World Electric Vehicle Journal

Article

Adaptive Multi-Mode Path Planning for Four-Wheel Independent Steering Vehicles

Jiawu Zhu [1], Gang Li [1,*], Ning Li [1] and Dong Zhang [2]

[1] School of Automobile and Traffic Engineering, Liaoning University of Technology, Jinzhou 121001, China; 18488016865@163.com (J.Z.); l_ning1979@126.com (N.L.)
[2] College of Engineering, Design and Physical Sciences, Brunel University of London, London UB8 3PH, UK; dong.zhang@brunel.ac.uk
* Correspondence: qcxyligang@lnut.edu.cn

Abstract

This study proposes an adaptive multi-mode graph search algorithm that integrates spatial previewing with terminal analytics to address node proliferation and terminal oscillation in path planning for four-wheel independent steering (4WIS) vehicles under complex, low-speed conditions. By employing line-of-sight checking and the Douglas–Peucker algorithm to extract the environmental topological skeleton, the proposed method generates Predictive Spatial Profiling (PSP) fields that precisely quantify channel safety margins. Departing from conventional soft-weight arbitration, a dynamic driving state machine leverages these rigid spatial constraints to deterministically prune redundant expansion branches—including Ackermann steering, crab steering, and in-place rotation—prior to node generation. Furthermore, a comprehensive cost function incorporating a mode-switching penalty and a gradient-heading heuristic is formulated to accelerate search convergence. To circumvent reliance on traditional empirical distance thresholds, a topology-triggered, multi-dimensional terminal analytical strategy is introduced, enabling a seamless transition from discrete search node expansion to continuous curve generation near the target. Extensive simulations demonstrate that the proposed algorithm reduces both the node expansion scale and optimization time by over 80% compared with conventional unconstrained methods, while effectively mitigating chaotic motion-mode transitions. Ultimately, integrating environmental spatial dimensionality reduction with terminal analytics yields a highly efficient and smooth global path-planning solution for 4WIS vehicles.

Keywords: four-wheel independent steering; path planning; graph search algorithm; predictive spatial profile; terminal analytics

Academic Editor: Michael Fowler

Received: 13 May 2026
Revised: 24 June 2026
Accepted: 26 June 2026
Published: 28 June 2026

1. Introduction

Driven by advancements in electronic and control systems, four-wheel independent steering (4WIS) vehicles now feature the independent articulation of all four wheels. Unlike conventional front-wheel-steering vehicles, this architecture provides exceptional maneuverability, enabling motion modes such as crab steering and zero-radius in-place rotation that remain unattainable for standard platforms [1,2]. Despite these capabilities, prevailing research on 4WIS intelligent driving technologies has predominantly focused on motion control [3–7]; consequently, path-planning methodologies remain comparatively underexplored. A primary challenge in 4WIS path planning lies in strategically deploying these diverse motion modes across varied operating conditions to maximize chassis maneuverability without compromising trajectory quality.

Historically, path-planning research has centered on graph-search or sampling-based algorithms designed to satisfy nonholonomic constraints. The Dijkstra algorithm [8] established the theoretical bedrock for graph-based planning. Subsequently, the A^* algorithm [9] leveraged heuristic functions to substantially enhance search efficiency. Following this, the Hybrid A^* algorithm [10] integrated vehicle kinematic constraints into the node expansion process. This integration demonstrated robust optimization in structured environments, enabling applications such as automated parking [11]. Among sampling-based approaches, Yang et al. [12] proposed a framework that, by merging with Reeds–Shepp curves, facilitated 4WIS automatic parking. However, in narrow or topologically intricate scenarios, relying solely on standard Ackermann steering frequently fails to produce physically viable paths. This limitation renders the search process susceptible to prolonged computational times or local deadlocks. To address this, Yan et al. [13] markedly improved planning efficiency and traversability by extracting environmental skeleton contours for partitioned planning. Nevertheless, these foundational studies remain largely confined to single-mode kinematics. Recognizing this gap, recent scholarship has shifted toward planning frameworks that integrate multiple motion modes with optimized node-expansion paradigms.

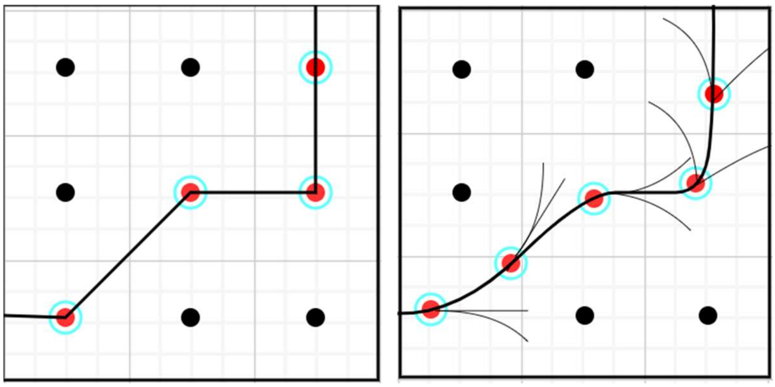
To accommodate the multi-mode nature of 4WIS vehicles, Qin et al. [14] introduced a graph-search algorithm featuring integrated mode decision-making. This method simultaneously evaluates Ackermann steering, crab steering, and in-place rotation during node expansion, governing mode transitions via a multi-objective cost function. Chu et al. [15] advanced this approach by decoupling multi-mode maneuvers into lateral displacement and yaw motion, utilizing a multi-mode risk field to optimize trajectory flexibility and stability. Additional efforts include Zhang et al.'s [16] Hybrid A*-based multi-mode planner and Liu et al.'s [17] collaborative path-planning strategy for four-wheel-steering agricultural robots navigating curved segments. Despite these conceptual advances, soft guidance strategies that rely on cost-function weights exhibit significant limitations in practical applications. First, the absence of a priori environmental geometric assessments causes these algorithms to execute numerous redundant, high-dimensional node explorations in clustered-obstacle environments, severely degrading search efficiency. Second, soft constraints frequently fail to enforce decisive mode arbitration at local extrema, resulting in chaotic mode oscillations along the planned trajectory. Finally, bounded by discrete expansion step sizes, conventional graph searches often suffer from tentative terminal oscillations near the target pose, which impede precise convergence [18,19].

To surmount these limitations, this paper proposes an adaptive multi-mode graph-search algorithm that synthesizes spatial previewing with terminal analytics. Unlike conventional integrations of existing geometric and heuristic modules, this framework introduces a fundamental methodological shift: transitioning from conventional, unconstrained soft-weight arbitration to deterministic, hard-geometric search-space reduction. By employing line-of-sight checking [20] and the Douglas–Peucker algorithm [21], the algorithm extracts the environmental topological skeleton to construct a Predictive Spatial Profiling (PSP) physical field. This PSP field directly quantifies channel safety margins, serving as a spatial map that structurally dictates a dynamic driving state machine. Prior to node expansion, this state machine adaptively prunes redundant motion-mode branches based on the local environmental geometry. To preserve global directionality and mitigate heading misalignment, the proposed method deploys a comprehensive cost function featuring a gradient-heading heuristic. Furthermore, a topology-triggered, multi-dimensional terminal analytical strategy is engineered to eradicate the terminal oscillations native to discrete graph searches. This strategy synergizes the unique crab steering and in-place rotation capabilities of 4WIS vehicles with Reeds–Shepp curves [22] to bypass empirical distance-threshold triggers, facilitating a direct and seamless transition from discrete search exploration to continuous multi-mode analytical curve generation. Ultimately, through these integrated mechanisms, this study fundamentally optimizes search efficiency and trajectory smoothness for 4WIS vehicle path planning in highly confined spatial domains.

2. Multi-Mode Node Expansion and Kinematics Construction

2.1. Multi-Mode Node Expansion

Graph-search algorithms generally navigate discrete maps through node expansion. They reconstruct the optimal trajectory via backtracking upon reaching the target node. As illustrated in Figure 1a, the standard A* algorithm restricts expanded nodes to grid-cell centers. Consequently, this restriction yields paths that violate fundamental vehicle kinematic constraints. Conversely, Figure 1b depicts the node-expansion mechanism of the conventional Hybrid A* algorithm. Although this approach integrates kinematic constraints during expansion to ensure physical feasibility, it remains explicitly tailored for standard front-wheel-steering vehicles [23].



(**a**) (**b**)

Figure 1. Node expansion methods of classical graph search algorithms. (**a**) Node expansion method of the A* algorithm; (**b**) Node expansion of the Hybrid A* algorithm. In both panels, the black dots represent the grid-cell centers, the red circles denote the expanded nodes, and the solid lines indicate the generated paths.

In contrast, 4WIS vehicles exhibit diverse and highly flexible motion modes. This study focuses on three representative maneuvers: four-wheel Ackermann steering, crab steering, and in-place rotation. Ackermann steering acts as the primary operational mode. Conversely, crab steering and in-place rotation serve as specialized high-maneuverability modes. To fully leverage these capabilities, the node-expansion framework must strictly incorporate the kinematic constraints of all three modes.

2.2. Multi-Mode Kinematic Modeling of 4WIS Vehicles

A multi-mode discrete kinematic model is established for 4WIS vehicles. The vehicle platform is assumed to feature a Four-Wheel Independent Drive and Four-Wheel Independent Steering (4WID-4WIS) configuration. This architecture ensures the physical feasibility of all designated motion modes, particularly zero-radius in-place rotation. Furthermore, the drivetrain guarantees independent wheel torque control. Such control constitutes a fundamental mechanical prerequisite for executing the centrally symmetric steering and differential driving demanded during in-place rotation. Let the global two-dimensional Cartesian coordinate system be XOY. The 4WIS vehicle is simplified as a rigid-body model, and its state vector in the global coordinate system is denoted as $\boldsymbol{s}_k = [x_k, y_k, \theta_k]^{\mathrm{T}}$, where (x, y) denotes the coordinates of the vehicle's geometric center, and θ denotes the vehicle body yaw angle. Let the distance from the vehicle's center of mass to the front axle be

L_{f}, and the distance to the rear axle be L_{r}, then the equivalent wheelbase of the vehicle is $L = L_{\mathrm{f}} + L_{\mathrm{r}}$. The complex continuous motions of the vehicle are distilled into the three most representative core motion primitives: four-wheel Ackermann steering, crab steering, and in-place rotation [24,25].

The four-wheel Ackermann steering mode (AS) is illustrated in Figure 2. In this mode, the front and rear wheels are steered in opposite directions at equal magnitudes, yielding $\delta_{\mathrm{r}} = -\delta_{\mathrm{f}}$, thereby achieving the minimum turning radius.

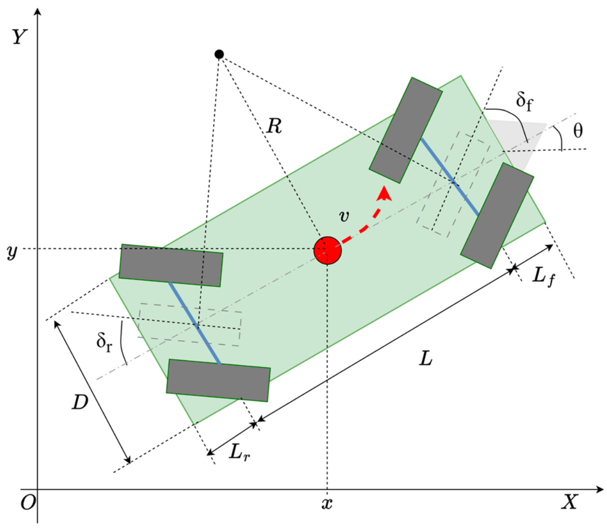


Figure 2. Schematic diagram of the four-wheel Ackermann steering mode. The red dot represents the geometric center of the vehicle, the red arrow indicates the velocity vector, and the dashed lines illustrate the geometric relationships of the turning radius and the instantaneous center of rotation.

The corresponding kinematic differential equation is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \frac{2v\tan\delta}{L} \end{bmatrix} \tag{1}$$

Let the fixed search step length be Δs, the current node state be $\boldsymbol{s}_k$, and the next node be $\boldsymbol{s}_{k+1}$; then, the discrete update equation is formulated as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \Delta s\cos\theta_k \\ \Delta s\sin\theta_k \\ \frac{2\Delta s\tan\delta}{L} \end{bmatrix} \tag{2}$$

where Δs is the fixed search step length, and δ represents the equivalent front-wheel steering angle (i.e., $\delta = \delta_f = -\delta_r$) used for node expansion. According to the geometric relationship of rigid-body rotation, the turning radius is governed by $R = \frac{L}{2\tan\delta}$. Consequently, when the steering angle reaches its mechanical upper limit δ_{max}, the minimum turning radius of the vehicle center is explicitly formulated as

$$R_{min} = \frac{L}{2\tan\delta_{max}} \tag{3}$$

When the vehicle drives straight, yielding $\delta = 0$, the update equation degenerates into straight-line motion along the current heading.

The crab steering mode (CR) is illustrated in Figure 3. All four wheels are steered in the same direction at equal angles, yielding $\delta = \delta_{\mathrm{r}} = \delta_{\mathrm{f}}$. Under this mode, the vehicle body yaw angle θ remains constant, and the vehicle's center of mass translates laterally as a whole along the resultant direction of the yaw angle and the wheel steering angle.

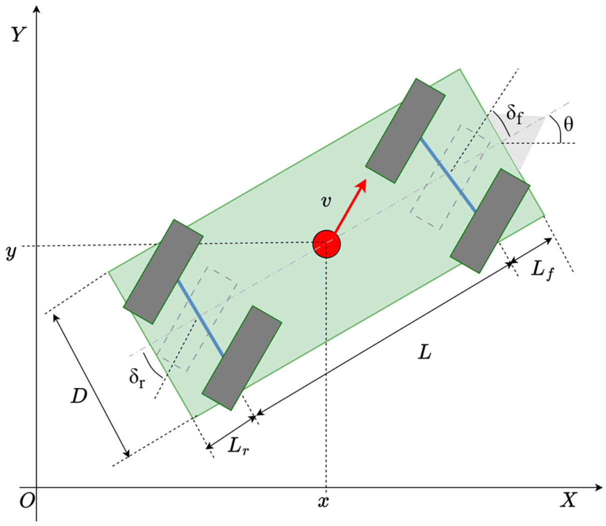


Figure 3. Schematic diagram of the crab steering mode. The red arrow indicates the translational velocity vector of the vehicle.

Its discrete update equation is expressed as

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta_k + \delta) \\ \Delta s \sin(\theta_k + \delta) \\ 0 \end{bmatrix} \tag{4}$$

Figure 4 illustrates the in-place rotation (SP) mode. The front and rear wheels adopt a centrally symmetric steering configuration. Concurrently, the left and right driving wheels execute counter-directional rotation. This precise synchronization compels the vehicle to rotate strictly around its geometric center. During this maneuver, the vehicle's spatial coordinates remain constant. Consequently, only the yaw angle θ changes.

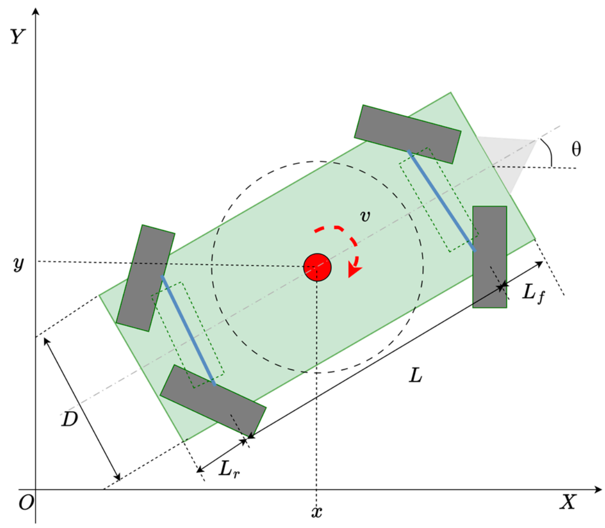


Figure 4. Schematic diagram of the in-place rotation mode. The red circular arrows indicate the yaw rotation around the geometric center, and the dashed circles represent the wheel trajectories.

Let the single-step steering angle step length be $\Delta\theta$; then, its state transition equation simplifies to

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k + \Delta\theta \end{bmatrix} \tag{5}$$

3. Environmental Topology and Predictive Spatial Profiling Construction

3.1. Extraction of Environmental Topological Skeleton Based on LOS-DP

During the initial phase of global planning, the target pose functions as the optimization starting point. From this pose, the Dijkstra algorithm constructs a distance potential field featuring a global gradient across the two-dimensional grid space. The negative gradient of this cost field inherently directs toward the target. Consequently, it provides robust heuristic guidance for the multi-mode graph search detailed in Section 4. Originating from the start position, a discrete greedy search traverses along this negative gradient to extract the theoretical shortest grid path. However, the discrete spatial characteristics of the grid map strictly constrain this initial descent trajectory. As a result, the generated path frequently exhibits an irregular zigzag pattern laden with numerous redundant local turning points [26,27].

To address this limitation, this study introduces a fusion approach combining line-of-sight (LoS) checking and the Douglas–Peucker (DP) algorithm to topologically refine the initial path [28]. As Figure 5 illustrates, the topology-extraction mechanism of this approach operates across two distinct levels. First, the red dashed lines and collision markers depict the greedy linearization process of the LoS algorithm. Specifically, the algorithm attempts to establish a direct connection between the starting point and a distant node. As indicated by the red cross markers, detecting obstacle interference triggers the strict retention of necessary intermediate turning points. This critical action preserves an obstacle-avoiding topology. Furthermore, the black dashed lines and orthogonal segments demonstrate the adaptive thinning principle of the DP algorithm. This phase utilizes the chord formed by two adjacent retained nodes as a reference baseline. It then evaluates the orthogonal distance d from an original local turning point to this chord. If d exceeds the preset thinning threshold e, the algorithm secures that specific feature node as a topological skeleton point. Ultimately, this dual filtering mechanism effectively condenses the high-density discrete grid points. It yields a concise set of topological skeleton points accurately capturing environmental connectivity.

3.2. Construction of the Predictive Spatial Profiling (PSP) Physical Field

Upon extracting the refined topological skeleton, the algorithm must quantify the surrounding physical geometric constraints. Consequently, it constructs a PSP physical field. This field functions as a comprehensive preview of environmental safety margins along the topological trajectory.

3.2.1. Lateral Orthogonal Profiling Mechanism

First, the algorithm computes the local tangent and normal vectors for each parameterized sampling point along the topological skeleton sequence. Subsequently, a spatial boundary-tracing operator is deployed to accurately quantify the lateral physical passage clearance under real-world conditions. This operator executes a bidirectional extension along the normal direction. Finally, Figure 6 illustrates the measurement principle of this operator.

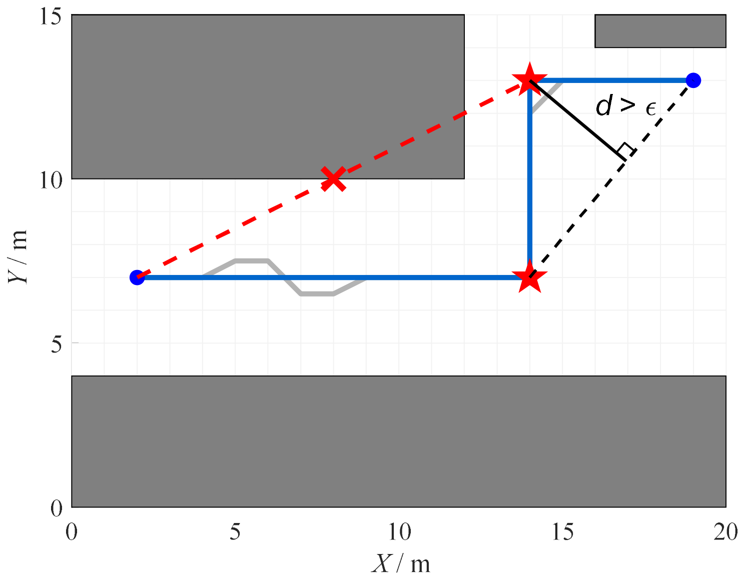


Figure 5. Schematic diagram of the topological skeleton based on the LoS-DP fusion algorithm.

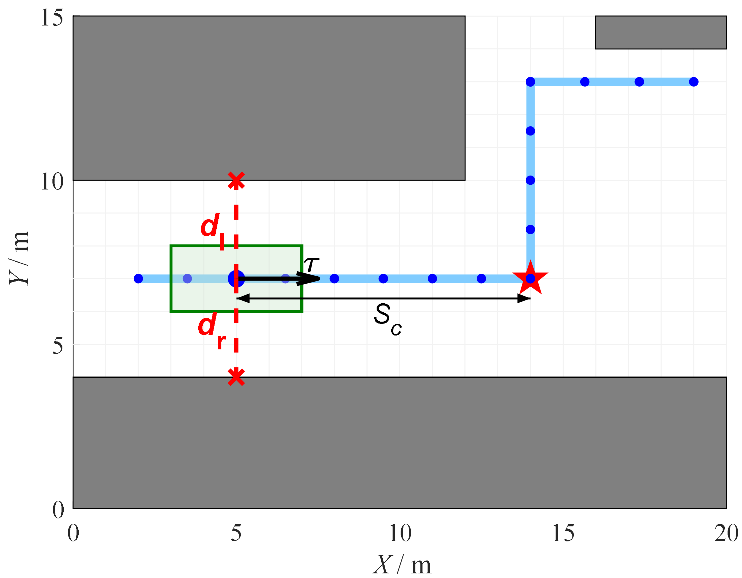


Figure 6. Schematic diagram of the lateral orthogonal profiling mechanism.

During the numerical evaluation, the algorithm initially establishes a safety inflation boundary completely enveloping the vehicle's geometric contour. This study uniformly sets the physical safety margin to 0.1 m for both theoretical derivations and subsequent simulations. Subsequently, the spatial boundary-tracing operator calculates the Euclidean distances, d_l and d_r, to the obstacle inflation boundaries along the left and right normal directions. These calculations determine the absolute physical channel width at that specific node as $W = d_\mathrm{l} + d_\mathrm{r}$.

During the subsequent heuristic expansion phase, the search algorithm leverages this width information to evaluate, in real time, whether the local region satisfies the passage criteria for Ackermann steering. The algorithm employs an absolute prohibition strategy for conventional zones to strictly prune redundant action spaces. Consequently, it forces the activation of high-maneuverability escape modes—such as crab steering and in-place rotation—only when spatially dictated.

3.2.2. Feature Tuple Definition and Inverse Projection

In addition to the passable width W, the algorithm defines the path distance S_c from the current point to the nearest upcoming turning point along the skeleton sequence. Ultimately, it extracts a spatial feature tuple (W, S_c) comprising both the passable width and the preview distance.

As Figure 7a illustrates, the algorithm inversely projects the extracted global spatial features into a hash matrix space. This matrix space matches the dimensionality of the global static grid map. Consequently, this projection constructs the complete PSP physical field. Within this visualization, five-pointed stars denote turning points, while dots represent parameterized sampling points. Furthermore, varied mapping colors indicate the channel's physical width margin within that specific coordinate domain. Traditionally, complex collision detection and geometric intersection calculations demand real-time execution within the search loop. The proposed method transforms these operations into a single matrix memory read, featuring a time complexity of $\mathcal{O}(1)$. This transformation significantly conserves computational resources. Ultimately, it facilitates the efficient operation of the subsequent multi-mode dynamic driving state machine.

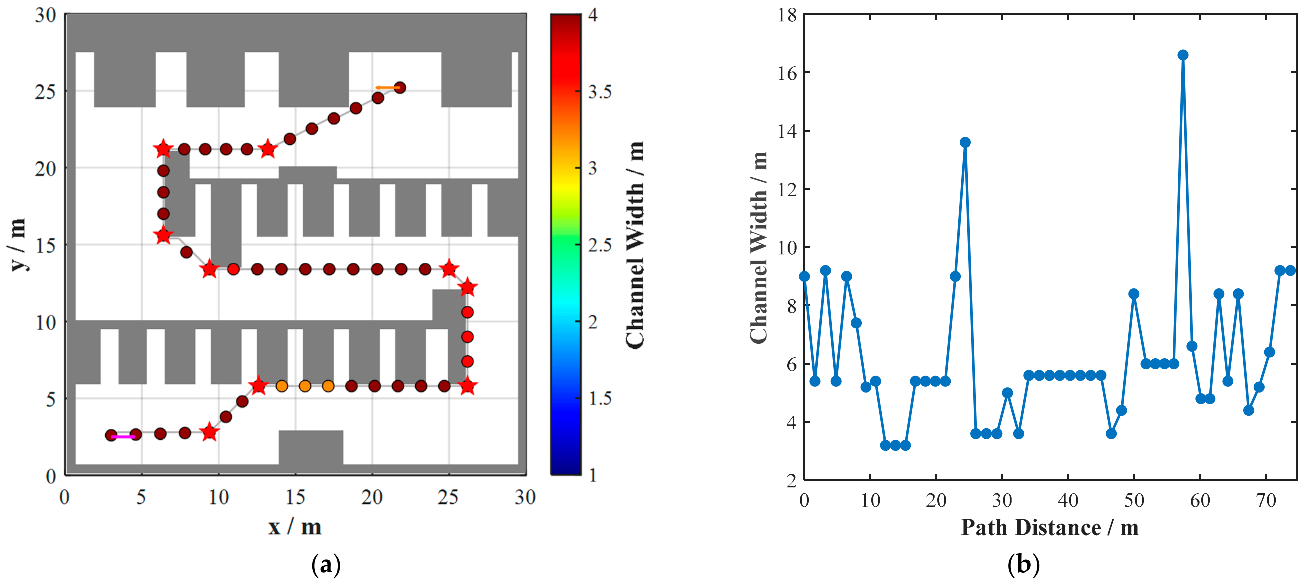


Figure 7. Visualization of the Predictive Spatial Profiling (PSP) field. (**a**) Extracted topological spatial skeleton. The color gradient indicates the available channel width, and red pentagrams highlight critical turning corners. (**b**) Continuous variation in the channel safety margin. This variation establishes the fundamental spatial boundary condition for the dynamic state machine.

Figure 7b presents a curve mapping passage width against path distance within a complex parking scenario. This visualization intuitively demonstrates the capability of the PSP field to extract environmental topological features. The curve precisely projects high-dimensional geometric constraints into a one-dimensional distance space. Consequently, it verifies the accuracy of the PSP measurement mechanism. Furthermore, it establishes a robust triggering basis for adaptive mode switching within the dynamic state machine.

4. Design of Adaptive Multi-Mode Graph Search Algorithm

This section introduces an adaptive multi-mode graph search algorithm leveraging the established Predictive Spatial Profiling (PSP) field. The proposed method fundamentally mitigates node proliferation induced by multi-mode expansion in 4WIS vehicles. Additionally, it prevents local oscillations near the target pose inherent to discrete searches. Specifically, a dynamic driving state machine adaptively reduces the dimensionality of the node-expansion process. Concurrently, a reconstructed comprehensive cost function and a dual-scale heuristic mechanism guide efficient optimization. Finally, a multi-dimensional terminal analytical strategy guarantees precise convergence to the target pose. Figure 8 illustrates the overall framework of the proposed algorithm.

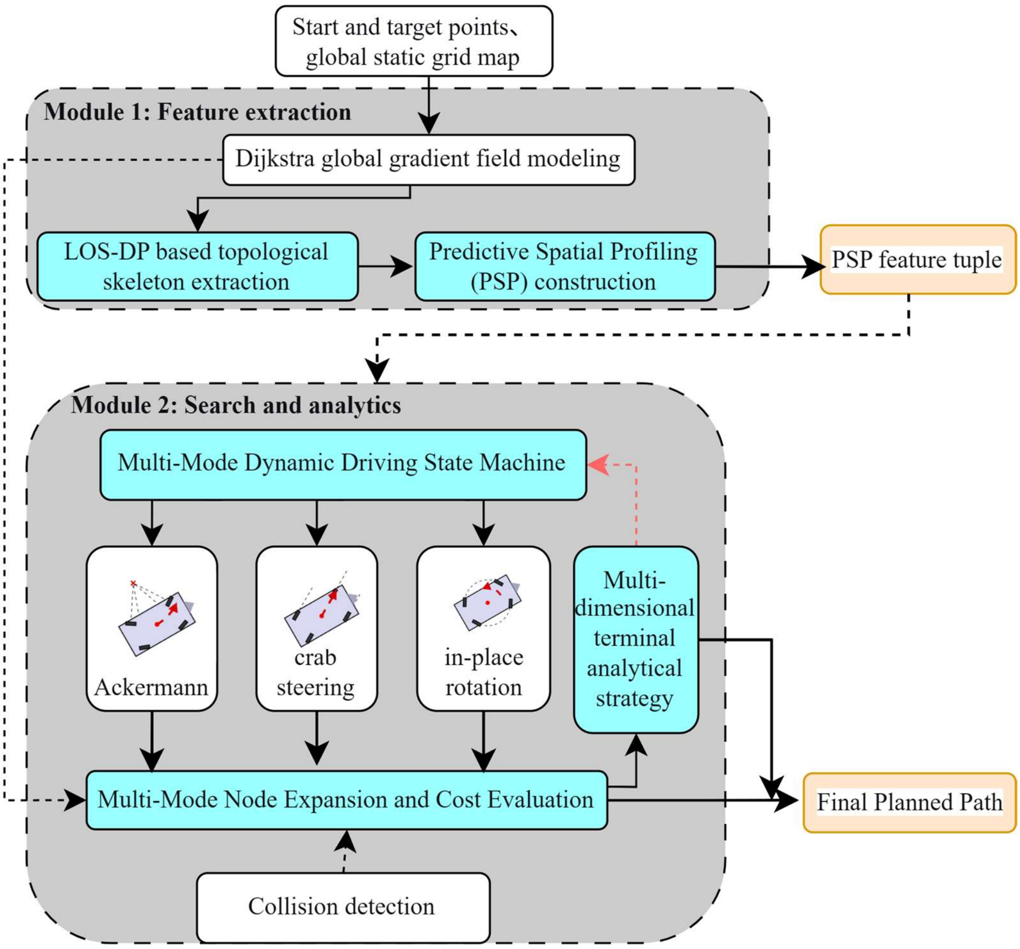


Figure 8. Algorithm framework diagram.

4.1. Multi-Mode Dynamic Driving State Machine Based on PSP

To facilitate adaptive and proactive search-tree pruning, the proposed algorithm formulates a dynamic driving state machine based on spatial geometric preview. The core mechanism of this state machine dynamically determines the set of permissible vehicle motion modes. The specific logic and decision-making process of this dynamic driving state machine are detailed in Algorithm 1. The available physical clearance within the localized environment strictly dictates this determination.

Prior to node expansion, the algorithm leverages the spatial feature tuple extracted in Section 3. Consequently, it defines the set of currently available motion modes, denoted as M_{active}. The following piecewise function governs this adaptive mode-decision logic:

$$M_{\text{active}} = \begin{cases} \{AS\}, & W \geq W_{\text{safe}} \\ \{AS, CR, SP\}, & W < W_{\text{safe}}, S_c \leq S_d, W \geq W_{\text{spin}} \\ \{AS, CR\}, & others \end{cases} \tag{6}$$

where M_{active} is the set of permissible motion modes; W is the absolute physical channel width at the current node; represents the path distance from the current node to the nearest upcoming topological turning point; and S_d defines the preset preview warning distance for upcoming curves. The parameter S_d functions as a spatial trigger threshold defining the region of interest for approaching corners. A path distance less than or equal to S_d signifies that the vehicle has entered the vicinity of a turning point. This configuration efficiently allocates computational resources. It prevents the premature evaluation of complex maneuvers within straight corridors. Concurrently, it guarantees an adequate spatial window for mode switching near topological bottlenecks.

The vehicle's rigid-body geometric dimensions and kinematic limits determine the boundary conditions for these safe passage thresholds. The safe passage width threshold for the Ackermann steering mode and its corresponding lateral expansion compensation are formulated as follows:

$$\begin{cases} W_{safe} = B_{veh} + \Delta W_{ack} + 2d_{safe} \\ \Delta W_{ack} = \sqrt{\left(R_{min} + \frac{B_{veh}}{2}\right)^2 + \left(\frac{L_{veh}}{2}\right)^2} - R_{min} - \frac{B_{veh}}{2} \end{cases} \tag{7}$$

where L_{veh} and B_{veh} represent the overall geometric length and width of the vehicle footprint, respectively; d_{safe} is the preset static physical safety margin on each side; and ΔW_{ack} denotes the additional swept width compensation generated when the vehicle negotiates its minimum turning radius R_{min} (as derived in Equation (3)) under the Ackermann steering mode. For a 4WIS vehicle exhibiting symmetric Ackermann steering ($\delta_r = -\delta_f$), its Instantaneous Center of Rotation (ICR) lies on the lateral axis passing through the vehicle's geometric center. By analytically calculating the discrepancy between the outermost bounding box corner trajectory and the innermost lateral edge trajectory, ΔW_{ack} is obtained as shown in Equation (7). Concurrently, the minimum physical clearance required to execute a zero-radius in-place rotation is governed by the vehicle's diagonal dimension, which is defined as

$$W_{spin} = \sqrt{L_{veh}^2 + B_{veh}^2} + 2d_{safe} \tag{8}$$

Importantly, the spatial thresholds in Equations (7) and (8) are rigid kinematic boundaries tied to the vehicle's physical profile, rather than empirically adjustable hyperparameters. The safety margin (uniformly set to 0.1 m) serves as a baseline geometric buffer in the grid space. While it provides preliminary mitigation for minor dynamic swept area variations caused by low-speed kinematic discrepancies, we acknowledge that it does not replace the need for real-time dynamic robust control to handle severe tire slip or significant actuator delays. This physically grounded derivation ensures absolute collision safety and algorithmic robustness across varying map topologies without requiring environment-specific manual parameter tuning.

Algorithm 1: Adaptive decision-making of multi-mode dynamic driving state machine

1: Initialize $M_{\text{active}} \leftarrow \varnothing$
2: $(W, S_{\text{c}}) \leftarrow M_{\text{PSP}}(\text{round}(x_k), \text{round}(y_k))$
3: **if** $W \geq W_{\text{safe}}$ **then**
4: $M_{\text{active}} \leftarrow \{AS\}$
5: **else if** $W < W_{\text{safe}}$ **and** $S_c \leq S_d$ **and** $W \geq W_{\text{spin}}$ **then**
6: $M_{\text{active}} \leftarrow \{AS, CR, SP\}$
7: **else**
8: $M_{\text{active}} \leftarrow \{AS, CR\}$
9: **end**
10: **return** M_{active}

https://doi.org/10.3390/wevj17070335

The state machine strictly prohibits node-expansion branches for crab steering and in-place rotation within spacious regions. Consequently, the search tree expands rapidly, constrained solely by four-wheel Ackermann kinematics. Conversely, the spatial preview may detect the vehicle entering a constrained environment near a topological dead zone. Provided sufficient clearance remains for in-place rotation, the algorithm subsequently grants full expansion authority across all maneuver modes. This strategy guarantees robust traversability. Concurrently, it strictly bounds computational overhead. Ultimately, dynamically modulating the permissible mode set based on local channel width achieves a highly adaptive coupling between vehicle kinematics and environmental geometry.

4.2. Comprehensive Cost and Heuristic Function Design

The A* algorithm is a graph search technique that employs a heuristic function to prioritize nodes, offering an efficient method for resolving optimal paths within static grid spaces. Conversely, the Hybrid A* algorithm is specifically tailored for path planning problems subject to vehicle kinematic constraints. Leveraging the capability of the Hybrid A* framework to accommodate these physical limitations, global path planning for 4WIS vehicles fundamentally entails finding a minimum-cost, kinematically feasible trajectory that connects the start and target poses within a continuous state space [29]. Let the space occupied by global static obstacles be denoted as O_{obs}, and the safe free space be defined as O_{free}. The planning task can thus be formulated as an optimization problem aimed at minimizing the comprehensive cost function $f(s_k)$

$$f(s_k) = g(s_k) + h(s_k) = \sum_{i=1}^{k-1} c(s_i) + h(s_k) \tag{9}$$

subject to the following kinematic and spatial hard constraints:

$$\begin{cases} \Omega(s_k) \in O_{free} \\ \delta_k \in [-\delta_{max}, \delta_{max}] \end{cases} \tag{10}$$

where $g(s_k)$ is the actual cumulative cost from the start state to the current state s_k; $c(s_{k-1})$ denotes the execution cost of a single-step state transition; $h(s_k)$ represents the heuristic cost from the current state to the target; $\Omega(s_k)$ defines the geometric footprint of the vehicle's rectangular contour at node s_k; and δ_{max} is the maximum wheel steering angle imposed by the vehicle's mechanical constraints.

4.2.1. Multi-Mode Comprehensive Cost Function Based on Mode Switching Penalty

In the multi-mode path planning process, the single-step actual cost $c(s_{k-1})$ accounts for not only the geometric displacement but also a specific penalty for mode switching. To suppress disorderly mode jumps and ensure trajectory smoothness, the recursive derivation of the actual cumulative cost s_k for a state node $g(s_k)$ is defined as follows:

$$g(s_k) = g(s_{k-1}) + c(s_{k-1}) = g(s_{k-1}) + w_{mode}\Delta s + P_{sw} I_{sw} \tag{11}$$

where $g(s_{k-1})$ is the parent node cost; w_{mode} represents the specific weight coefficient of the corresponding motion mode; P_{sw} is the mode switching penalty base; and I_{sw} denotes the mode-switching indicator function. When a mode change occurs, I_{sw} takes a value of 1, triggering a significant penalty. This design constrains the search tree to maintain a consistent motion mode as much as possible, thereby ensuring the smoothness of the planned trajectory. For translational modes—namely forward, backward, and crab steering—the single-step state variation Δs adopts the Euclidean distance

$$\Delta s = \sqrt{(x_{new} - x_{curr})^2 + (y_{new} - y_{curr})^2} \tag{12}$$

When in-place rotation is performed, it adopts the product of the angular change and a weight coefficient

$$\Delta s = \omega|\Delta\theta| \tag{13}$$

Notably, the cost function and the physical hard constraints of the state machine operate synergistically. The state machine categorically disables specific motion modes based on rigid spatial thresholds. Concurrently, the mode-switching penalty within the cost function provides soft heuristic guidance across the remaining permissible modes. Ultimately, this dual mechanism robustly prevents the algorithm from executing frequent, chaotic mode transitions within localized regions.

4.2.2. Dual-Scale Heuristic Mechanism Based on Environmental Gradients

The heuristic function critically dictates both search efficiency and path quality within graph-search algorithms. Traditional heuristic designs typically rely on calculating Reeds–Shepp curve distances. However, this calculation proves computationally intensive and imposes substantial real-time overhead. To accelerate search-tree convergence, Klančar et al. [29] proposed pre-constructing a global cost potential field via the Dijkstra algorithm. Their approach guides the robot's motion by leveraging potential-field gradient information. Building upon this methodology, this study introduces a dual-scale heuristic mechanism. This mechanism rigorously integrates static topological distance with dynamic environmental gradients. The heuristic function $h(s_k)$ is formulated as follows:

$$h(s_k) = h_{\text{Dijkstra}}(s_k) + w_{\text{yaw}}\Delta\theta_{\text{yaw}} \tag{14}$$

where $h_{\text{Dijkstra}}(s_k)$ is the Dijkstra distance precomputed based on the global static grid; w_{yaw} denotes the heading deviation penalty weight; and $\Delta\theta_{\text{yaw}}$ represents the minimized absolute deviation between the node's current yaw angle θ_k and the ideal gradient heading angle θ_{ideal} of the environment.

The algorithm performs discrete difference operations on the precomputed Dijkstra topological potential field F_{Dijkstra} along the horizontal and vertical coordinate axes to derive the ideal environmental guidance heading θ_{ideal}. To ensure that the guidance vector consistently points toward the target—specifically following the direction of steepest potential descent—the computation employs negative gradient logic

$$\nabla_x F_{\text{Dijkstra}} = \frac{F_{\text{Dijkstra}}(x+1, y) - F_{\text{Dijkstra}}(x-1, y)}{2\Delta x} \tag{15}$$

$$\nabla_y F_{\text{Dijkstra}} = \frac{F_{\text{Dijkstra}}(x, y+1) - F_{\text{Dijkstra}}(x, y-1)}{2\Delta y} \tag{16}$$

$$\theta_{ideal} = \text{atan2}(-\nabla_y F_{Dijkstra}, -\nabla_x F_{Dijkstra}) \tag{17}$$

The angular displacement minimization criterion is adopted to map the deviation between the node's current yaw angle θ_k and the ideal environmental gradient heading θ_{ideal} to the minimum absolute value domain $[0, \pi]$

$$\Delta\theta_{yaw} = \min(|\theta_k - \theta_{ideal}|,\ 2\pi - |\theta_k - \theta_{ideal}|) \tag{18}$$

where $\nabla_x F_{Dijkstra}$ and $\nabla_y F_{Dijkstra}$ denote the horizontal and vertical spatial gradients of the Dijkstra distance field, respectively. Equation (18) ensures that the single-step heading deviation $\Delta\theta_{yaw}$ is calculated using the shortest angular path constraint, effectively mapping the heading error strictly within the domain $[0, \pi]$. This dual-scale heuristic mechanism enables the search tree to precisely avoid obstacles while ensuring the vehicle's body posture conforms to the geometric alignment of the physical channel, thereby effectively suppressing redundant expansion nodes.

4.3. Multi-Dimensional Terminal Analytical Strategy

4.3.1. Multi-Dimensional Terminal Analytical Triggering Mechanism

Traditional heuristic methods frequently exhibit redundant oscillations as search nodes approach the target region. Discrete expansion steps and nonholonomic constraints primarily induce these terminal instabilities. To resolve this issue, this study introduces a terminal analytical strategy integrating spatial preview features with parallel maneuver modes. During the terminal optimization phase, this strategy facilitates a seamless transition from discrete exploration to continuous algebraic resolution.

Let the sequence of extracted topological turning points be defined as a set $T = \{p_1, p_2, \ldots, p_N\}$, where p_N represents the final turning point before the target p_{target}. During the multi-mode search, each node state S_k dynamically maintains a tracking index $i_k \in \{1, 2, \ldots, N\}$ to indicate its currently active segment along the topological skeleton. For a node with geometric center $p_{curr} = (x_k, y_k)$, the algorithm activates the terminal analytics if and only if its tracking index has reached the final segment ($i_k = N$) and the following geometric projection condition is satisfied:

$$(p_{curr} - p_N) \cdot (p_{target} - p_N) > 0 \tag{19}$$

Geometrically, this dot product guarantees that the angle between the vector from the final turning point to the current node and the vector to the target is acute. The logical combination of index gating and orthogonal half-space checking ensures the vehicle has strictly crossed the final turning bottleneck and entered the terminal unconstrained corridor.

4.3.2. Analytical Trajectory Combination and Comprehensive Cost Evaluation

To guarantee robust convergence, the proposed strategy constructs three representative terminal connection trajectories in parallel. The primary candidate is the Reeds–Shepp (RS) curve. This curve rapidly aligns the vehicle with the target pose within relatively unconstrained spaces. If the RS curve violates local geometric constraints, the algorithm subsequently evaluates two composite analytical maneuvers. The first composite approach guides the vehicle to the target position via an Ackermann arc. Subsequently, an in-place rotation aligns the final heading. Conversely, the second approach executes an initial in-place rotation to achieve the target orientation. Following this, crab steering facilitates a direct translation to the goal.

To select the candidate path with the minimum overhead, this study constructs a terminal analytical cost function J_{term}

$$J_{term} = w_{dist} L_{term} + w_{spin} \Delta\theta_{spin} + w_{sw} N_{sw} \tag{20}$$

where L_{term} is the total length of the candidate trajectory; $\Delta\theta_{spin}$ denotes the accumulated angular displacement during in-place rotations; N_{sw} represents the frequency of motion mode transitions; and w_{dist}, w_{spin}, and w_{sw} are the corresponding penalty weight coefficients assigned to each component. These weighting coefficients are utilized to balance the distinct physical dimensions of path length, angular displacement, and switching frequency, integrating them into a comprehensive terminal cost. Furthermore, this normalization strictly bounds their functional scope to the local algebraic resolution phase executed after the final turning point. Consequently, these parameters remain completely independent of the heuristic weights governing the global search.

Prior to evaluating the cost function, the algorithm invokes the collision-detection module to verify whether the analytical trajectory physically encroaches upon obstacle-occupied spaces. Should the module detect interference, it strictly assigns the corresponding cost J_{term} to infinity. Ultimately, the algorithm selects the candidate path exhibiting the minimum comprehensive cost as the optimal terminal trajectory. This approach effectively eradicates the tentative oscillations inherent in discrete searches. Consequently, it achieves a seamless, precise alignment with the target pose.

4.4. Adaptive Multi-Mode Graph Search Algorithm Flow

By integrating the spatial preprocessing, the dynamic multi-mode state machine, and the terminal-pose analytical modules, the proposed adaptive multi-mode graph-search algorithm establishes a comprehensive search and optimization workflow. Figure 9 illustrates this complete process.

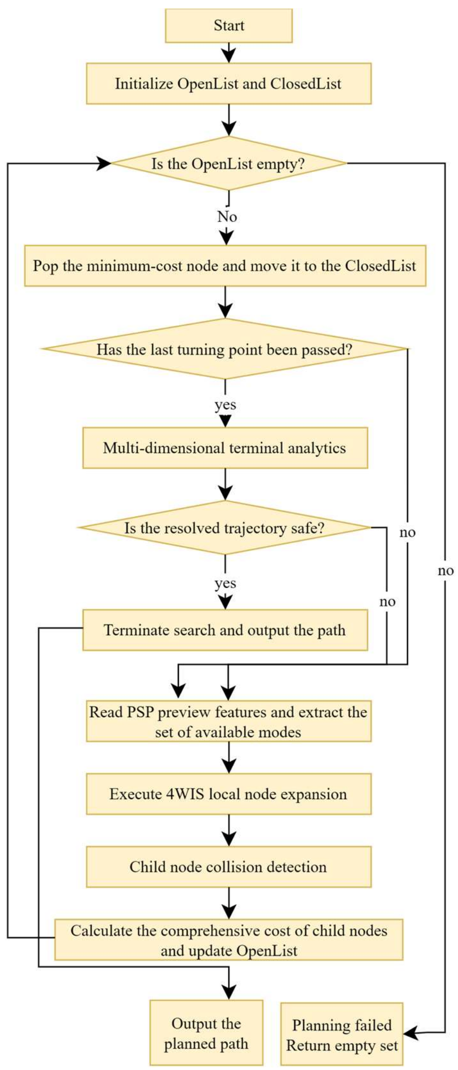


Figure 9. Optimization flowchart of the adaptive multi-mode graph search algorithm.

5. Simulation Verification and Results Analysis

To evaluate the feasibility and performance of the proposed adaptive path planning algorithm, which is driven by Predictive Spatial Profiling (PSP) and multi-dimensional terminal analytics, comprehensive simulation comparisons and ablation studies were conducted across various unstructured scenarios. All simulations were performed in MATLAB 2023a (MathWorks, Natick, MA, USA) on a personal computer (ASUS TUF Gaming F15 FX506LI_FX506LI, ASUS, Taipei, Taiwan) equipped with an Intel Core i5 processor (Intel Corporation, Santa Clara, CA, USA) and 16 GB of RAM.

5.1. Simulation Setup and Verification Scheme

The resolution of the global static grid map was uniformly set to 0.2 m. Vehicle kinematic constraints, such as the maximum wheel steering angle and wheelbase, as well as the collision detection mechanisms, were kept consistent across all experimental groups. To provide a comprehensive overview of the experimental conditions, the core algorithm parameters and vehicle configurations used in all simulation scenarios are summarized in Table 1.

Table 1. Comprehensive setup of algorithm parameters.

Parameter	Symbol	Value/Formula
Map resolution		0.2 m
Vehicle overall length	L_{veh}	4.0 m
Vehicle overall width	B_{veh}	2.0 m
Equivalent wheelbase	L	2.6 m
Maximum steering angle	δ_{max}	$\pi/4$
Safety margin	d_{safe}	0.1 m
Preview warning distance	S_d	3.0 m
Ackermann swept compensation	ΔW_{ack}	Equation (7)
Safe passage width	W_{safe}	Equation (7)
Spin minimum width	W_{spin}	Equation (8)
Mode-switching penalty base	P_{sw}	1.1
Heading deviation penalty weight	w_{yaw}	0.2
Terminal distance weight	w_{dist}	1.0
Terminal spin weight	w_{spin}	1.3
Terminal switch weight	w_{sw}	1.0

For the ablation study, the complete architecture of the proposed multi-mode graph search algorithm, integrating spatial preview and terminal analytics, serves as the benchmark. To maintain consistency throughout the empirical evaluation and prevent potential misinterpretation, the ablation study in this section is conducted within the identical complex parking scenario detailed subsequently in Section 5.5. The PSP-based dynamic driving state machine and the multi-dimensional terminal analytical strategy were systematically deactivated to verify their individual contributions to pruning high-dimensional redundant branches, mitigating mode chattering, and suppressing tentative terminal oscillations.

To rigorously evaluate the performance of the proposed adaptive multi-mode planning framework, comparative simulations are conducted against two representative baseline algorithms. To ensure a fair comparison and eliminate the influence of basic implementation settings, all three algorithms share identical underlying environmental configurations, such as the 0.1 m safety margin expansion and the rectangular collision detection footprint. The baseline planners are defined as follows.

The first baseline, Classic 2WS-HA*, represents the standard Ackermann-steering Hybrid A* algorithm. It reflects the conventional planning capabilities of standard autonomous vehicles, providing a reference to evaluate the necessity and advantage of introducing the 4WIS configuration in constrained environments.

The second baseline is the Conventional 4WIS-HA*, a reconstructed multi-mode Hybrid A* algorithm based on the method proposed by Qin et al. [14]. This baseline incorporates Ackermann, crab, and in-place rotation modes within a unified graph-search framework and employs a multi-objective cost function with mode-switching penalties to guide the search. Unlike the proposed method, it lacks geometric environmental pre-analysis and globally permits all maneuver options during node expansion. Furthermore, it relies on standard heuristics and conventional distance-threshold terminal connections, representing a typical state-of-the-art unconstrained multi-mode planner.

By standardizing the physical collision boundaries and basic kinematic constraints, the comparative study focuses on a system-level evaluation. Any observed improvements in planning efficiency and node reduction can be objectively attributed to the proposed integrated architecture, specifically the PSP-based dynamic state machine, the dual-scale heuristic, and the multi-dimensional terminal analytical strategy. The detailed architectural differences among the evaluated planners are summarized in Table 2.

Table 2. Methodological comparison between the proposed algorithm and the state-of-the-art baseline.

Component	Proposed Algorithm	4WIS-HA*	2WS-HA*
Available Motion Modes	Ackermann, Crab, In-place Spin	Ackermann, Crab, In-place Spin	Ackermann
Node Expansion Strategy	Adaptive Pruning	Global Unconstrained	Continuous
Heuristic Function	Dual-scale Heuristic	2D Dijkstra RS Distance	2D Dijkstra RS Distance
Cost Function	Mode-switching Penalty	Mode-switching Penalty	Without
Terminal Convergence	Multi-dimensional Terminal analytics	Distance threshold single Reeds–Shepp shot	Distance threshold single Reeds–Shepp shot

To ensure the evaluation of computational efficiency is statistically rigorous, the planning times reported in the subsequent comparative experiments are obtained by averaging 20 independent trials under identical environmental conditions, expressed in the format of mean value plus or minus standard deviation. This protocol mitigates random variations introduced by operating system CPU scheduling and the MATLAB runtime framework. Conversely, because the graph search initialization and node expansion rules are deterministic, the remaining evaluation metrics, including node expansions, path lengths, and mode-switching frequencies, remain constant across trials and are presented as single deterministic values.

To intuitively illustrate the trajectory characteristics generated by each algorithm, a unified color-coding scheme is applied in the subsequent figures. Solid green and blue lines denote forward and reverse trajectories in the Ackermann steering mode, respectively. Solid red lines represent the crab steering mode, and dark purple dots indicate the operating poses during in-place rotation. Additionally, clusters of light gray line segments visualize the search tree branches generated during optimization, with the start and target poses marked by distinctively colored arrows.

5.2. Progressive Ablation Study

To evaluate the individual contributions of the Predictive Spatial Profiling (PSP) mechanism and the multi-dimensional terminal analytical strategy to the proposed adaptive multi-mode graph-search algorithm, an ablation study was conducted in a complex parking scenario. Using the complete algorithmic architecture as a benchmark, the PSP-based dynamic driving state machine and the terminal analytical strategy were systematically disabled. This approach verified their respective efficacies in pruning high-dimensional redundant branches, mitigating mode chattering, and suppressing terminal oscillations.

As shown in Table 3, the baseline algorithm—which lacks both the PSP state machine and the terminal analytical mechanism—consumed 1.54 s of planning time and expanded 7116 nodes (see Figure 10). Furthermore, the optimization process suffered from three chaotic mode transitions. This substantial computational overhead primarily stems from the blind exploration of high-dimensional nodes across an expansive, unconstrained state space.

Table 3. Comparison of ablation study simulation results.

PSP State Machine	Terminal Analytics	Time (s)	Number of Node Expansions	Mode Switch Count	Mean Absolute Curvature (m^{-1})
Without	Without	1.54 ± 0.058	7116	3	0.199
With	Without	1.13 ± 0.225	2923	2	0.226
With	With	0.96 ± 0.018	920	1	0.178

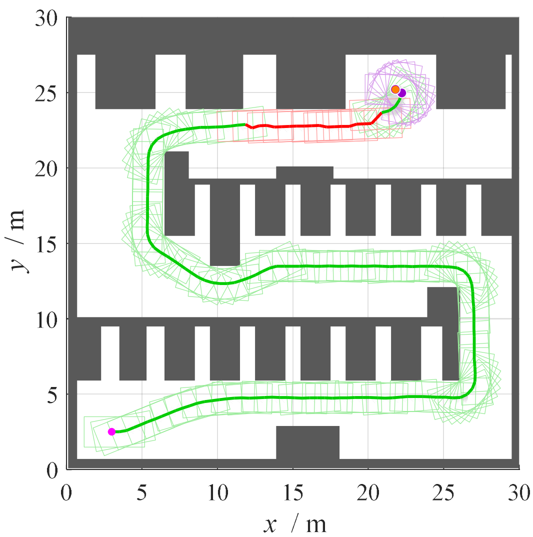


Figure 10. Simulation path without the PSP state machine and terminal analytical mechanism.

Introducing the PSP-based dynamic driving state machine decreased the planning time to 1.13 s, reduced the number of expanded nodes to 2923, and curtailed the mode-transition frequency to two (see Figure 11). These quantitative results confirm that extracting spatial prior features effectively prunes redundant branches from the search tree. Specifically, within spacious channels, this mechanism strictly bounds both the node-expansion scale and the frequency of mode transitions.

Activating both the predictive spatial preview and the terminal analytical mechanisms yielded a solution time of 0.96 s and drastically compressed the node-expansion scale to just 920 nodes, as depicted in Figure 12. Compared with the pure graph-search baseline, the complete proposed algorithm reduced the planning time by 37.66% and the number of expanded nodes by 87.07%. Ultimately, the adaptive transition to terminal algebraic curves successfully eradicated the tentative oscillations inherently caused by discrete searches near the target pose.

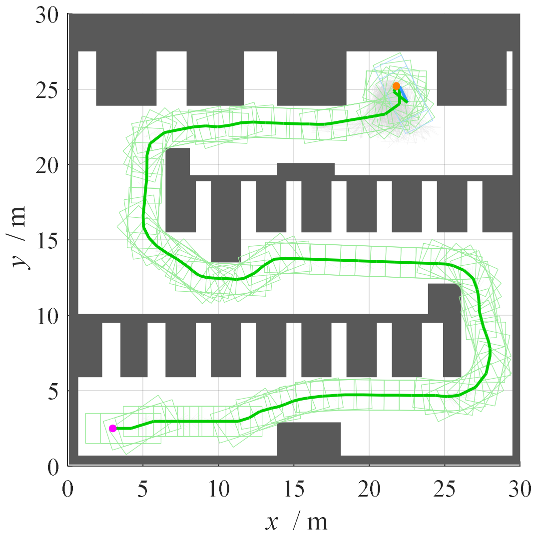


Figure 11. Simulation path without the terminal analytical mechanism.

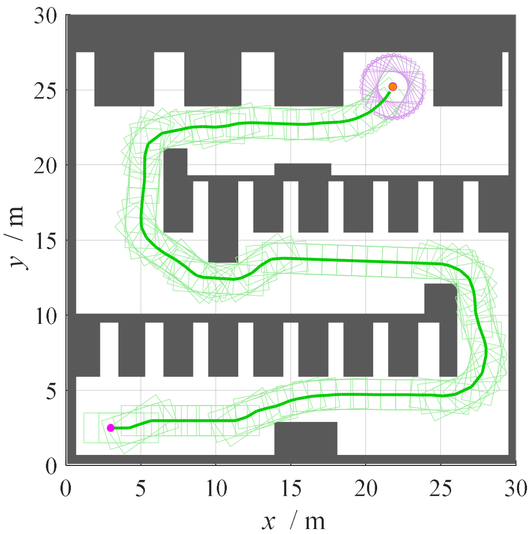


Figure 12. Simulation path of the complete proposed algorithm.

To quantitatively evaluate trajectory smoothness, Figure 13 plots the signed-curvature profile as a function of path distance. The configuration relying solely on the graph search (dashed red line) exhibits pronounced high-frequency terminal oscillations near the target zone, primarily due to the orientation constraints inherent to discrete state lattices. Notably, applying the predictive spatial preview alone (dash-dotted green line) slightly increases the mean absolute curvature. This occurs because the rigid spatial constraints of the preview mechanism forcefully restrict node-expansion directions, leading to somewhat stiffer localized maneuvers. Conversely, the complete proposed algorithm (solid blue line) fully eliminates both these oscillations and the localized stiffness by establishing a continuous mathematical mapping to the target pose. As summarized in Table 3, this combined approach ultimately achieves the lowest mean absolute curvature. These statistical results validate that the terminal analytical strategy effectively smooths the localized stiffness caused by spatial pruning, thereby enhancing overall trajectory smoothness without the need for post-processing.

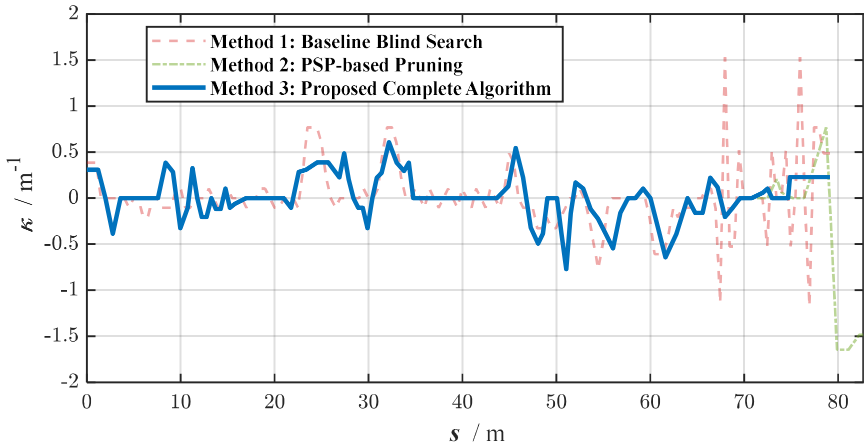


Figure 13. Trajectory smoothness evaluation based on signed curvature. The dashed red line represents the baseline blind search, the dash-dotted green line indicates the configuration with predictive spatial preview only, and the solid blue line represents the complete proposed framework.

5.3. Simulation Analysis of the Conventional Obstacle Corridor Scenario

The conventional corridor scenario primarily evaluates the fundamental obstacle-avoidance and trajectory-optimization capabilities of the algorithms amidst standard static obstacles. As summarized in Table 4, the simulation results indicate that the 2WS-HA* algorithm imposed a substantial computational burden, consuming 4.66 s of planning time and expanding 5530 nodes (see Figure 14). By leveraging multi-mode maneuverability, the 4WIS-HA* algorithm significantly reduced the search time to 0.57 s and the node-expansion scale to 602 (see Figure 15). However, relying solely on cost-weight configurations for mode decision-making, this baseline executed two redundant mode transitions while navigating standard curves.

Table 4. Comparison of simulation results for the conventional obstacle corridor scenario.

Algorithm	Time (s)	Number of Node Expansions	Path Length (m)	Mode Switch Count
2WS-HA*	4.66 ± 0.500	5530	27.83	1
4WIS-HA*	0.57 ± 0.019	602	25.66	2
Proposed algorithm	0.04 ± 0.006	157	25.75	1

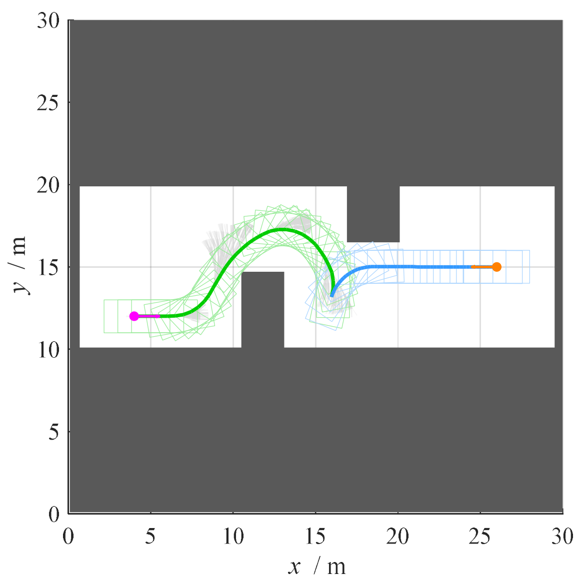


Figure 14. Simulation path of 2WS-HA* in the conventional obstacle corridor scenario.

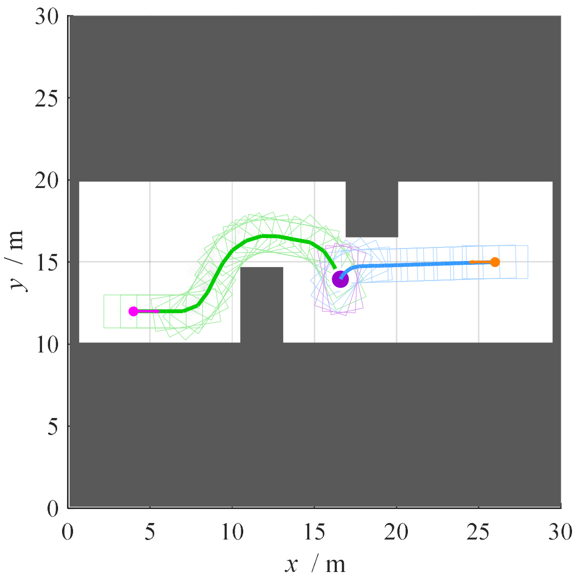


Figure 15. Simulation path of 4WIS-HA* in the conventional obstacle corridor scenario.

In contrast, guided by the global PSP preview, the dynamic state machine predetermined that the physical channel's width margin comfortably accommodated Ackermann forward kinematic constraints. Consequently, it proactively disabled the exploration of crab steering and in-place rotation from the onset of the search. Furthermore, the proposed algorithm successfully activated the terminal analytical strategy, circumventing excessively long reversing paths and complex shunting maneuvers near the goal to achieve precise target-pose alignment. Compared with the 4WIS-HA* baseline, the proposed approach further reduced the search time by 92.98% and the number of expanded nodes by 73.92%. Ultimately, the proposed algorithm generated an optimal trajectory (Figure 16) in merely 0.04 s while strictly limiting mode transitions to a single occurrence. These results compellingly demonstrate its exceptional scenario-adaptive decision-making capabilities.

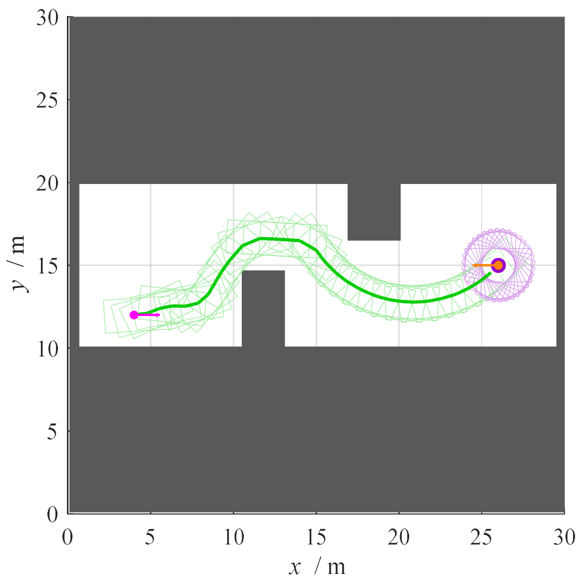


Figure 16. Simulation path of the proposed algorithm in the conventional obstacle corridor scenario.

5.4. Simulation Analysis of the Narrow Maze Scenario

The narrow maze scenario is characterized by challenging right-angle turns and topological blind spots. As detailed in Table 5, in this environment, the 2WS-HA* algorithm deteriorated into an exhaustive exploration trap, resulting in a prolonged search time of 27.73 s (see Figure 17). By leveraging its high maneuverability, the 4WIS-HA* algorithm successfully constrained the planning time to 5.02 s. However, because its soft-cost guidance mechanism exhibited severe decision-making hysteresis when navigating extreme angles, this baseline required 4304 node expansions and executed six chaotic mode transitions, as shown in Figure 18.

Table 5. Comparison of simulation results for the narrow maze scenario.

Algorithm	Time (s)	Number of Node Expansions	Path Length (m)	Mode Switch Count
2WS-HA*	27.73 ± 6.718	30,310	73.96	1
4WIS-HA*	5.02 ± 0.138	4304	68.91	6
Proposed algorithm	2.61 ± 0.080	988	69.44	1

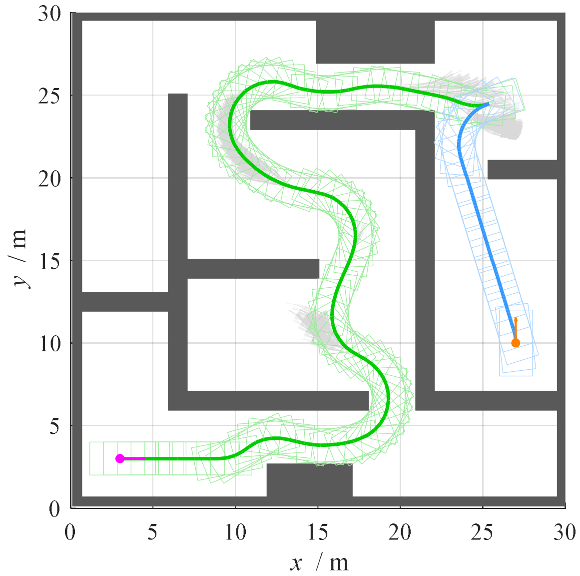


Figure 17. Simulation path of 2WS-HA* in the narrow maze scenario.

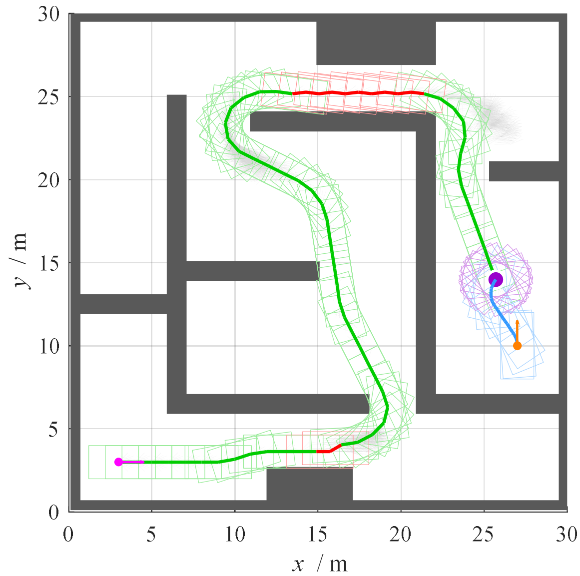


Figure 18. Simulation path of 4WIS-HA* in the narrow maze scenario.

Under these extreme operating conditions, the proposed algorithm demonstrates the predictive superiority of its spatial geometric hard-constraint framework. The PSP mechanism proactively identified topological bottlenecks, releasing full maneuver-expansion authority only when the vehicle approached highly constrained corners. Compared with the 4WIS-HA* baseline, the proposed approach further reduced the search time by 48.01% and the number of expanded nodes by 77.04%. Ultimately, the algorithm completed optimal path planning in merely 2.61 s (Figure 19) while restricting the entire trajectory to a single, strictly necessary mode transition. This performance fully exploits the obstacle-negotiation potential of 4WIS vehicles within severely constrained environments.

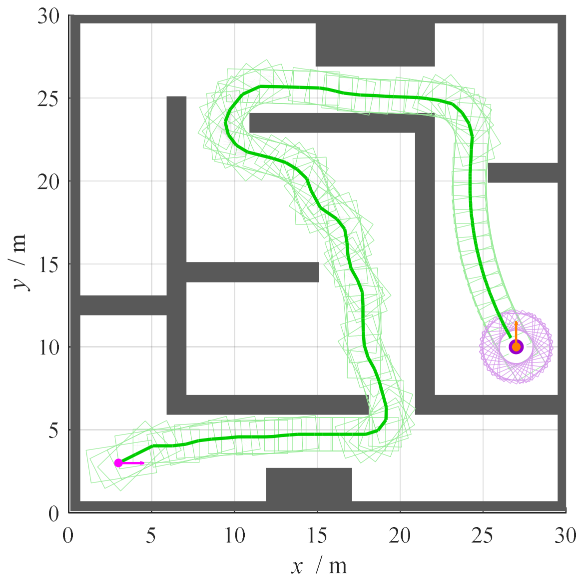


Figure 19. Simulation path of the proposed algorithm in the narrow maze scenario.

5.5. Simulation Analysis of the Complex Parking Scenario

The complex parking scenario features narrow driving lanes and imposes stringent requirements on terminal-pose convergence accuracy. As demonstrated in Table 6, confronted with severely limited adjustment space near the target area, the 2WS-HA* baseline suffered severe search disorientation, consuming 48.20 s of computational time and exceeding 50,000 expanded nodes (see Figure 20). Although the 4WIS-HA* algorithm successfully generated a feasible path, its lack of a dedicated terminal connection mechanism caused it to execute five tentative mode transitions near the endpoint to forcibly align with the target pose (Figure 21).

Table 6. Comparison of simulation results for the complex parking scenario.

Algorithm	Time (s)	Number of Node Expansions	Path Length (m)	Mode Switch Count
2WS-HA*	48.20 ± 0.205	51,781	85.48	3
4WIS-HA*	1.54 ± 0.036	1488	79.68	5
Proposed algorithm	0.96 ± 0.018	920	79.14	1

The proposed multi-dimensional terminal analytical strategy decisively resolved this bottleneck. Once the search tree crossed the terminal topological threshold, the parallel-triggered analytical operator instantly assumed control of the optimization process. Based on the geometric relationship between the current and target poses, this operator directly derived an interference-free analytical curve that minimized the comprehensive cost. Compared with the 4WIS-HA* baseline, the proposed approach further reduced the search time by 37.66% and the number of expanded nodes by 38.17%. Ultimately, the algorithm completed the planning task in merely 0.96 s, optimizing the total path length to 79.14 m (see Figure 22). Furthermore, it fundamentally resolved the discrete oscillation problem during the terminal stage, achieving precise and smooth docking at the target pose with a strictly single-mode transition.

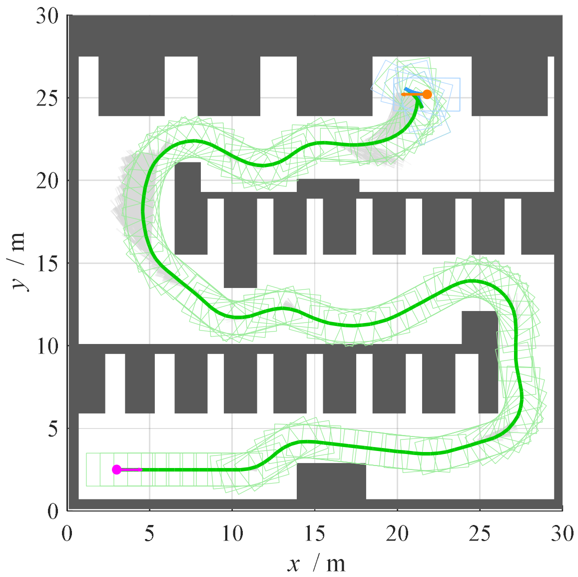


Figure 20. Simulation path of 2WS-HA* in the complex parking scenario.

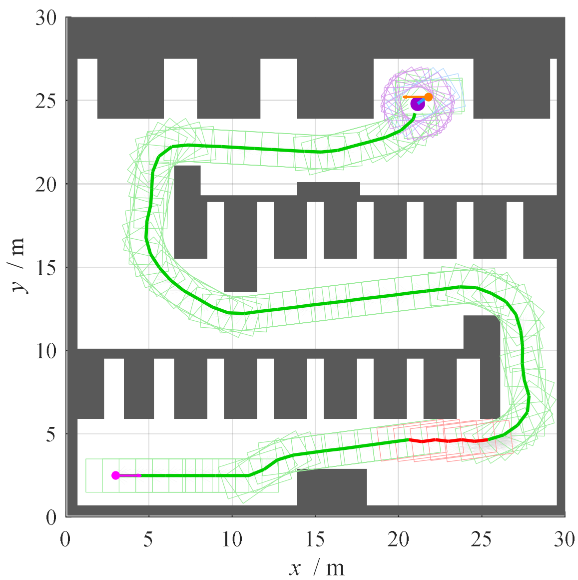


Figure 21. Simulation path of 4WIS-HA* in the complex parking scenario.

5.6. Parameter Configuration and Sensitivity Analysis

As summarized in Table 1, the proposed framework incorporates multiple parameters. Before presenting the results, it is necessary to categorize these parameters to clarify the scope of this sensitivity analysis. The spatial thresholds W_{safe} and W_{spin}, and alignment factors ΔW_{ack} used for local terminal analytics, are boundaries derived from the physical profile and kinematic limits of the vehicle. These parameters possess explicit physical meanings and are not globally tunable heuristics. The preview distance S_d serves as a spatial trigger for allocating computational resources, while the mode-switching penalty P_{sw} and the heuristic heading weight w_{yaw} directly affect the trade-off between global path exploration and trajectory smoothness. Therefore, to evaluate the robustness of the algorithm against threshold variations, the sensitivity analysis is specifically conducted on the mode-switching penalty P_{sw}, the heuristic heading weight w_{yaw} and the preview distance S_d. The tests are performed in the complex parking scenario detailed in Section 5.5 to observe the variation in planning performance. To maintain mathematical precision and exclude runtime fluctuations caused by the software environment, this analysis evaluates three deterministic structural metrics: node expansions, path lengths, and mode-switching frequencies.

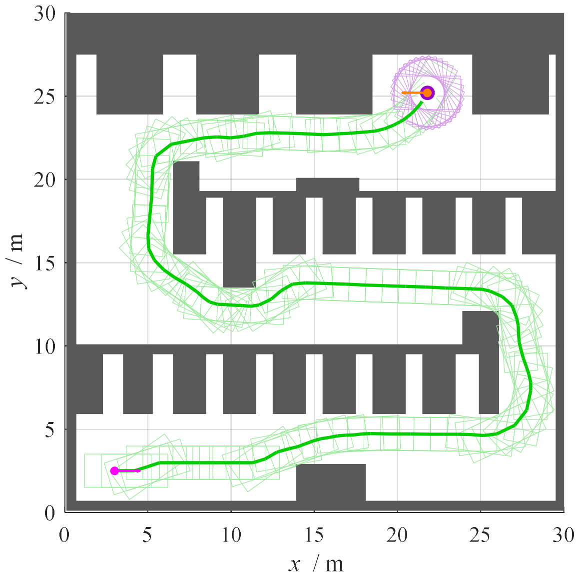


Figure 22. Simulation path of the proposed algorithm in the complex parking scenario.

Table 7 shows the influence of P_{sw}. When P_{sw} is set to 0.0, the algorithm frequently changes motion modes to minimize the spatial path length, resulting in 23 mode switches. Although the node expansion is slightly lower at 825, the frequent state transitions disrupt the trajectory coherence. Setting P_{sw} to 1.1 effectively suppresses redundant mode switches, reducing them to 1 while generating a node expansion of 920 and a refined path length of 79.14 m. Further increasing the penalty to 5.0 yields identical results to the 1.1 setting. This indicates that a threshold of 1.1 is sufficient to eliminate unnecessary switching behaviors without forcing the algorithm into rigid search patterns.

Table 7. Sensitivity analysis of mode-switching penalty P_{sw}.

Value of P_{sw}	Number of Node Expansions	Path Length (m)	Mode Switch Count
0.00	825	79.84	23
1.1	920	79.14	1
5.0	920	79.14	1

Table 8 presents the effect of w_{yaw}. Removing the heading guidance by setting w_{yaw} to 0.0 increases the node expansions to 994, the path length to 80.15 m, and the mode switches to 3, as the search process lacks directional focus in the lateral space. Conversely, a high weight of 1.0 forces the vehicle to strictly align with the target heading. This over-constrained condition reduces the total node count to 884 but increases the mode switches to 5, as the algorithm must perform multiple fragmented maneuvers to satisfy the rigid heading requirements. The proposed value of 0.2 provides a suitable balance, guiding the search efficiently while maintaining maneuvering flexibility.

Table 8. Sensitivity analysis of heuristic heading weight w_{yaw}.

Value of w_{yaw}	Number of Node Expansions	Path Length (m)	Mode Switch Count
0.0	994	80.15	3
0.2	920	79.14	1
1.0	884	78.99	5

To investigate the impact of the spatial preview distance S_d, tests were conducted using values of 1.0 m, 3.0 m, and 6.0 m. As shown in Table 9, the results indicate that the algorithm successfully found a viable path under all conditions with a stable mode-switch count of one, demonstrating excellent robustness against threshold variations. When the preview distance was reduced to 1.0 m, the vehicle remained restricted to the conventional Ackermann mode until it was extremely close to the corner. This severely limited the branching factor of the search tree, yielding the minimum node expansions (668) and a slightly shorter path of 78.73 m. While the local geometry of the current parking scenario fortuitously allowed a successful traverse at $S_d = 1.0$ m, yielding the fewest expanded nodes, this empirical optimum does not satisfy the theoretical safety bounds required for universal autonomous navigation. From a strictly kinematic perspective, the preview distance S_d must provide an adequate longitudinal spatial buffer for the vehicle to execute complex multi-mode transitions safely. Given the vehicle's equivalent wheelbase L of 2.6 m, an S_d of 1.0 m structurally implies that the vehicle's front footprint has already encroached deeply into the intersection before high-maneuverability modes are unlocked. In a theoretical worst-case narrow turn, this delayed triggering mathematically guarantees a physical collision during the spin or crab transition phase. Therefore, configuring S_d to 3.0 m—a value deliberately chosen to be slightly larger than the equivalent wheelbase—functions as a fundamental geometric hard constraint rather than an empirical tuning parameter. It systematically guarantees that the vehicle maintains a full chassis-length operational window to safely transition modes and align its posture entirely before its physical footprint enters the collision-prone topological bottleneck, prioritizing absolute collision avoidance over minor localized search efficiency. Conversely, extending the preview distance to 6.0 m caused the state machine to prematurely unlock high-freedom motions (like crab steering and in-place rotation) on straight or gentle segments. This globally expanded the search space and generated redundant maneuvering branches, driving the node expansions up to 1150. The chosen baseline of 3.0 m achieved a vital balance. It prevented potential heuristic backtracking in tighter spaces while effectively suppressing the proliferation of redundant nodes in wide corridors.

Table 9. Sensitivity analysis of the preview warning distance S_d.

Value of S_d	Number of Node Expansions	Path Length (m)	Mode Switch Count
1.0	668	78.73	1
3.0	920	79.14	1
6.0	1150	79.48	1

6. Discussion

The adaptive multi-mode graph search algorithm proposed herein demonstrates significantly superior planning efficiency and mode transition smoothness compared to baseline methods across various constrained scenarios. Considering previous studies and the working hypotheses of this research, the implications of these findings warrant further discussion.

Regarding mode decision-making mechanisms, existing literature primarily relies on soft weights within the cost function to guide four-wheel independent steering (4WIS) vehicles in executing motion mode transitions. Although this approach benefits from implementation simplicity, inherent decision-making hysteresis is inevitable when encountering local geometric extrema, such as narrow mazes or complex parking environments. Consequently, this hysteresis results in substantial node expansion and frequent mode chattering. The ablation study validates this limitation: even with an identical cost function, removing the Predictive Spatial Profiling (PSP)-based state machine causes the node expansion count to surge to 7116, accompanied by three disorderly mode transitions. Conversely, incorporating the state machine drastically reduces the expanded nodes to 2923 and curtails mode transitions to two. These quantitative results confirm that strict geometric admission pruning suppresses redundant exploration more effectively than relying solely on a posteriori cost parameter tuning. This finding directly corroborates the primary hypothesis of this study: for multi-mode vehicles, the feedforward utilization of environmental geometric features significantly enhances decision-making agility and robustness compared to traditional reactive cost penalties.

Building upon the empirical success of the spatial profiling mechanism, the theoretical properties of the algorithm regarding completeness and optimality require clarification. Although the state machine employs an aggressive pruning strategy for motion modes, this pruning is governed by strict geometric lower bounds defined in Equations (7) and (8), which are derived directly from the rigid body physical envelope of the vehicle. Consequently, the state machine does not eliminate any physically viable configuration space. This property ensures that the algorithm maintains resolution completeness, guaranteeing a solution if one exists within the given grid resolution. Conversely, from a theoretical standpoint, the algorithm is deliberately designed to be sub-optimal. The introduction of mode switching penalties and the dual-scale heuristic intentionally relaxes the admissibility criterion of classical heuristic search, making a deliberate trade-off between strict theoretical spatial optimality and practical computational efficiency. This mathematical violation trades absolute spatial optimality for a substantial acceleration in computational speed and a significant enhancement in trajectory smoothness. For practical engineering applications of four-wheel independent steering vehicles, eliminating high-frequency steering oscillations and reducing planning time are strictly prioritized over finding the absolute shortest mathematical path.

Regarding the computational overhead, the proposed framework introduces specific preprocessing and evaluation steps yet maintains high overall efficiency. Constructing the Predictive Spatial Profiling field relies on line-of-sight checking and the Douglas–Peucker algorithm to extract the environmental topology. This step introduces a one-time preprocessing time complexity of $\mathcal{O}(V \log V + E)$, with V and E representing the vertices and edges of the spatial graph. Once the environment is mapped into a discrete spatial hash matrix, the real-time threshold queries during the node expansion phase operate in strictly constant time $\mathcal{O}(1)$. Similarly, the terminal analytical strategy incurs a negligible computational burden. The triggering mechanism relies on a simple geometric projection evaluated in constant time $\mathcal{O}(1)$, and the subsequent generation of the analytic Reeds–Shepp path also requires constant time $\mathcal{O}(1)$. Although these modules necessitate a slight preprocessing investment, they deterministically prune redundant motion modes and avoid global arbitration. This mechanism effectively suppresses the exponential time growth associated with unconstrained multi-mode branching, which is typically bounded by $\mathcal{O}(b^d)$, where b signifies the branching factor and d represents the search depth. Consequently, the massive reduction in node expansions thoroughly offsets the initial computational overhead and validates the algorithmic efficiency demonstrated in the simulation results.

The core innovation of the multi-dimensional terminal analytical strategy lies in transforming the inherent terminal oscillation dilemma of discrete searches into continuous algebraic curve resolution. In the complex parking simulation, the complete algorithm successfully reduced the number of mode transitions from five to strictly one, while simultaneously shortening the search time from 1.54 s to 0.96 s. However, the efficacy of this strategy depends on its triggering condition—specifically, whether the preview distance parameter indicates that the vehicle has successfully passed the final topological turning point. In the current implementation, this critical turning point relies on the global static skeleton for determination. Should unexpected minor obstacles or temporarily occluded zones appear within the channel, the purely static preview criterion may fail. Therefore, introducing local perception data to dynamically calibrate this triggering threshold represents a critical engineering challenge that must be addressed prior to real-world deployment.

From a broader perspective, the integrated spatial preview and state machine framework is not limited to 4WIS vehicles. Any ground robotic system with multiple motion modes—including omnidirectional mobile platforms, tracked vehicles, or structurally deformable wheeled robots—can utilize the lateral orthogonal profiling mechanism proposed herein to construct custom mode admission criteria, provided their specific geometric passage conditions can be mathematically defined. Furthermore, the PSP mechanism essentially simplifies high-dimensional environmental constraints into a unique one-dimensional passage width curve. This powerful dimensionality reduction paradigm holds significant application potential for other complex planning problems requiring environmental prescreening, such as corridor allocation for warehouse robot swarms or terrain traversability assessment for search-and-rescue units.

The primary limitation of this study lies in the assumption of a strictly static environment. All simulations are executed on pre-mapped global grids without accounting for localized topological perturbations induced by dynamic obstacles. If a dynamic obstacle temporarily occludes the passage, the precomputed PSP field becomes invalid, thereby necessitating local replanning. At present, the replanning mechanism merely falls back to a heuristic search following an analytical failure, lacking a dedicated module to proactively monitor spatiotemporal topological shifts.

Furthermore, deploying this purely geometric global planner in real-world scenarios introduces significant practical implementation challenges, particularly concerning map uncertainties and localization errors. The proposed terminal analytical strategy relies heavily on precise spatial coordinates. If severe localization drift occurs during physical deployment, the projected analytical curve may inadvertently violate the safety margin, potentially leading to collisions. Another critical failure case emerges in the presence of dynamic obstacles. If a dynamic entity suddenly occludes the final topological turning point, the algebraic triggering condition—formulated as the dot product between the current position vector and the target position vector relative to the turning point being greater than zero—might fail to activate. Under such circumstances, the global algorithm could become trapped in an exhaustive discrete search or output a path that conflicts with the dynamic environment. To mitigate these risks and ensure robust real-world performance, this global graph search framework must be hierarchically coupled with a local receding horizon planner. A dedicated local planner utilizing real-time perception data is strictly required to execute dynamic obstacle avoidance and compensate for underlying localization deviations.

To address these limitations, future research will proceed in three directions. First, it aims to construct a spatiotemporal joint graph search framework that encodes the predicted trajectories of dynamic obstacles as time-varying constraints. This will enable the state machine to preemptively adjust motion modes based on the anticipated temporal evolution of the passage width. Second, kinodynamic speed planning will be coupled with the existing spatial path planning. Enabling the vehicle to synchronously modulate its velocity during crab steering or in-place rotation will significantly mitigate tracking errors caused by abrupt changes in the velocity profile. Third, the currently fixed mode-switching penalty coefficient will be upgraded to an adaptive penalty strategy driven by fuzzy logic. This enhancement will allow the penalty intensity to dynamically scale according to local obstacle density, thereby further fortifying the algorithm's generalization capabilities across diverse, unstructured scenarios.

7. Conclusions

This study successfully validates the effectiveness of integrating environmental prior dimensionality reduction with a strict driving state machine within a global graph search framework. By mitigating search redundancy and eliminating terminal discrete oscillations, the proposed algorithm provides a highly viable and robust mechanism for the smooth navigation of 4WIS vehicles under complex, low-speed operating conditions. While this purely geometric framework operates under low-speed assumptions and abstracts complex physical limitations such as tire slip, steering actuator dynamics, wheel torque limits, steering transition delays, and vehicle stability constraints, the insights and paradigms established herein lay a solid theoretical foundation for the subsequent transition toward dynamic, real-time autonomous navigation.

Author Contributions: Conceptualization, J.Z. and G.L.; methodology, J.Z. and N.L.; software, J.Z.; validation, J.Z. and D.Z.; formal analysis, J.Z.; investigation, J.Z. and G.L.; resources, G.L. and J.Z.; data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z. and D.Z.; visualization, J.Z. and N.L.; supervision, J.Z. and G.L.; project administration, G.L.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the International Industrial Technology Research and Development Project of Liaoning Province, grant number 2025J101900027.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Lam, T.L.; Qian, H.; Xu, Y. Omnidirectional Steering Interface and Control for a Four-Wheel Independent Steering Vehicle. *IEEE/ASME Trans. Mechatron.* **2010**, *15*, 329–338. [CrossRef]
2. Ye, Y.; He, L.; Zhang, Q. Steering Control Strategies for a Four-Wheel-Independent-Steering Bin Managing Robot. *IFAC-PapersOnLine* **2016**, *49*, 39–44. [CrossRef]
3. Hang, P.; Chen, X. Towards Autonomous Driving: Review and Perspectives on Configuration and Control of Four-Wheel Independent Drive/Steering Electric Vehicles. *Actuators* **2021**, *10*, 184. [CrossRef]
4. Zhang, L.; Zhang, Z.; Wang, Z.; Deng, J.; Dorrell, D.G. Chassis Coordinated Control for Full X-by-Wire Vehicles-A Review. *Chin. J. Mech. Eng.* **2021**, *34*, 42. [CrossRef]
5. Zhu, M.; Chen, H.; Xiong, G. A Model Predictive Speed Tracking Control Approach for Autonomous Ground Vehicles. *Mech. Syst. Signal Process.* **2017**, *87*, 138–152. [CrossRef]
6. Li, M.; Jia, Y. Decoupling and Robust Control of Velocity-Varying Four-Wheel Steering Vehicles with Uncertainties via Solving Attenuating Diagonal Decoupling Problem. *J. Frankl. Inst.* **2017**, *354*, 105–122. [CrossRef]
7. Kim, D.H.; Kim, C.W. Research on Four-wheel Steering-based Mobile Robot Driving Control Strategy to Implement Autonomous Driving Service. *J. Korean Soc. Precis. Eng.* **2024**, *41*, 1009–1015. [CrossRef]
8. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
9. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]

10. Han, Z.; Sun, H.; Huang, J.; Xu, J.; Tang, Y.; Liu, X. Path Planning Algorithms for Smart Parking: Review and Prospects. *World Electr. Veh. J.* **2024**, *15*, 322. [CrossRef]
11. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path Planning for Autonomous Vehicles in Unknown Semi-Structured Environments. *Int. J. Robot. Res.* **2010**, *29*, 485–501. [CrossRef]
12. Yang, J.; Wang, J.; Li, J.; Meng, X.; Jiang, X.; Lu, C. Sobol Sequence RRT* and Numerical Optimal Joint Algorithm-Based Automatic Parking Trajectory Planning of Four-Wheel Steering Vehicles. *Robot. Auton. Syst.* **2025**, *186*, 104909. [CrossRef]
13. Zhao, T.; Yan, Z.; Wang, L.; Pan, R.; Wang, X.; Liu, K.; Guo, K.; Hu, Q.; Chen, S. Hybrid Path Planning Method Based on Skeleton Contour Partitioning for Robotic Additive Manufacturing. *Robot. Comput.-Integr. Manuf.* **2024**, *85*, 102633. [CrossRef]
14. Qin, H.; Jin, Y.; Yang, Z.; Hu, M.; Cui, Q.; Xu, B. Path Planning Method Integrated with Mode Decision for 4WIS Vehicles. *J. Hunan Univ. (Nat. Sci.)* **2024**, *51*, 176–184. [CrossRef]
15. Chu, Y.; Zhao, W.; Zhou, X.; Zhang, Z.; Luan, Z.; Liang, W.; Wang, C.; Sun, H. Multi-Mode Trajectory Planning for Four-Wheel Independent Steering Vehicles Under Emergency Obstacle Avoidance Scenarios. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 14735–14749. [CrossRef]
16. Zhang, L.; Zhang, T.; An, Y.; Niu, T.; Wang, S.; Wang, J. 4WIDS-MAPF: Multiagent Pathfinding for Four-Wheel Independent Drive/Steering Robots. *IEEE Internet Things J.* **2025**, *12*, 38795–38803. [CrossRef]
17. Liu, J.; Yan, F.; Dong, H.; Fu, W.; Chen, Y.; Zhang, S. Optimal Collaborative Path Planning for Four-Wheel Steering Picking Robots Focusing on Improving Efficiency at Bends. *Proc. Inst. Mech. Eng. Part D. J. Automob. Eng.* **2025**, 09544070251378165. [CrossRef]
18. Li, Y.; Li, G.; Wang, X. Research on Trajectory Planning of Autonomous Vehicles in Constrained Spaces. *Sensors* **2024**, *24*, 5746. [CrossRef] [PubMed]
19. Chang, T.; Tian, G. Hybrid A-Star Path Planning Method Based on Hierarchical Clustering and Trichotomy. *Appl. Sci.* **2024**, *14*, 5582. [CrossRef]
20. Xie, G.; Fang, L.; Su, X.; Guo, D.; Qi, Z.; Li, Y.; Che, J. A Path-Planning Approach for an Unmanned Vehicle in an Off-Road Environment Based on an Improved A* Algorithm. *World Electr. Veh. J.* **2024**, *15*, 234. [CrossRef]
21. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica* **1973**, *10*, 112–122. [CrossRef]
22. Reeds, J.; Shepp, L. Optimal Paths for a Car That Goes Both Forwards and Backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [CrossRef]
23. Chen, G.; Yao, J.; Gao, Z.; Gao, Z.; Zhao, X.; Xu, N.; Hua, M. Emergency Obstacle Avoidance Trajectory Planning Method of Intelligent Vehicles Based on Improved Hybrid A*. *SAE Int. J. Veh. Dyn. Stab. NVH* **2024**, *8*, 3–19.
24. Qiao, Y.; Chen, X.; Yin, D. Coordinated Control for the Trajectory Tracking of Four-Wheel Independent Drive–Four-Wheel Independent Steering Electric Vehicles Based on the Extension Dynamic Stability Domain. *Actuators* **2024**, *13*, 77. [CrossRef]
25. Yang, Z.; Dai, Y.; Hu, M.; Bian, Y.; Cui, Q.; Li, Y. Improving Path Tracking Performance of 4WIS Vehicles via Constraint-Oriented Consistent Coordinated Steering. *Chin. J. Mech. Eng.* **2025**, *38*, 158. [CrossRef]
26. Chen, Y.; Liu, Y.; Xu, W. Improved Hybrid A* Algorithm Based on Lemming Optimization for Path Planning of Autonomous Vehicles. *Appl. Sci.* **2025**, *15*, 7734. [CrossRef]
27. Manrique-Cordoba, J.; De La Casa-Lillo, M.Á.; Sabater-Navarro, J.M. N-Dimensional Reduction Algorithm for Learning from Demonstration Path Planning. *Sensors* **2025**, *25*, 2145. [CrossRef] [PubMed]
28. Noël, T.; Lehuger, A.; Marchand, E.; Chaumette, F. Skeleton Disk-Graph Roadmap: A Sparse Deterministic Roadmap for Safe 2D Navigation and Exploration. *IEEE Robot. Autom. Lett.* **2024**, *9*, 555–562. [CrossRef]
29. Klančar, G.; Zdešar, A.; Krishnan, M. Robot Navigation Based on Potential Field and Gradient Obtained by Bilinear Interpolation and a Grid-Based Search. *Sensors* **2022**, *22*, 3295. [CrossRef] [PubMed]