarios are viable with normal FPS, whereas the proportion increases to 60.1% with optimal FPS.

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

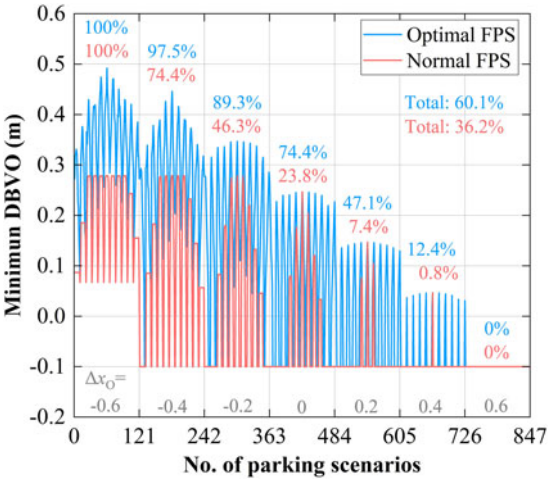


Figure 13. *Minimum DBVO for optimal and normal FPS.*

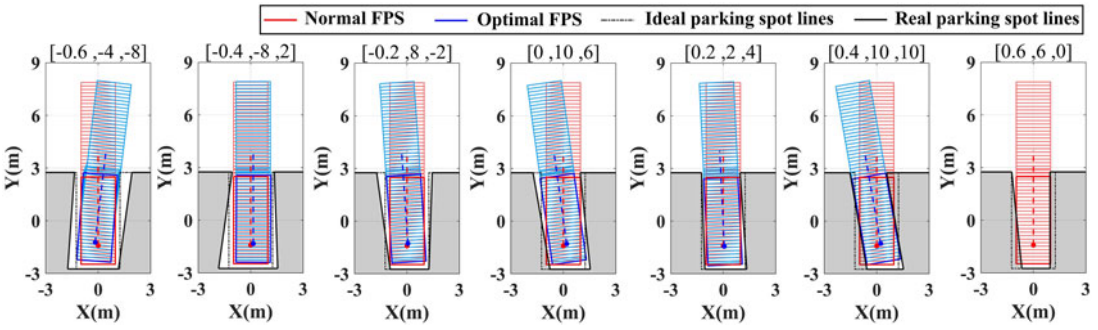


Figure 14. *Adjustment and entry path for optimal and normal FPS.*

Figure 14 shows the adjustment and entry paths for optimal and normal FPS across 7 parking scenarios. In the first three scenarios ($[-0.6, -4, -8]$, $[-0.4, -8, 2]$, $[-0.2, 8, -2]$), although there are no collisions adopting normal FPS, the minimum DBVO values are quite low at 0.20m, 0.09m, and 0.05m, respectively. After optimization of the FPS, the minimum DBVO values increased to 0.39m, 0.20m, and 0.15m, respectively. Furthermore, scenarios $[0, 10, 6]$, $[0.2, 2, 4]$, and $[0.4, 10, 10]$ are not available with normal FPS but achieve a minimum DBVO greater than 0.05m with FPS optimization. For scenario $[0.6, 6, 0]$, even with FPS optimization, it remains unavailable.

Overall, FPS optimization increases the number of available parking scenarios by 65.8%. The minimum DBVO for scenarios available with normal FPS improves by an average of 0.12m following FPS optimization, demonstrating the effectiveness of the proposed method for first-step optimization.

5.3. Optimization results of transition and maneuver path

5.3.1. SNPS requirements for parking manners

In an ideal parking scenario, the SNPS requirements for driving from the left and right are consistent due to symmetry. Figure 15 displays the Pareto front for the corridor width and front space required for each parking manner in the ideal parking scenario, obtained by the optimization model shown in Eq. (39).

For reverse parking, the minimum corridor width (W_C) required for R1 is 5.11 m, with a minimum front space (S_F) of 7.10 m. For R2, the minimum W_C remains the same as R1, but the required S_F is smaller. When the W_C is sufficient, the minimum S_F required for R2 is 1.06 m. The minimum W_C for

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

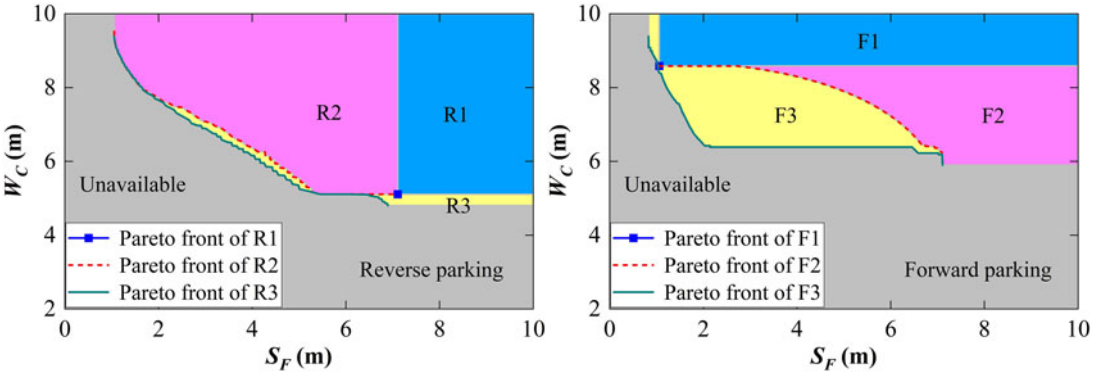


Figure 15. *Critical SNPS requirements for parking manner in ideal parking scenarios.*

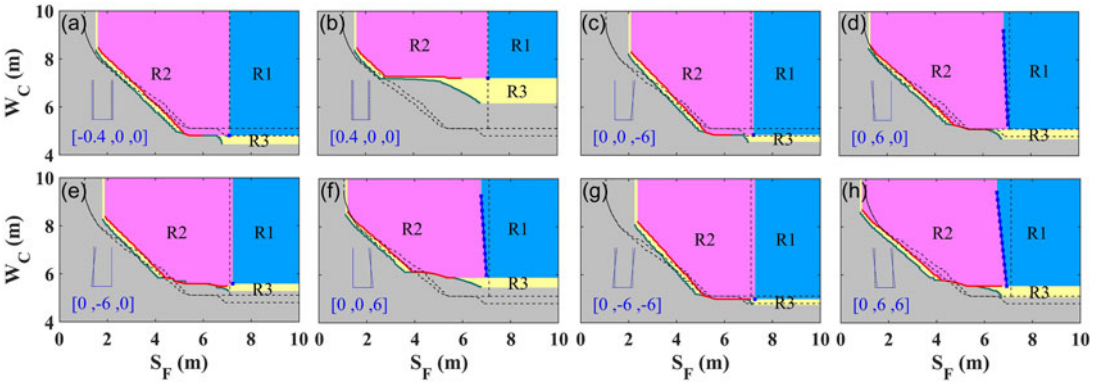


Figure 16. *Impact of intrusion on the SNPS requirements.*

R3 is less than that for R1 and R2, at only 4.78 m, corresponding to a minimum S_F of 6.89m. When the S_F is small (< 5.3m), the minimum W_C required for R3 is slightly lower than that for R2. For forward parking, the minimum W_C required for F1 is 8.59 m. The minimum W_C required for both F2 and F3 is 5.90 m, corresponding to an S_F of 7.10 m. When S_F is less than 7.10m, the minimum W_C required for F3 is significantly lower than that for F2. Overall, forward parking requires a higher W_C compared to reverse parking.

According to the Pareto front and the priority order of parking manners, the SNPS parameter space can be divided into four regions. For the same parking direction, the lower left area of the R3(F3) Pareto front is the unavailable region, where parking requirements cannot be met with fewer than three maneuvers. The upper right area of the R1(F1) Pareto front is the preferred region for R1(F1). The area to the left of the R1(F1) Pareto front and to the upper right of the R2(F2) Pareto front is the preferred region for R2(F2). The remaining area is the preferred region for R3(F3).

Using the example of driving from the left and parking in reverse, the impact of intrusion on the SNPS requirements is analyzed. The results are depicted in Figure 16, where the black dashed line represents the ideal scenario. As shown in Figure 16(a) and (b), $\Delta x_O = -0.4$ m, the parking spot becomes wider, and the required minimum W_C slightly decreases by 0.29 m. Conversely, when $\Delta x_O = 0.4$ m, the parking spot narrows causing the required minimum W_C to increase sharply by approximately 2.10 m. Figure 16 (c) and (f) illustrate that with $\Delta x_O = 0$ m, if the right line of the parking spot is shifted clockwise or the left line is shifted counter-clockwise, the parking spot's opening widens, leading to a decrease in the required minimum W_C. Conversely, if the parking spot's opening narrows, the required minimum W_C increases. Moreover, the direction of the parking spot's opening affects the magnitude of this change. Specifically, for this example, rightward deviation reduces the required corridor width. As shown in

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

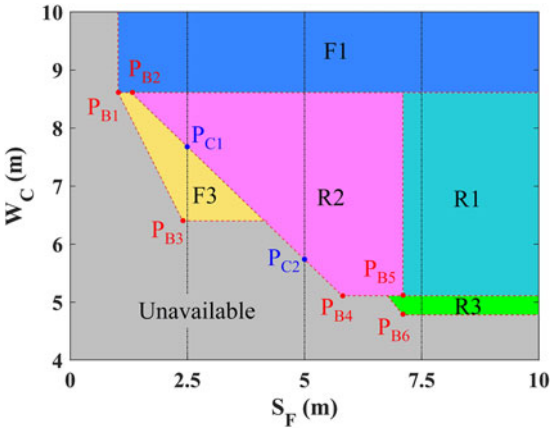


Figure 17. *Preferred region for various parking manners.*

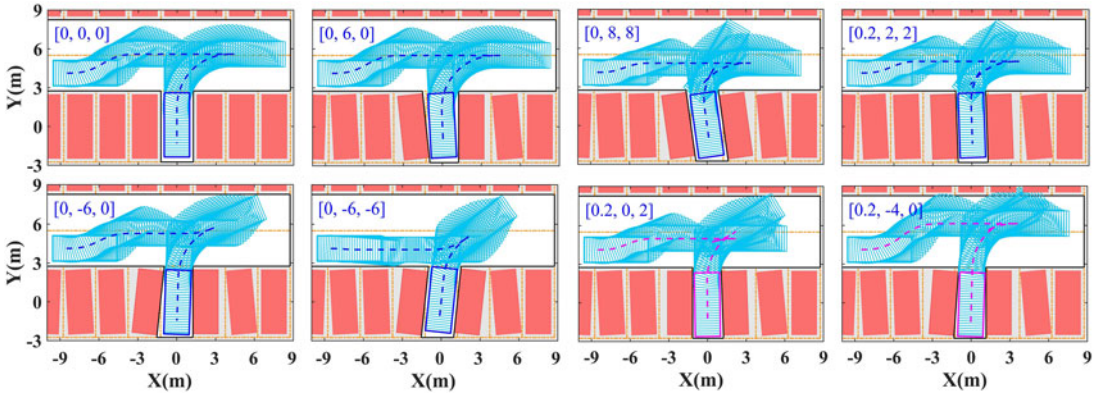


Figure 18. *Impact of intrusion on maneuver and transition paths.*

Figure 16 (g) and (h), an overall tilt of the parking spot increases the required minimum W_C; however, a tilt orientation opposite to the vehicle's approach direction (in this case, rightward tilt) will mitigate this increase and may even reduce the required W_C.

5.3.2. Optimal results of parking manner in intrusion scenarios

Figure 17 depicts the preferred regions for six commonly used parking manners. Although intrusion affects the partitioning to some extent, the overall distribution pattern remains unchanged.

For practical use, the partition boundaries have been simplified. These boundaries are defined by six points, PB1 through PB6, which can be calculated offline. The coordinates of the boundary points for the ideal parking scenario are provided in Table III.

The front space is typically determined by the number of parking spots in front of the target spot. When there is one parking spot in front, if $W_C \in [y_{PB3}, y_{PC1}]$, F3 is preferred; if $W_C \in [y_{PC1}, y_{PB2}]$, R2 is preferred; and if $W_C > y_{PB2}$, F1 is preferred. This pattern is applicable to other front space conditions as well.

Taking the SNPS with $S_F = 7.5$ m and $W_C = 5.5$ m as an example, the impact of intrusion on maneuver and transition paths was analyzed, with the results shown in Figure 18.

The vehicle's starting states were set to $[-8.5, 4.125, 0]$, and the FPS were set to the optimal FPS. In the ideal parking scenario, R1 is applicable. Due to a reduced minimum W_C in parking scenarios $[0, 6, 0]$, R1 remains applicable. However, in scenarios $[0, -6, 0]$ and $[0, -6, -6]$, the minimum SF for R1 slightly increases, making R2 the preferred parking manner. In contrast, scenarios $[0, 8, 8]$ and

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

Table III. *Boundary points for the ideal parking scenarios.*

Boundary points	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**
x_{PB} (m)	1.02	1.30	2.40	5.80	7.10	7.10
y_{PB} (m)	8.61	8.61	6.40	5.11	5.11	4.78

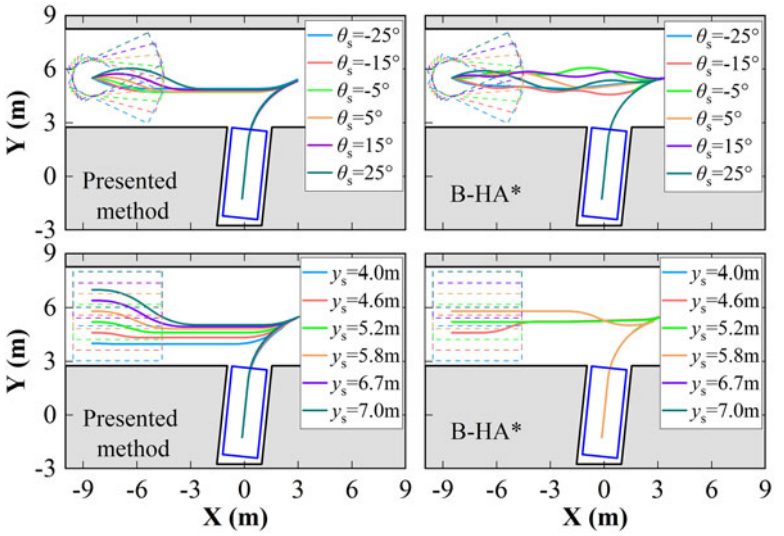


Figure 19. *Impact of intrusion on maneuver and transition paths.*

[0.2, 2, 2] significantly increase the minimum W_C for both R1 and R2, so R3 is the only feasible option. For scenarios [0.2, 0, 2] and [0.2, −4, 0], no parking manner meets the SNPS constraints.

5.4. Performance validation of presented methods

To assess the effectiveness of the presented method, comparative experiments were conducted using the hybrid A-star algorithm with bidirectional search (B-HA*) [30]. Key parameters of B-HA* were configured as follows: Only two path curvatures were allowed: 0 (straight-line segment) and $1/R_{min}$ (corresponding to the minimum turning radius); The search step length was set to 0.05m. It is important to note that the Voronoi potential field $\rho_V(x, y)$, employed for safety evaluation in B-HA*, requires updating for parking scenarios involving different levels of intrusion.

5.4.1. Adaptability to starting state

Consider the parking scenario with a SNPS of $W_C = 5.5$ m and $S_F = 7.5$ m, specifically [0, −6, −6]. The impact of the starting state on the parking path was analyzed, with results shown in Figure 19. When the starting point is at (− 8.5 m, 5.5 m), both the presented method and B-HA* can generate an effective parking path within the starting heading angle (θ_s) range of [− 25°, 25°]. However, the path generated by B-HA* exhibits more fluctuation. For $\theta_s = 0$ and $x_s = -8.5$m, the presented method can deliver an effective parking path within the y_s range of [4 m, 7 m]. In contrast, B-HA* struggles to find an effective path when the starting point is near the parking lot boundaries.

5.4.2. Adaptability to intrusion parking scenarios

Using the starting state [− 8.5, 4.3, 0] as an example, the adaptability of the proposed method and B-HA* to intrusion parking scenarios with $\Delta x_O = 0$ was analyzed. The results for two different SNPSs

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

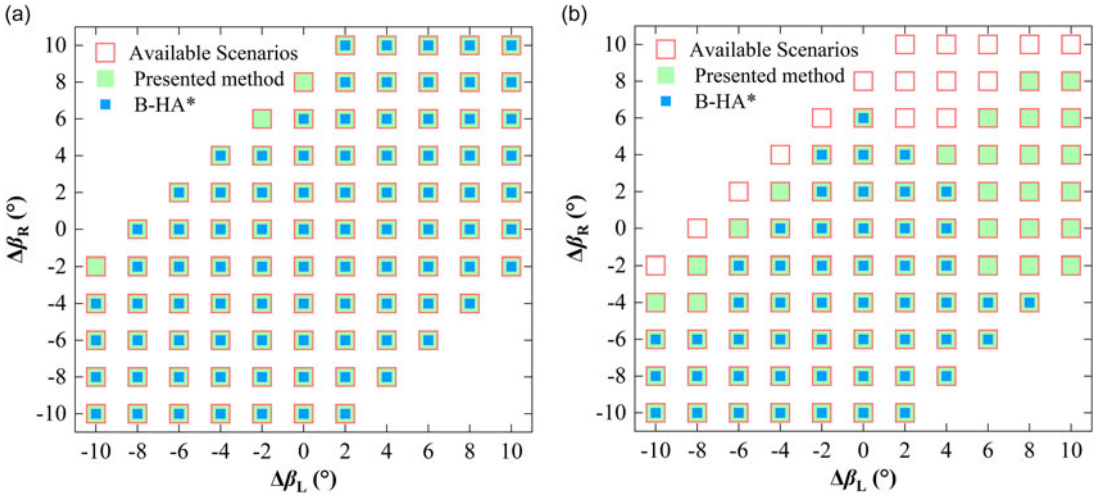


Figure 20. *Adaptability to intrusion parking scenarios. (a) SNPS1; (b) SNPS2.*

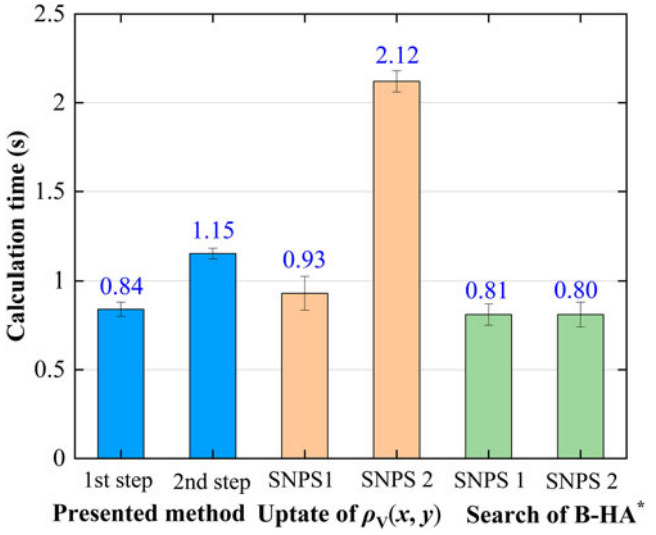


Figure 21. *Calculation time for presented method and B-HA*.*

(SNPS1: $W_C = 7$ m, $S_F = 10$ m; SNPS 2: $W_C = 5.5$ m, $S_F = 7.5$ m) are presented in Figure 20. As shown in Figure 20(a), when the SNPS is sufficiently large, both the presented method and B-HA* can generate effective parking paths for all intrusion scenarios. However, when the SNPL is relatively narrow, as depicted in Figure 20(b), B-HA* is able to handle only 50 scenarios, whereas the presented method can address 73 scenarios. This indicates that the presented method can increase the number of available intrusion parking scenarios by 46%.

5.4.3. Calculation time

Using the intrusion parking scenarios detailed in Section 5.4.2, the calculation time of the presented method was analyzed statistically, with the results displayed in Figure 21. The average optimization time per step for the presented method is 0.84s and 1.15s, leading to an overall average time of 1.99s. For B-HA*, the average update times for the Voronoi potential field under SNPS1 and SNPS2 constraints are 0.93 and 2.12 s, respectively. These fields are pre-computable before path optimization. The average

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

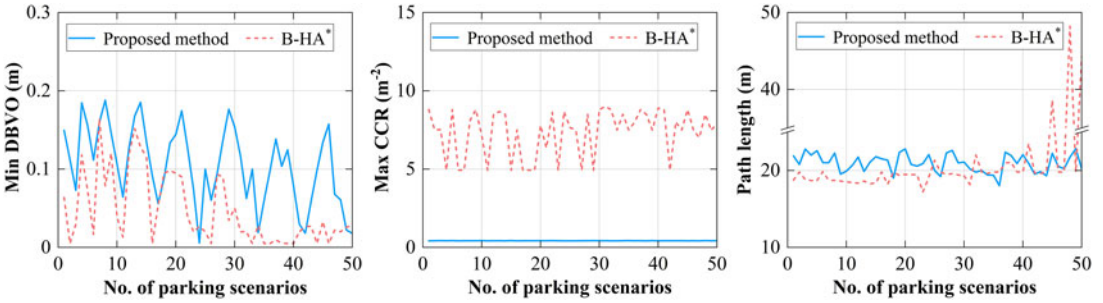


Figure 22. *Quality of parking path.*

search times for B-HA* under both SNPS1 and SNPS2 constraints are approximately 0.80s. Given that B-HA* requires the first step of optimization to determine the optimal FPS, the average total time for B-HA* is 1.64s, slightly lower than that of the presented method. Both methods' computation times are adequate to meet real-time engineering requirements.

5.4.4. Quality of parking path

Using publicly available intrusion parking scenarios under the SNPS2 constraint as outlined in section 5.4.2, the quality of parking paths generated by the proposed method and B-HA* based on minimum DBVO, path length, and maximum curvature change rate (CCR) was evaluated, with the results illustrated in Figure 22. The presented method consistently achieves a higher minimum safety distance compared to B-HA*, with an average improvement of 0.06m. In terms of path length, the presented method generates longer paths in the first 40 scenarios, averaging 1.69m more than those produced by B-HA*. Conversely, in the final 10 scenarios, it produces shorter paths, averaging 6.57m less. Additionally, the maximum curvature change of the proposed method's paths is approximately 0.32, significantly lower than that of B-HA*, indicating that the presented method yields paths with superior smoothness and curvature continuity, making them easier to track and control.

Utilizing kinematics-based PID and Stanley algorithms [36] for path tracking control, we examine the 20th scenario [0, −2, −6] as an illustrative example. The results are presented in Figure 23, where the maximum acceleration is set at 1m/s^2, while the reference speed is set at 0.5 m/s.

As illustrated in Figure 23(c) – (f), the B-HA* encounters significant difficulties in path tracking. This is evidenced by pronounced fluctuations in the control variable of steering angle, with maximum lateral error and heading angle error reaching 0.27 m and 16.3°, respectively, which far exceed acceptable thresholds. Additionally, the final heading angle error reaches 4.0°, and these substantial deviations result in the tracking path failing to satisfy the non-collision requirement, as illustrated in Figure 23(a). In contrast, the proposed method exhibits relatively smooth control variables for acceleration and steering angle. The maximum lateral error and heading angle error are 0.08 m and 4.3°, respectively. In the FPS, the lateral error and heading angle error are reduced to 0.01 m and 0.8° respectively. As illustrated in Figure 23(b), these errors remain within acceptable limits and do not breach non-collision constraints, demonstrating the proposed method's effective feasibility for path planning in non-ideal scenarios.

Overall, the simulation results demonstrate that the presented method not only expands the range of usable intrusion parking scenarios but also enhances the success rate of effective path generation while improving path quality.

6. Conclusions

To clarify the impact of intrusion into the target parking spot caused by abnormal parking of adjacent vehicles on the parking of the ego vehicle, this article conducted a quantitative analysis. The findings revealed that normal FPS can be available to only 36.2% of intrusion scenarios, but this comes at the cost

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

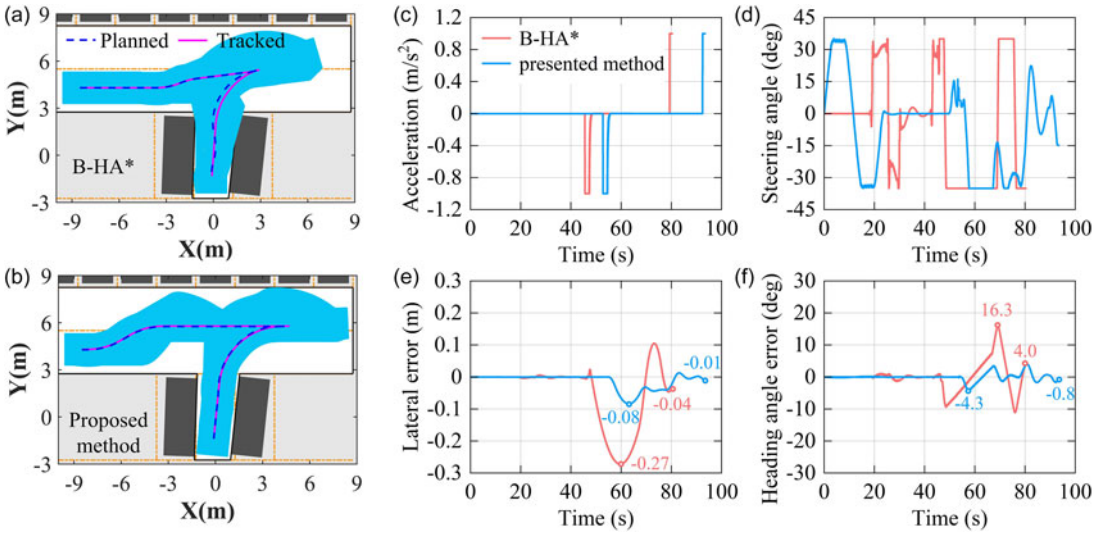


Figure 23. *Traceability performance of paths. (a) Planned and tracked path of B-HA*; (b) Planned and tracked path of presented method; (c) Acceleration; (d) Steering angle; (e) lateral error; (f) Heading angle error.*

of significantly reduced safety. Furthermore, narrow SNPS constraints will further decrease this proportion. To address these issues, this paper introduced a two-step parking path optimization method based on the minimum DBVO field and a predefined geometric set. This approach successfully increases the proportion of usable intrusion scenarios and enhances the quality of parking paths, including path length, minimum DBVO, maximum CCR, and traceability. While the proposed methodology has demonstrated remarkable efficacy in generating parking paths under intrusion scenarios, its current implementation has not incorporated dynamic vehicle interactions, unstructured environments, or real-world trajectory tracking control validation. These limitations will constitute the primary focus of subsequent research endeavors.

Author contributions. Xiaoxiao Lv completed the main research work and analysis, including problem analysis, model establishment, programming solution, verification, and initial manuscript writing; Wenrui Jin proposed research questions and provided financial support and guidance on the research process; Xiangping Qiu provided a verification plan and specific images of parking scenarios; Fan Mo provided guidance on the structure of the paper and revised the manuscript. Jiaxue Li undertook some of the data analysis and chart drawing work, as well as manuscript revisions; Min Fang undertook some of the data analysis and chart drawing work.

Financial support. This work was supported by the Interdisciplinary Joint Research Project of Tongji University in 2023 (grant no. 2023-2-ZD-09).

Competing interests. The authors have no financial or proprietary interests in any material discussed in this article.

Ethical approval. None.

References

[1] L. Cai, H. Guan, Z. Y. Zhou, F. L. Xu, X. Jia and J. Zhan, "Parking planning under limited parking corridor space," *IEEE Trans. Intell. Transp. Syst.* **24**(2), 1962–1981 (2023).

[2] G. Chen, Z. Gao, H. Hu, J. Du and H. Wang, "Multi-Maneuver vertical parking trajectory planning and tracking control in narrow environments," *Automot. Innov.* **7**(2), 300–311 (2024).

[3] T. Rajabioun and P. Ioannou, "On-street and off-street parking availability prediction using multivariate spatiotemporal models," *IEEE Trans. Intell. Transp. Syst.* **16**(5), 2913–2924 (2015).

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

[4] M. Kim, J. Ahn and J. Park, "TargetTree-RRT*: Continuous-Curvature path planning algorithm for autonomous parking in complex environments," *IEEE Trans. Autom. Sci. Eng.* **21**(1), 606–617 (2024).
[5] L. Cai, H. Guan, H. L. Zhang, X. Jia and J. Zhan, "Multi-Maneuver vertical parking path planning and control in a narrow space," *Rob. Auton. Syst.* **149**, 103964 (2022).
[6] I. Han, "Geometric path plans for perpendicular/Parallel reverse parking in a narrow parking spot with surrounding space," *Vehicles* **4**(4), 1195–1208 (2022).
[7] H. Vorobieva, S. Glaser, N. Minoiu-Enache and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. Intell. Transp. Syst.* **16**(1), 396–410 (2015).
[8] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Rob. Res.* **29**(5), 485–501 (2010).
[9] J. Lian, W. Ren, D. Yang, L. Li and F. Yu, "Trajectory planning for autonomous valet parking in narrow environments with enhanced hybrid A* search and nonlinear optimization," *IEEE Trans. Intell. Veh.* **8**(6), 3723–3734 (2023).
[10] T. C. Meng, T. H. Yang, J. Huang, W. R. Jin, W. Zhang, Y. F. Jia, K. Q. Wan, G. Xiao, D. G. Yang and Z. H. Zhong, "Improved hybrid a - star algorithm for path planning in autonomous parking system based on multi-stage dynamic optimization," *Int. J. Automot. Technol.* **24**(2), 459–468 (2023).
[11] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.* **17**(5), 1105–1118 (2009).
[12] L. Han, Q. H. Do and S. Mita, "Unified Path Planner for Parking an Autonomous Vehicle based on RRT," **In:** *2011 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2011).
[13] K. Bergman, O. Ljungqvist and D. Axehill, "Improved path planning by tightly combining lattice-based path planning and optimal control," *IEEE Trans. Intell. Veh.* **6**(1), 57–66 (2021).
[14] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Trans. Intell. Transp. Syst.* **23**(8), 11970–11981 (2022).
[15] J. Wang, J. Li, J. Yang, X. Meng and T. Fu, "Automatic parking trajectory planning based on random sampling and nonlinear optimization," *J. Frankl. Inst.* **360**(12), 9579–9601 (2023). https://doi.org/10.1016/j.jfranklin.2023.06.037
[16] J. Yang, J. Wang, J. Li, X. Meng, X. Jiang and C. Lu, "Sobol Sequence RRT* and Numerical Optimal Joint Algorithm-Based Automatic Parking Trajectory Planning of Four-Wheel Steering Vehicles," *Robot. Auton. Syst.* **186**(4), 104909 (2025). https://doi.org/10.1016/j.robot.2024.104909
[17] K. Li, X. Li, Y. Wu, Z. Deng, Y. Wang, Y. Meng, B. Li, X. Su, L. Wang and X. Wang, "Autonomous dispatch trajectory planning of carrier-based vehicles: An iterative safe dispatch corridor framework," *Def. Technol.* **56**, 83–95 (2026).
[18] S. Khan and J. Guivant, "Design and implementation of proximal planning and control of an unmanned ground vehicle to operate in dynamic environments," *IEEE Trans. Intell. Veh.* **8**(2), 1787–1799 (2023).
[19] L. L. Mero, D. Yi, M. Dianati and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.* **23**(9), 14128–14147 (2022).
[20] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Veh.* **8**(1), 673–683 (2023).
[21] X. Zhang, C. Zhao, F. Liao, X. Li and Y. Du, "Online parking assignment in an environment of partially connected vehicles: A multi-agent deep reinforcement learning approach," *Transp. Res. Part C Emerg. Technol.* **138**(5), 103624 (2022). Elsevier Ltd.
[22] P. Zhang, L. Xiong, Z. Yu, P. Fang, S. Yan, J. Yao and Y. Zhou, "Reinforcement learning-based end-to-end parking for automatic parking system," *Sensors (Switzerland)* **19**(18), 3996 (2019).
[23] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.* **35**(4), 5778–5792 (2024).
[24] S. Y. Song, H. Chen, H. W. Sun and M. C. Liu, "Data efficient reinforcement learning for integrated lateral planning and control in automated parking system," *Sensors* **20**(24), 7297 (2020).
[25] J. Bernhard, R. Gieselmann, K. Esterle and A. Knol, "Experience-Based Heuristic Search: Robust Motion Planning with Deep Q-Learning," **In:** *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, 2018 - November (2018) pp. 3175–3182 (IEEE).
[26] J. Zhang, H. Chen, S. Song and F. Hu, "Reinforcement learning-based motion planning for automatic parking system," *IEEE Access* **8**(2020), 154485–154501 (2020).
[27] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia and S. Chai, "Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver," *IEEE Trans. Autom. Sci. Eng.* **20**(3), 1633–1647 (2023).
[28] R. Q. Chai, A. Tsourdos, A. Savvaris, S. C. Chai, Y. Q. Xia and C. L. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Trans. Neural Netw. Learn. Syst.* **33**(4), 1400–1413 (2022).
[29] X. Tang, Y. Yang, T. Liu, X. Lin, K. Yang and S. Li, "Path planning and tracking control for parking via soft actor-critic under non-ideal scenarios," *IEEE/CAA J. Autom. Sin.* **11**(1), 181–195 (2024).
[30] W. Jin, J. Li, X. Lv and T. Zhang, "An optimized path planning approach for automatic parking using hybrid A* bidirectional search," *Appl. Intell.* **55**(2), 1–19 (2025).
[31] J. Lian, W. Ren, D. Yang, L. Li and F. Yu, "Trajectory planning for autonomous valet parking in narrow environments with enhanced hybrid A * search and nonlinear optimization," *IEEE Trans. Intell. Veh.* **8**(6), 3723–3734 (2023).
[32] S. Li and J. Wang, "Parallel parking path planning in narrow space based on a three-stage curve interpolation method," *IEEE Access* **11**(8), 93841–93851 (2023).

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press

[33] B. Li, T. Acarman, X. Peng, Y. Zhang, X. Bian and Q. Kong, "Maneuver Planning for Automatic Parking with Safe Travel Corridors: A Numerical Optimal Control Approach," **In:** *Eur. Control Conf. 2020, ECC 2020, (C), 1993–1998* (2020, IEEE).
[34] Y. Wu, X. Li, J. Gao and X. Yang, "Research on automatic vertical parking path-planning algorithms for narrow parking spaces," *Electron* **12**(20), 1–20 (2023).
[35] G. Han, Z. Zhou, T. Zhang, H. Wang, L. Liu, Y. Peng and M. Guizani, "Ant-colony-based complete-coverage path-planning algorithm for underwater gliders in Ocean Areas with thermoclines," *IEEE Trans. Veh. Technol.* **69**(8), 8959–8971 (2020).
[36] R. Halaly and E. E. Tsur, "Autonomous driving controllers with neuromorphic spiking neural networks," *Front. Neurorobot.* **17**(2023), 1234962 (2023).

Appendix. Detailed design of CAC curve

Clothoid is a curve whose curvature is proportional to its length. A clothoid with an initial curvature of 0 and a CCR of σ can be described by

$$\kappa(s) = \sigma s \tag{42}$$

where κ and s are the curvature and length of the clothoid, respectively.

A clothoid that starts at the origin and ends with a curvature of $\frac{1}{R}$ can be calculated by

$$\begin{cases} x_{\mathrm{CL}} = \sum_{i=0}^{N} (-1)^i \dfrac{\sigma^{2i} s^{4i+1}}{2^{2i}(4i+1)(2i)!} \\ y_{\mathrm{CL}} = \sum_{i=0}^{N} (-1)^i \dfrac{\sigma^{2i+1} s^{4i+3}}{2^{2i+1}(4i+3)(2i+1)!} \\ \theta_{\mathrm{CL}} = \frac{\sigma s^2}{2} \end{cases}, \quad s \in \left[0, \frac{1}{\sigma R}\right] \tag{43}$$

where x_{CL} and y_{CL} are the abscissa and ordinate of the clothoid point, respectively. θ_{CL} is the heading angle of the corresponding vehicle position. For accurate results, a value of $N = 2$ is sufficient.

To fully leverage the characteristics of the ego vehicle, the minimum turning radius is employed in the clothoid design. Consequently, the center angle (the difference in heading angle between the end and start points, δ_{Cm}) and total length (s_m) of the clothoid are calculated as follows.

$$\delta_{\mathrm{Cm}} = \frac{1}{2\sigma R_{\mathrm{min}}^2} \tag{44}$$

$$s_{\mathrm{m}} = \frac{1}{\sigma R_{\mathrm{min}}} \tag{45}$$

$$R_{\mathrm{min}} = \left| \frac{l_{\mathrm{A}}}{\tan \varphi_{\mathrm{max}}} \right| \tag{46}$$

where R_{min} is the minimum turning radius. φ_{max} is the maximum equivalent turning angle of the front wheel.

When the center angle of CAC $\delta_T > 2\delta_{Cm}$, the CAC curve comprises two clothoids and an arc segment. The overall length of such a CAC curve is given by:

$$S_{\mathrm{C}} = 2s_{\mathrm{m}} + R_{\mathrm{min}}(\delta_{\mathrm{T}} - 2\delta_{\mathrm{Cm}}) \tag{47}$$

However, when $\delta_T \leq 2\delta_{Cm}$, the CAC curve simplifies to just two clothoids. In this case, the maximum curvature of the curve stands at $\sqrt{\delta_T \sigma}$, and its total length calculates to:

$$S_{\mathrm{C}} = 2\sqrt{\frac{\delta_{\mathrm{T}}}{\sigma}} \tag{48}$$

Cite this article: X. Lv, W. Jin, J. Li, X. Qiu, F. Mo, M. Fang, Q. Yu, Y. Yu and J. Yu (2026). "Two-step path planning for vertical parking in narrow non-ideal scenarios based on predefined path pattern and numerical optimization", Robotica **44**, 883–907. https://doi.org/10.1017/S0263574726103324

https://doi.org/10.1017/S0263574726103324 Published online by Cambridge University Press