



Geodesic-based path planning for port transfer robots on Riemannian manifolds

Runjiao Bao , Junzheng Wang, Shoukun Wang *

School of Automation, Beijing Institute of Technology, Beijing, China



ARTICLE INFO

Keywords:

Autonomous robot
Path planning
Riemannian manifolds
Geodesic
Geometric heat flow

ABSTRACT

With the rapid intelligent transformation of the automotive industry and the surge in production volume, intelligent autonomous robots equipped with integrated perception and planning systems are playing an increasingly vital role in vehicle transfer operations. Optimizing dispatch paths of robots is essential for improving overall operational efficiency, yet achieving a balance among path length, feasibility, and safety margin remains a significant challenge. To address this issue, we propose a geodesic-based path planning method formulated on Riemannian manifolds. The approach jointly considers directional motion constraints, steering effort, and obstacle accessibility boundaries to construct a Riemannian metric tensor that encodes local path cost structures. This transforms the planning task into a geodesic shortest path problem, which is efficiently solved using the Geometric heat flow (GHF) method. The resulting paths naturally comply with kinematic constraints and exhibit strong obstacle-avoidance capabilities, significantly enhancing safety and executability. Extensive simulations and real-world experiments in high-density port yard environments demonstrate the practicality and robustness of the proposed method under complex spatial constraints and obstacle configurations.

1. Introduction

With the global automotive industry undergoing rapid intelligent transformation and capacity expansion, the throughput of commercial vehicles at ports has been continuously increasing. The traditional roll-on/roll-off (Ro-Ro) operation mode can no longer meet the growing demands for high-efficiency logistics, prompting an urgent shift toward automation. Autonomous robots, which integrate perception, planning, and control systems, are emerging as core equipment for port automation due to their low operational cost and high dispatch efficiency (Xu et al., 2025). A typical workflow of transfer robots receiving task instructions from a centralized scheduling system, autonomously planning a path to the target vehicle, sensing (Kasmaiee et al., 2025b) and avoiding surrounding obstacles (Kasmaiee et al., 2025a), performing automatic docking and lifting, and transporting the vehicle to a designated unloading point (Bao et al., 2025). In practice, inter-yard travel time of robots accounts for over 50 % of the total operation duration, making it a major bottleneck in system efficiency. Particularly in high-density environments such as commercial vehicle yards—with long-distance layouts featuring consecutive intersections and narrow corridors—robots must meet stringent motion constraints while ensuring both feasibility and safety in path planning (Zhang et al., 2025a).

Current path planning approaches can be broadly categorized into four types: search-based, sampling-based, optimization-based, and learning-based algorithms. Search-based methods discretize the state space and find paths through search algorithms. Representative algorithms include Dijkstra (Dijkstra, 1959) and A* (Hart et al., 1968), while Hybrid A* (Dolgov et al., 2008) introduces motion primitives to enhance feasibility. Recent research employs discontinuous motion primitives to improve scalability (Hönig et al., 2022). However, these methods struggle to balance safety, feasibility, and efficiency, typically serving as initial solutions for subsequent optimization.

Sampling-based methods avoid complete space discretization through random sampling. Representative approaches include RRT (LaValle, 1998), PRM (Kavraki et al., 1996), and RRT-Connect (Kuffner & LaValle, 2000). These methods provide only probabilistic completeness without optimality guarantees. RRT* (Karaman & Frazzoli, 2011) achieves asymptotic optimality through iteration, while Dynamic RRT* (Webb & van den Berg, 2013) incorporates dynamic constraints but with low computational efficiency. Recent research focuses on improving iterative search efficiency (Jeong et al., 2019; Wang et al., 2023) and optimizing sampling strategies (Guo et al., 2025). Overall, sampling-based methods can rapidly generate feasible paths but struggle to obtain high-quality solutions within limited computation time.

* Corresponding author.

E-mail addresses: 3120230765@bit.edu.cn (R. Bao), wangjz@bit.edu.cn (J. Wang), bitwsk@bit.edu.cn (S. Wang).

Optimization-based methods formulate path planning as constrained optimization problems, solving for high-quality paths in continuous space through defined objective functions and constraints. Representative approaches include CHOMP (Ratliff et al., 2009), TrajOpt (Schulman et al., 2014), STOMP (Kalakrishnan et al., 2011), and GuSTO (Bonalli et al., 2019), which respectively employ potential field-guided gradient descent, convex optimization decomposition, stochastic sampling enhancement, and controlled linearization strategies. Recent research attempts to integrate bionic optimization (Chai et al., 2019; Huo et al., 2024) and fuzzy reasoning methods (Chai et al., 2024) while optimizing adaptive expansion capabilities (Li et al., 2024). Although these methods excel in trajectory quality and dynamic feasibility, they heavily depend on initial solution quality, are prone to local optima, and may fail to converge in constraint-dense or long-distance environments.

Learning-based path planning has gained increasing attention in recent years. Reinforcement learning optimizes policies through environment interaction and trial-and-error learning, exhibiting strong adaptability and robustness. Representative methods include DRL-VO (Xie & Dames, 2023) and RMRL (Yang et al., 2023). Although RL demonstrates intelligent decision-making abilities, it suffers from high training costs, slow convergence, and sensitivity to reward function design, with policy transfer hindered by the “reality gap”. Supervised learning methods learn mappings from perception inputs to path outputs by mimicking expert demonstrations, such as NeuPAN (Han et al., 2025) and recurrent neural network approach (Chai et al., 2023). However, limitations in feature representation and generalization capability still hinder the applicability of most learning-based approaches in long-range planning tasks.

Despite the diversity of existing approaches, their applicability to port scenarios remains limited. Search-based and sampling-based methods often neglect kinematic constraints or yield suboptimal trajectories, which is problematic in long-distance navigation across yards with narrow lanes and frequent intersections. Optimization-based approaches can generate high-quality paths but incur prohibitive computational costs in large-scale, constraint-intensive layouts typical of vehicle yards. Learning-based methods demonstrate adaptability in dynamic scenes, yet their reliance on extensive training data and difficulty in transferring policies to real-world port operations hinder practical deployment (Kasmaiee et al., 2024). These limitations underscore the need for a unified geometric framework that inherently guarantees feasibility and safety. Riemannian manifolds provide such a foundation, which naturally motivates their use in our path planning approach for port robots.

Recent studies have demonstrated that Riemannian manifolds, through spatiotemporal geometric transformations, can unify diverse constraints and objectives by mapping them onto a differential manifold structure, thereby gaining widespread application in various robot path planning tasks. For instance, in three-dimensional spatial scenarios, geodesic modeling enables unmanned aerial vehicles to efficiently traverse complex airspaces (Huang et al., 2015). Combining Riemannian manifold methods with sampling techniques allows robots to autonomously avoid unstable regions even in scenarios with uneven fields in multiple dimensions (Zhang et al., 2025b). In autonomous driving scenarios, constructing dynamic manifolds by integrating spatiotemporal vehicle information can generate trajectories that balance trackability and collision avoidance (Liu et al., 2022). For high-dimensional systems, introducing manifold modeling and embedding motion constraints effectively facilitates path planning for robotic arms (Laux & Zell, 2021). Moreover, this method has been extended to applications under boundary constraints (Laux & Zell, 2023) and prevents joint collisions by incorporating obstacle functions into the metric (Klein et al., 2023). Beyond single-path planning, Riemannian manifold methods are also suitable for more complex problems, such as serving as reference paths for multi-robot systems (Kim et al., 2013) or heterogeneous teams comprising robots and humans, to explore unknown or partially known environments in search and rescue operations (Govindarajan et al., 2016). Additionally, combining neighborhood-augmented graphs with Rieman-

nian manifolds enables the shortest feasible path planning for complex tethered robot systems (Sahin & Bhattacharya, 2024).

A review of related research reveals that the core advantage of path planning schemes based on Riemannian manifolds lies in their geometric intuitiveness, which enables unified modeling of multi-source constraints and enhances path executability and safety. Therefore, designing appropriate Riemannian metrics according to specific scenarios and task objectives, along with efficiently solving geodesics, represents the key to applying this method. To achieve robot path planning in port environments, this paper introduces Riemannian manifolds, synergistically considering motion direction constraints, steering costs, and obstacle reachability boundaries to construct a unified Riemannian metric, and provides specific solution methods along with experimental validation. The main contributions of this research include the following aspects:

- 1) Constructed a Riemannian measure that synergistically considers robot motion constraints and control variable costs, integrating constraint conditions and costs under different metrics into a unified metric representation.
- 2) Accurately modeling obstacles in port environments through potential field methods, and incorporating these potential fields as coefficients in the Riemannian metric. This design ensures that obstacle avoidance is given intrinsic priority during path planning, while remaining compatible with other motion and control objectives.
- 3) Introduced the Geometric heat flow (GHF) method, efficiently transforming the geodesic solving problem into partial differential equation solving.

The rest of this article is organized as follows. Section 2 elaborates on background knowledge regarding methods and scenarios. Section 3 provides detailed introduction of the proposed path planning method. Section 4 presents extensive experimental results with comparative analysis against benchmark methods. Finally, Section 5 draws conclusions.

2. Preliminaries

2.1. Working environment and background

Ro-Ro terminals typically consist of several functional zones, including loading and unloading areas, storage yards, and multi-level parking structures. The loading and unloading areas are designated for the docking and handling of import and export vehicles, while the storage yards and parking structures are responsible for the temporary storage and classification management of commercial vehicles, as illustrated in Fig. 1. At present, vehicle transportation between these zones primarily relies on manual driving, which results in high labor costs, low operational efficiency, and significant carbon emissions. These limitations hinder the realization of efficient, green, and automated operations, which are essential to the development of intelligent ports.

To enhance port intelligence, the deployment of robots with self-operating capabilities has emerged as a key direction. Robot systems are required to execute transportation tasks for commercial vehicles



Fig. 1. Illustration of a Ro-Ro Yard operation scene for vehicle storage.

across multiple regions and target points within a complex port environment. The operational workflow involves the cloud-based scheduling system, which assigns tasks according to the port's production management system and performs unified scheduling and global path planning for multiple robots, generating navigation routes that cover the entire port area. Subsequently, onboard control systems execute local path planning based on the global route and real-time perception data, guiding the robots to accurately approach the vehicle pickup location and sequentially perform vehicle identification, docking, handling, and unloading tasks.

It is worth emphasizing that the port environment exhibits highly structured geometric characteristics. The main layout consists of storage blocks and trunk roads, where obstacle regions are predominantly axis-aligned rectangular structures. The overall road network can be abstracted as a grid topology formed by intersecting bidirectional traffic lanes. Such geometric structuring imposes specific constraints on path planning, including frequent right-angle turns and narrow passage navigation, necessitating specialized algorithmic designs to adapt to these strongly structured spatial features.

2.2. Robot Kinematic constraint modeling

To ensure the physical feasibility and safety of the generated path, it is essential to establish a kinematic model that considers the robot structure, control constraints, and spatial occupancy characteristics.

In terms of steering control strategy, we adopt an angular coordination mechanism based on the Ackermann steering geometry. This principle requires that all wheels' steering extensions converge to a common instantaneous center of rotation, thereby reducing tire slip and steering load. Fig. 2 shows the kinematic model of the transfer robot. Given that the robot is equipped with a four-wheel steering (4WS) system, we introduce additional constraints to ensure steering symmetry and velocity coordination:

$$\begin{cases} \delta_{fl} = -\delta_{rl}, \quad \delta_{fr} = -\delta_{rr} \\ v_{fl} = v_{rl}, \quad v_{fr} = v_{rr} \\ \cot \delta_{fr} - \cot \delta_{fl} = \frac{2W}{L} \\ v_{fl} \sin \delta_{fl} = v_{fr} \sin \delta_{fr} \end{cases} \quad (1)$$

where $\{\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}\}$ denote the steering angles of the front-left, front-right, rear-left, and rear-right wheels, respectively, and $\{v_{fl}, v_{fr}, v_{rl}, v_{rr}\}$ represent the translational velocities of the corresponding wheels under a no-slip condition. L and W denote the wheelbase and track width, respectively.

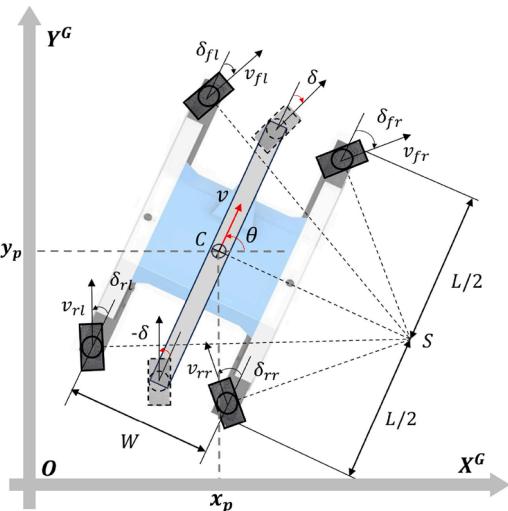


Fig. 2. Kinematic model of the 4WS robot.

Although the robot follows a complex 4WS kinematic model, we employ a simplified two-degree-of-freedom bicycle model for path planning. Port robots typically operate at moderate speeds on structured, high-friction surfaces, where tire slip is negligible, so the bicycle model captures the essential dynamics while greatly reducing computational complexity. This model can be expressed as:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \frac{2v}{L} \tan \delta, \end{cases} \quad (2)$$

where (x, y) denotes the position of the geometric center of the robot in the global coordinate system, θ is the heading angle, v is the equivalent linear velocity, and δ is the simplified equivalent front-wheel steering angle. Trajectories generated with this model can be faithfully executed on the full 4WS system, as the equivalent steering angle δ can be uniquely mapped to the actual steering angles and wheel velocities via the Ackermann constraints.

Based on the Ackermann geometry, the actual steering angles and wheel speeds of the left and right front wheels can be derived as:

$$\begin{cases} \cot \delta_{fl} = \cot \delta - \frac{W}{L}, \quad \cot \delta_{fr} = \cot \delta + \frac{W}{L} \\ v_{fl} = \frac{v \tan \delta}{\sin \delta_{fl}}, \quad v_{fr} = \frac{v \tan \delta}{\sin \delta_{fr}}. \end{cases} \quad (3)$$

The inputs for the rear wheels can be directly derived from the front-wheel inputs based on the constraints in Eq. (1) and are therefore omitted here for brevity.

For further path planning modeling, we define the state and control variables. The robot state vector is $X = [x, y, \theta, v]^T$, and the control input vector is $U = [a, \omega]^T$, where a represents linear acceleration and ω the steering rate. To comply with physical and actuator limitations, the state and control inputs must satisfy the following bounds:

$$X_{min} \leq X \leq X_{max}, \quad U_{min} \leq U \leq U_{max}. \quad (4)$$

To account for environmental obstacles, the actual occupied area of the robot in the 2D space must also be described. Let W denote the local shape matrix of the robot, $M(x)$ the rotation matrix representing its orientation, and $T(x)$ the translational transformation for its position. Then, the occupied region of the robot \mathcal{V} is:

$$\mathcal{V}(X) = M(X)W + T(X). \quad (5)$$

To avoid collisions with the set of static obstacles \mathcal{O} , the generated path must satisfy the non-intersection condition:

$$\mathcal{V}(X) \cap \mathcal{O} = \emptyset. \quad (6)$$

The aforementioned modeling framework reveals three key challenges that motivate our methodological approach. The highly structured port environment with axis-aligned obstacles and grid-like topology creates natural geometric constraints. Additionally, the steering constraints introduce complex interdependencies between position and orientation. Furthermore, the concurrent requirements of path smoothness, dynamic feasibility, and collision avoidance necessitate a unified framework capable of embedding these heterogeneous objectives into a coherent geometric representation.

2.3. Geometric modeling of constrained manifold

In our application scenario, the planning problem would benefit from a geometric framework that can: encode complex constraints through metric design, generate naturally smooth trajectories, and provide unified treatment of spatial and kinematic constraints. This motivates our adoption of Riemannian manifold-based path planning.

A manifold is a topological space \mathcal{M} that is locally Euclidean. That is, for any point $p \in \mathcal{M}$, there exists a neighborhood $U \subseteq \mathcal{M}$ that is

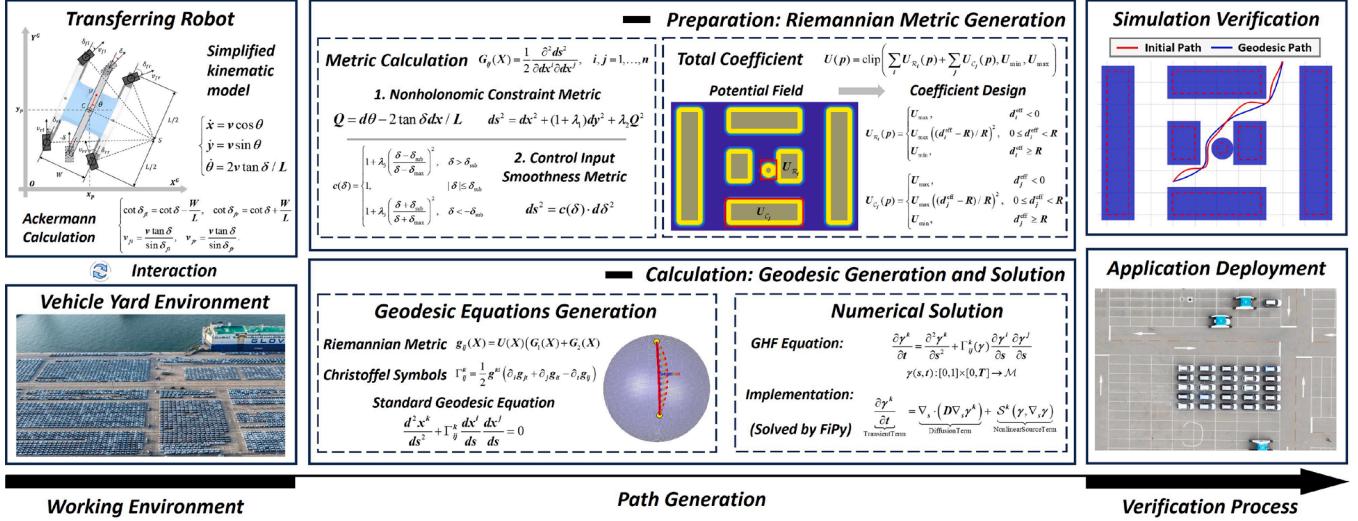


Fig. 3. Overall architecture of the Riemannian manifold-based geodesic path planning framework.

homeomorphic to an open subset of a d -dimensional Euclidean space. In this case, \mathcal{M} is referred to as a d -dimensional manifold. A Riemannian manifold is a specific type of manifold where a Riemannian metric endows each tangent space $T_q\mathcal{M}$ with an inner product structure. For any $q \in \mathcal{M}$, the metric tensor $G(q)$ in local coordinates is a symmetric positive-definite matrix. Specifically, the Riemannian inner product of two tangent vectors $u, v \in T_q\mathcal{M}$ at point q is defined as:

$$\langle u, v \rangle_q = u^\top G(q)v = \sum_{i,j} g_{ij}(q)u^i v^j, \quad \forall u, v \in T_q\mathcal{M}. \quad (7)$$

The Riemannian metric introduces notions of length and angle on the tangent space, enabling measurement of distances and angles on the manifold. Analogous to straight lines in Euclidean space, geodesics are locally length-minimizing curves on Riemannian manifolds. One key property of geodesics is that, under uniform motion on a Riemannian manifold, the acceleration vector has no component tangent to the surface—i.e., the velocity vector X' is orthogonal to the acceleration vector X'' .

By incorporating the Riemannian manifold into the planning problem, we can design customized metric tensors that elegantly capture both the structured characteristics of the port environment and the kinematic properties of the robot (Fig. 3).

3. Methodology

Our objective is to leverage the natural smoothness and physical plausibility of geodesic motion to ensure trajectory feasibility and trackability. In three-dimensional Minkowski spacetime, geodesics reduce to uniform straight-line motion, which is inadequate for path planning in complex environments. To address this, we propose constructing a novel spatiotemporal geometric framework where the local geometry dynamically adapts to the vehicle's state, unifying kinematic constraints, obstacle avoidance, and path optimality within a single Riemannian framework.

The central component of this framework is a composite metric tensor that simultaneously encodes motion feasibility and spatial constraints, enabling geodesic paths to naturally satisfy all requirements while inherently avoiding obstacles and adhering to kinematic constraints. The design of each metric tensor, their combination into a unified structure, and the use of the GHF for geodesic computation constitute the key innovations of our approach.

3.1. Motion constraint metric design

3.1.1. Nonholonomic constraint metric

As described in Section 2.2, the nonholonomic constraints of the robot based on the bicycle model include the no lateral slip condition and a kinematic constraint that couples the robot's heading dynamics with its steering geometry (Yusheng et al., 2020). In the robot's local coordinate frame, this constraint system can be formally expressed as:

$$\begin{cases} \dot{y} = 0 \\ \dot{\theta} = \dot{x} \frac{2\tan\delta}{L} \end{cases} \quad (8)$$

To soften these hard constraints, we define an error metric representing the deviation from ideal kinematics:

$$Q = d\theta - 2 \tan\delta dx/L. \quad (9)$$

The state space formed by the key variables $\{x, y, \theta, \delta\}$ can be viewed as a smooth manifold equipped with a global chart, where each point corresponds to a specific robot state. To capture richer trajectory costs on this manifold, we augment the traditional path-length-based metric with a geometry-aware cost related to the nonholonomic error. Specifically, we define the following local distance function on the Riemannian manifold:

$$ds^2 = \sum_{i,j=0}^n g_{ij}dx^i dx^j = dx^2 + (1 + \lambda_1)dy^2 + \lambda_2 Q^2, \quad (10)$$

where g_{ij} denotes the components of the metric tensor, dx^i and dx^j are the coordinate differentials, n is the dimension of the manifold, and λ_1 and λ_2 are penalty coefficients for violating the nonholonomic constraints. The tensor components g_{ij} in this local coordinate representation directly constitute the elements of the metric tensor matrix G'_1 as:

$$G'_1 = \begin{bmatrix} \frac{L^2 + 4\lambda_2(\tan\delta)^2}{L^2} & 0 & -\frac{4\lambda_2\tan\delta}{L} & 0 \\ 0 & 1 + \lambda_1 & 0 & 0 \\ -\frac{4\lambda_2\tan\delta}{L} & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

To ensure the positive definiteness of the metric, appropriate values of the penalty coefficients should be chosen. Next, we transform the Riemannian manifold defined in the local coordinate frame into the global

coordinate frame. The transformation matrix between local and global frames is:

$$R = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (12)$$

Accordingly, the metric tensor in the global coordinate frame can be obtained by:

$$G_1 = R^T G'_1 R. \quad (13)$$

This formulation enables the integration of nonholonomic motion constraints into the Riemannian geometry-based trajectory planning framework.

3.1.2. Control smoothness metric

To ensure trajectory smoothness, we generally seek to avoid excessively large steering angles during path planning. Therefore, we introduce a steering cost into the local Riemannian distance formulation as follows:

$$ds^2 = c(\delta) \cdot d\delta^2, \quad (14)$$

where the penalty coefficient λ_3 is defined as:

$$c(\delta) = \begin{cases} 1 + \lambda_3 \left(\frac{\delta - \delta_{\text{sub}}}{\delta - \delta_{\text{max}}} \right)^2, & \delta > \delta_{\text{sub}} \\ 1 + \lambda_3 \left(\frac{\delta + \delta_{\text{sub}}}{\delta + \delta_{\text{max}}} \right)^2, & \delta < -\delta_{\text{sub}} \\ 1, & |\delta| \leq \delta_{\text{sub}}, \end{cases} \quad (15)$$

where δ_{max} denotes the maximum allowable steering angle, δ_{sub} is the threshold at which the penalty is triggered and λ_3 is a penalty coefficient that controls the sensitivity of the cost function to large steering angles. As δ approaches δ_{max} , $c(\delta)$ increases nonlinearly toward infinity, effectively discouraging the steering angle from exceeding its physical limit.

To ensure numerical stability, the initial guess of δ is constrained within a reasonable range to avoid large gradients near extreme values. During iteration, δ is limited to $[\delta_{\text{min}} + \epsilon, \delta_{\text{max}} - \epsilon]$ with a small margin ϵ (e.g., 10^{-3}) to prevent the denominator in $c(\delta)$ from approaching zero. Values exceeding this range are truncated each step, maintaining stability while preserving the penalty's physical meaning.

Accordingly, the metric tensor component corresponding to control smoothness is:

$$G_2 = \text{diag}(0, 0, 0, c(\delta)). \quad (16)$$

3.2. Obstacle constraint coefficient design

3.2.1. Three-circle spatial approximation

To simplify geometric modeling and enhance computational efficiency, we adopt a three-circle approximation model to represent the robot's physical footprint. Specifically, three equal-radius disks are placed along the longitudinal symmetry axis of the robot, corresponding to the front, middle, and rear structural regions. This configuration ensures complete coverage of the robot's physical occupancy while minimizing redundancy. As shown in Fig. 4, the intersections between adjacent circles lie precisely at the lateral edges of the robot, ensuring geometric compactness and symmetry.

The global coordinates of the centers of the three circles are defined as:

$$\begin{cases} O_f^x = x + \frac{L}{6} \cos \theta, & O_f^y = y + \frac{L}{6} \sin \theta \\ O_m^x = x, & O_m^y = y \\ O_r^x = x - \frac{L}{6} \cos \theta, & O_r^y = y - \frac{L}{6} \sin \theta, \end{cases} \quad (17)$$

where O_f, O_m, O_r represent the centers of the front, middle, and rear circles, respectively, and (x, y, θ) denote the current position and heading

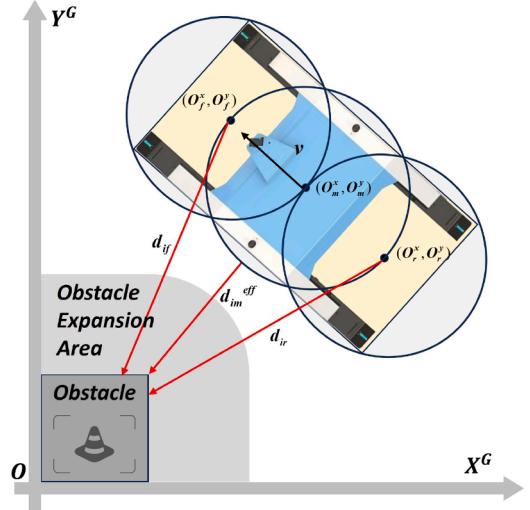


Fig. 4. The collision model between robot and obstacles.

angle of the robot. The radius of each circle is defined as:

$$r_c = \sqrt{(L/6)^2 + (W/2)^2}, \quad (18)$$

ensuring that the entire robot outline is covered without blind spots.

Based on the three-circle model, the spatial occupancy region of the robot, denoted $\mathcal{V}(X)$, is given by the union of the three circular regions:

$$\mathcal{V}(X) = \bigcup_{k \in \{f, m, r\}} C_k, C_k = \{p \in \mathbb{R}^2 \mid \|p - p_k\| \leq r_c\}. \quad (19)$$

Under this formulation, the obstacle avoidance constraint is formalized by requiring that the robot's occupied region does not intersect with the set of obstacles \mathcal{O} :

$$\mathcal{V}(X) \cap \mathcal{O} = \emptyset \Leftrightarrow C_k \cap \mathcal{O} = \emptyset, \forall k \in \{f, m, r\}. \quad (20)$$

3.2.2. Obstacle avoidance potential field metric

After establishing the motion constraint metrics and control smoothness metrics, we now need to design obstacle constraint coefficients and integrate them into a unified metric tensor to achieve collision-free navigation.

In practical trajectory generation tasks, the spatial relationship between robots and obstacles significantly affects the feasibility and safety of paths. To enhance obstacle avoidance robustness and dynamic reachability, it is necessary to introduce geometry-related obstacle avoidance potential coefficients into the geodesic energy functional, enabling trajectories to naturally avoid obstacles.

We employ a composite potential field model to achieve this objective, which integrates the boundary diffusion mechanism of rectangular obstacles with the radial decay characteristics of circular obstacles, adapting to complex structured environments. Specifically, given a set of rectangular obstacles $\mathcal{R}_i = [x_{\min}^{(i)}, x_{\max}^{(i)}] \times [y_{\min}^{(i)}, y_{\max}^{(i)}]$ and a set of circular obstacles $C_m = (c_m, r_m)$ in two-dimensional space, the total potential field at any spatial point $p = (x, y)$ is defined as:

$$U(p) = \sum_{i=1}^{N_r} U_i^{\text{rect}}(p) + \sum_{m=1}^{N_c} U_m^{\text{circle}}(p), \quad (21)$$

where N_r denotes the number of rectangular obstacles and N_c denotes the number of circular obstacles.

The nearest distance from point p to the i -th rectangular obstacle is calculated as follows:

$$d_i = \left[\max(0, x_{\min}^{(i)} - x)^2 + \max(0, x - x_{\max}^{(i)})^2 + \max(0, y_{\min}^{(i)} - y)^2 + \max(0, y - y_{\max}^{(i)})^2 \right]^{1/2}. \quad (22)$$

Defining the effective distance as $d_i^{\text{eff}} = d_i - R$, the potential energy function contributed by this obstacle is:

$$U_{R_i}(\mathbf{p}) = \begin{cases} U_{\max}, & d_i^{\text{eff}} < 0 \\ U_{\max} \left(\frac{d_i^{\text{eff}} - R}{R} \right)^2, & 0 \leq d_i^{\text{eff}} < R \\ U_{\min}, & d_i^{\text{eff}} \geq R, \end{cases} \quad (23)$$

where U_{\max} represents the maximum penalty for penetrating obstacles, and U_{\min} represents the minimum coefficient in obstacle-free regions.

Similarly, for each circular obstacle with center $\mathbf{c}_j = [x_j, y_j]^T$ and radius r_j , the effective distance to point \mathbf{p} is:

$$d_j^{\text{eff}} = \|\mathbf{p} - \mathbf{c}_j\| - r_j - R. \quad (24)$$

The corresponding potential energy function is:

$$U_{C_j}(\mathbf{p}) = \begin{cases} U_{\max}, & d_j^{\text{eff}} < 0 \\ U_{\max} \left(\frac{d_j^{\text{eff}} - R}{R} \right)^2, & 0 \leq d_j^{\text{eff}} < R \\ U_{\min}, & d_j^{\text{eff}} \geq R. \end{cases} \quad (25)$$

The overall obstacle potential field is defined as the superposition of potential energies contributed by all obstacles. To avoid numerical instability and ensure computational efficiency, we apply clipping to the potential values to prevent numerical overflow in extreme cases:

$$U(\mathbf{p}) = \text{clip} \left(\sum_i U_{R_i}(\mathbf{p}) + \sum_j U_{C_j}(\mathbf{p}), U_{\min}, U_{\max} \right). \quad (26)$$

This potential function is continuously differentiable and forms gradient-ascending repulsive barriers near obstacle boundaries, facilitating obstacle avoidance behavior during path planning.

Based on the three-circle approximate model described in A, the overall obstacle avoidance potential energy of the robot is mapped to the superposition of the potential fields of each disk:

$$\hat{U}(X) = \sum_{k \in \{f, m, r\}} U(\mathbf{p}_k). \quad (27)$$

With both motion and obstacle constraints formulated as metric components, we can now construct geodesic paths through a unified geometric framework, allowing path optimization while respecting all constraints.

3.3. Geodesic generation and solution

3.3.1. Energy functional and standard formulation of geodesic equations

As introduced in Section 2.3, geodesics characterize locally shortest paths on smooth Riemannian manifolds \mathcal{M} . Based on the aforementioned construction, we define the total Riemannian metric as:

$$g_{ij}(X) = \hat{U}(X)(G_1(X) + G_2(X)). \quad (28)$$

This design ensures collision avoidance as the top priority by using the obstacle potential as a scalar modulation of the Riemannian metric. In dense obstacle regions, obstacle potential dominates by relaxing kinematic constraints, while in free space, kinematic and control costs take precedence. Compared with directly embedding the potential into the metric, the global scaling design offers better numerical stability and computational efficiency.

Consider a parameterized curve $\gamma(s) = \{x^i(s)\}$ connecting two points on the manifold. Its path length functional over the interval $s \in [a, b]$ can be simplified as:

$$\mathcal{L}[\gamma] = \int_a^b \sqrt{g_{ij}(x(s)) \frac{dx^i}{ds} \frac{dx^j}{ds}} ds \approx \frac{1}{2} \int_a^b g_{ij}(x) \frac{dx^i}{ds} \frac{dx^j}{ds}. \quad (29)$$

The functional \mathcal{L} represents a manifold energy functional that quantifies the geometric cost associated with a curve embedded in the curved space. This formulation extends the classical notion of arc length from

Euclidean geometry to general Riemannian manifolds, where the local geometry determines the relative cost of different path directions.

By applying the Euler-Lagrange equation to the energy functional \mathcal{L} , and letting $\dot{x}^k = dx^k/ds$, we obtain:

$$\frac{d}{ds} \left(g_{kj} \frac{dx^j}{ds} \right) - \frac{1}{2} \partial_k g_{ij} \frac{dx^i}{ds} \frac{dx^j}{ds} = 0. \quad (30)$$

Utilizing the symmetry of the metric tensor, we introduce the Christoffel symbols Γ_{ij}^k , defined as:

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl} (\partial_l g_{jl} + \partial_j g_{il} - \partial_l g_{ij}), \quad (31)$$

where g^{kl} denotes the inverse of the metric tensor g . The Christoffel symbols encode the intrinsic curvature information of the manifold and serve as correction terms for covariant differentiation in curved spaces. They characterize how the coordinate basis vectors change from point to point, effectively capturing the non-Euclidean nature of the underlying geometry.

Substituting into the equation yields the standard form of the geodesic equation:

$$\frac{d^2 x^k}{ds^2} + \Gamma_{ij}^k \frac{dx^i}{ds} \frac{dx^j}{ds} = 0. \quad (32)$$

This equation system represents a set of second-order ordinary differential equations that describe the behavior of a curve following a geodesic path under the affine parameter s .

3.3.2. Numerical solution implementation

To efficiently compute geodesic paths under the constructed Riemannian metric tensor $g_{ij}(X)$, we adopt a numerical method based on Geometric Heat Flow (GHF) (Colding et al., 2015). It is worth noting that while GHF itself is a well-established method, its application to geodesic computation represents a novel idea. The key idea is to interpret path optimization as a heat diffusion process on the manifold, where an initial curve is gradually smoothed toward the geodesic.

This method introduces a pseudo-time variable t , which does not represent physical time but instead acts as an artificial evolution index that gradually smooths the curve toward a geodesic. In contrast, s is the normalized arc-length parameter along the curve, serving as the spatial index of points on the evolving path. By introducing t , the rigid boundary value problem is transformed into a parabolic partial differential equation. The core formulation is:

$$\frac{\partial \gamma^k}{\partial t} = \frac{\partial^2 \gamma^k}{\partial s^2} + \Gamma_{ij}^k(\gamma) \frac{\partial \gamma^i}{\partial s} \frac{\partial \gamma^j}{\partial s}, \quad (33)$$

where $\gamma(s, t) : [0, 1] \times [0, T] \rightarrow \mathcal{M}$ denotes the evolving path. The boundary conditions are imposed in Dirichlet form, where the start and goal states are enforced as fixed constraints throughout the evolution:

$$\gamma(0, t) = X_{\text{start}}, \quad \gamma(1, t) = X_{\text{goal}}, \quad \forall t \geq 0. \quad (34)$$

where X_{start} and X_{goal} represent the initial and target configurations in the robot's configuration space, respectively. This ensures that the path optimization strictly adheres to the task requirements at both ends. When the time derivative vanishes, the equation naturally degenerates into the standard geodesic equation, and the steady-state solution corresponds to the desired geodesic. Fig. 5 illustrates the path evolution process under the Geometric Heat Flow.

Compared with traditional Runge-Kutta methods, the geometric heat flow approach eliminates the need for accurately guessing the initial velocity direction. This significantly improves robustness to initial conditions. Furthermore, the diffusion nature of the parabolic PDE effectively suppresses numerical error accumulation and mitigates divergence in high-curvature regions. The monotonic energy decay property of geometric heat flow ensures theoretical convergence during evolution.

In the specific solution process, we employ the A* algorithm to generate an initial path, followed by resampling techniques to precisely align it with the GHF discretization grid. The robot's heading angles are

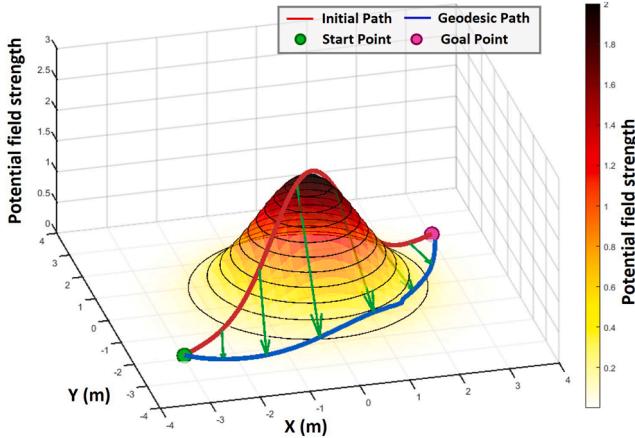


Fig. 5. Path evolution process under geometric heat flow.

computed from directional changes between adjacent waypoints, while the corresponding wheel steering angles are derived through inverse kinematic relationships. Although this initial path may not fully satisfy strict kinematic constraints, it provides a reasonable initial configuration for the GHF iterative optimization process, significantly enhancing the solver's convergence efficiency and computational performance.

4. Experiments

To verify the feasibility and effectiveness of the proposed path planning method, we conducted multi-scenario tests in a simulation environment and verified the effect in actual scenarios.

4.1. Experimental preparation

To evaluate the performance of the proposed path planning method, simulation studies were performed on a workstation equipped with an Intel i7-13700H CPU and an RTX 4060 GPU. The robot parameters, summarized in **Table 1**, were based on real robot specifications to ensure practical relevance. Key algorithm configurations are listed in **Table 2**. The numerical solution of the GHF equations is carried out using FiPy (Guyer et al., 2009), which supports explicit construction of transient, diffusion, and nonlinear source terms via a finite volume framework.

To benchmark performance, three representative baseline algorithms—A*, RRT*, and Hybrid A*—were implemented under the same simulation conditions. These comparisons highlight the advantages of the proposed method in terms of path smoothness, obstacle avoidance, and compliance with kinematic constraints. For real-world validation, the proposed planner was deployed on transfer robots operating in a port yard. The experimental results confirm the feasibility and practical applicability of the generated paths.

In terms of performance evaluation, we establish a comprehensive indicator system encompassing five key metrics for port robot path planning. Path length directly reflects transportation efficiency and energy consumption; maximum and average curvature quantify the geometric

Table 1
Transfer robot geometry parameters.

Parameters	Description	Value
W , mm	Vehicle width	3000
L , mm	Vehicle length	6000
d , mm	Wheelbase	2570
b , mm	Track width	5438
η_{\max} , rad	Max heading angle	$\pi/4$
r_c , m	Min turning radius	1.802

Table 2
Key parameters for path planning method.

Parameters	Description	Values
λ_1	Constraint penalty coefficients	30
λ_2	Constraint penalty coefficients	30
λ_3	Steering penalty coefficients	10
U_{\max}	Max Potential field value	100
U_{\min}	Min Potential field value	1

smoothness of paths; control cost is obtained by calculating the cumulative change in heading angle after uniform resampling, where larger values indicate more severe steering changes and higher control difficulty; minimum and average obstacle distances assess safety margins from local and global perspectives; feasibility evaluates practical achievability by calculating the proportion of path points where curvature exceeds kinematic constraint thresholds.

This multi-dimensional evaluation framework comprehensively characterizes algorithm performance from the dimensions of efficiency, safety, executability, and practicality, fully covering the core performance requirements of port robot path planning. Compared to traditional single-indicator evaluation methods, it provides a more objective and comprehensive assessment of algorithm applicability and effectiveness in actual port environments.

4.2. Simulation experiments

To comprehensively evaluate the performance of the proposed algorithm under different environmental conditions, simulation experiments were conducted in two representative scenarios: spacious and narrow environments. These scenarios evaluate the algorithm's adaptability under varying environmental complexity. For each scenario, 10 independent experiments were performed to examine the consistency and robustness of the algorithm.

4.2.1. Spacious scene experiments

As illustrated in **Fig. 6**, the simulation tests encompass three typical port operation scenarios: continuous alternating turns (Scenario 1), continuous unidirectional turns (Scenario 2), and compound turns (Scenario 3). **Table 3** summarizes the key performance indicators, with best results in bold and second-best underlined. Subsequent performance tables follow the same annotation convention.

Simulation results indicate that the proposed algorithm delivers superior overall performance. While both A* and RRT* theoretically yield shortest paths, A* tends to produce longer paths in practice due to cumulative errors caused by the limited resolution of grid-based maps. In addition, its inherent search mechanism leads to higher path curvature. RRT* outperforms A* in most metrics, particularly in smoothness and feasibility, yet both algorithms generate partially unexecutable paths. Hybrid A* integrates vehicle kinematic constraints, producing smoother and more executable paths, with a maximum curvature over 40% lower than that of A* and RRT*. However, its heuristic often induces unnecessary initial displacements toward the goal, bringing paths closer to obstacles and triggering reverse maneuvers in turning segments, which significantly increases path length.

By contrast, the proposed geodesic-based method achieves an effective trade-off between path length and smoothness. Although slightly longer than A* and RRT* in distance, it attains smoothness and feasibility comparable to Hybrid A* and markedly surpasses A* and RRT*. More critically, it provides enhanced obstacle avoidance, with a significantly greater minimum obstacle distance, indicating superior safety margins and robustness in open environments.

4.2.2. Narrow scene experiments

To comprehensively evaluate the adaptability of the proposed path planning algorithm in complex environments, we constructed representative test scenarios featuring restricted navigable space. Scenario

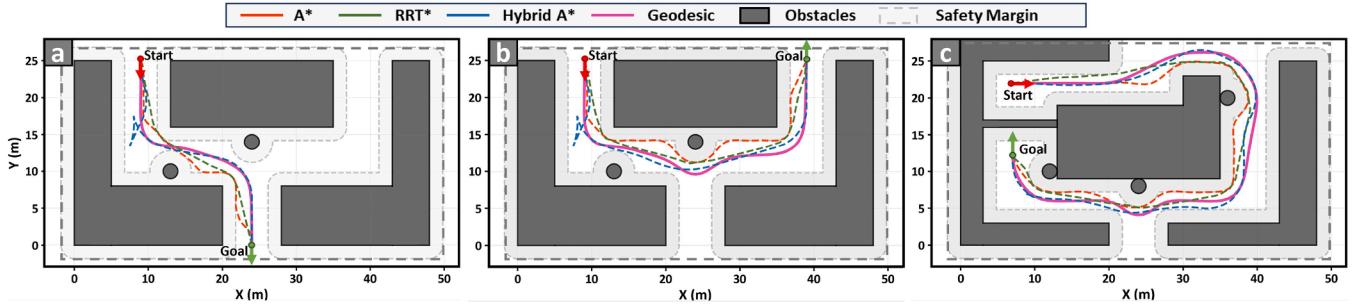


Fig. 6. Navigation scenarios in spacious environments. (a) Consecutive alternating turns: left-right-left turning sequences. (b) Consecutive unidirectional turns: same-direction turning sequences. (c) Compound turns: mixed turning patterns.

Table 3

Comparison of path planning methods in spacious scenarios.

Scenarios	Methods	Path Length	Max curv.	Mean curv.	Control Cost	Mean Obs. Dist.	Min Obs. Dist.	Feasibility
Scenario 1	A*	<u>32.859</u>	1.2466	0.2316	7.2655	2.6256	1.7930	78 %
	RRT*	<u>31.573</u>	0.9003	<u>0.0811</u>	2.4070	2.7681	1.8070	90 %
	Hybrid A*	42.922	0.2689	<u>0.0663</u>	4.1617	3.4487	1.9576	100 %
	Ours	34.686	<u>0.3650</u>	0.0844	<u>2.9827</u>	3.2993	2.3635	100 %
Scenario 2	A*	<u>49.025</u>	0.9016	0.1968	9.3951	2.3376	1.7930	91 %
	RRT*	<u>47.251</u>	0.9772	<u>0.0765</u>	3.4457	3.2594	<u>1.8022</u>	96 %
	Hybrid A*	58.615	0.3701	<u>0.0646</u>	5.4918	2.7825	1.7176	100 %
	Ours	52.039	<u>0.3855</u>	0.0957	<u>5.2902</u>	3.2709	2.3572	100 %
Scenario 3	A*	<u>80.842</u>	0.5902	0.1292	10.5443	2.2682	1.6128	96 %
	RRT*	<u>79.769</u>	1.0269	<u>0.0891</u>	7.0165	2.4648	1.8224	97 %
	Hybrid A*	87.059	<u>0.3701</u>	<u>0.0928</u>	9.1693	2.9235	<u>1.9367</u>	100 %
	Ours	85.405	0.3448	0.0938	<u>8.3172</u>	3.2709	2.6754	100 %

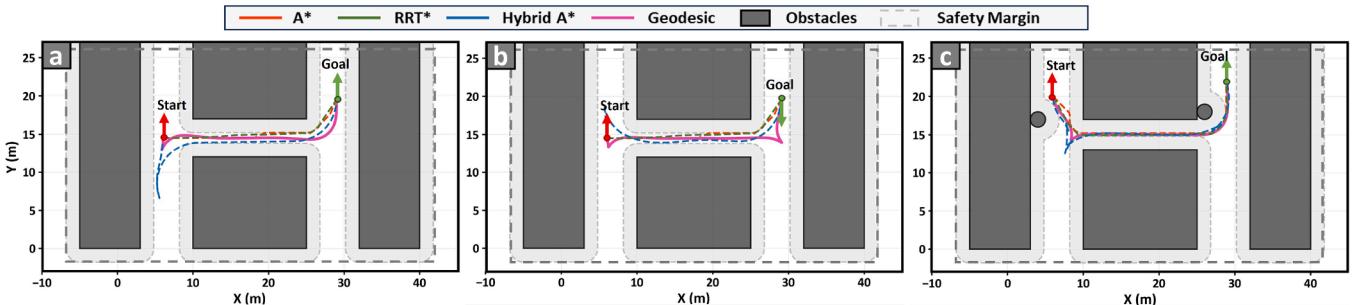


Fig. 7. Navigation scenarios in narrow environments. (a) Forward navigation: passing narrow corridors and executing turning maneuvers. (b) Reverse parking: passing narrow corridors and performing backward U-turns. (c) Constrained navigation: narrower corridors combined with additional obstacles.

1 simulates a robot proceeding forward after traversing a constricted path, Scenario 2 involves the robot reversing into a target location after passing through a constricted path, and Scenario 3 further reduces the width of the narrow corridor based on Scenario 2 while introducing additional obstacles, as illustrated in Fig. 7. Key performance metrics of all compared planning algorithms under these scenarios are summarized in Table 4.

Simulation results reveal fundamental limitations of A* and RRT* algorithms. Both algorithms fail to account for robot orientation constraints: they generate identical paths for Scenarios 1 and 2, which is clearly unreasonable since forward and reverse maneuvers should require different optimal strategies. More critically, the generated paths exhibit unacceptable kinematic violations, including physically impossible operations such as lateral movements and significant terminal orientation errors. Since the paths are inherently unexecutable, discussing their control costs is meaningless. Therefore, relevant data for these methods are omitted from the table in certain scenarios.

In Scenario 1, Hybrid A* produces a clearly unnecessary detour, while in Scenario 2, it generates a long reverse segment that contra-

dicts typical operational logic and compromises execution efficiency. It is important to note that this limitation stems not from the design of the heuristic function, but from the inherent structural property of Hybrid A*, whose heuristic cannot precisely capture the actual motion cost of the vehicle, thus failing to guarantee global optimality. In addition, due to the discreteness of the control of hybrid A*, it requires multiple parameter adjustments to successfully plan the results in extremely narrow environments such as scenario 3, which affects the practicality of the algorithm.

In contrast, the proposed geodesic-based path planning method achieves a favorable balance among path length, smoothness, and feasibility. The method's priority-based design emphasizes obstacle avoidance over strict kinematic compliance, reflecting a sound engineering approach where safety takes precedence. In extremely narrow scenarios, the algorithm may generate a few points with marginally higher curvature to maintain adequate safety clearances. This intelligent prioritization ensures collision avoidance is never compromised, while minor kinematic violations can be easily addressed through post-processing techniques. This demonstrates that even in constrained environments,

Table 4
Comparison of path planning methods in narrow scenarios.

Scenarios	Methods	Path Length	Max curv.	Mean curv.	Control Cost	Mean Obs. Dist.	Min Obs. Dist.	Feasibility
Scenario 1	A*	<u>25.301</u>	1.1906	0.1113	—	<u>2.4468</u>	1.7928	89 %
	RRT*	<u>25.180</u>	1.3028	<u>0.0601</u>	—	2.3458	1.8240	93 %
	Hybrid A*	39.052	<u>0.2689</u>	0.1070	<u>1.3943</u>	2.2773	<u>1.8742</u>	100 %
	Ours	29.290	<u>0.3763</u>	0.0787	<u>1.1484</u>	<u>2.6027</u>	<u>2.2104</u>	100 %
Scenario 2	A*	<u>25.301</u>	1.1906	0.1113	—	2.5083	1.7928	88 %
	RRT*	<u>25.180</u>	1.3028	<u>0.0601</u>	—	2.3451	1.8240	93 %
	Hybrid A*	31.631	<u>0.3734</u>	0.1031	1.2334	<u>2.6062</u>	<u>1.9334</u>	100 %
	Ours	30.306	<u>0.3830</u>	0.0777	<u>1.0473</u>	<u>2.6217</u>	<u>2.3478</u>	100 %
Scenario 3	A*	<u>31.211</u>	1.5088	0.1625	3.8785	1.9779	1.7915	85 %
	RRT*	<u>31.322</u>	1.3164	0.0923	<u>2.7847</u>	<u>2.0928</u>	<u>1.8188</u>	90 %
	Hybrid A*	42.522	<u>0.3701</u>	0.1107	3.6371	2.0158	1.6645	100 %
	Ours	32.861	<u>0.3930</u>	0.0825	<u>2.3762</u>	<u>2.1112</u>	<u>1.9533</u>	99 %

Table 5
Planning time comparison of different methods (mean&std).

Scenarios	A*	RRT*	Hybrid A*	Ours
Spacious 1	<u>0.675 ± 0.010</u>	2.940 ± 0.744	<u>1.317 ± 0.031</u>	2.527 ± 0.023
Spacious 2	<u>0.764 ± 0.048</u>	5.400 ± 0.810	<u>3.354 ± 0.037</u>	3.483 ± 0.065
Spacious 3	<u>0.978 ± 0.008</u>	7.003 ± 0.355	4.254 ± 0.076	<u>3.863 ± 0.042</u>
Narrow 1	<u>0.180 ± 0.010</u>	2.090 ± 0.554	2.154 ± 0.016	2.915 ± 0.035
Narrow 2	<u>0.147 ± 0.005</u>	<u>1.957 ± 0.325</u>	2.288 ± 0.019	2.916 ± 0.038
Narrow 3	<u>0.163 ± 0.007</u>	3.544 ± 0.853	3.301 ± 0.022	3.729 ± 0.124

our method can guide robots to reach targets efficiently, safely, and reliably.

4.2.3. Real-time analysis

To evaluate the computational efficiency of the proposed path planning method, we conducted a statistical analysis of the planning time for different algorithms across scenarios (three scenarios in each of the two environments described above). **Table 5** presents the real-time performance of each algorithm, including the mean computation time and standard deviation over 10 independent runs for each scenario. It should be noted that RRT* is an anytime algorithm without a predefined termination condition, potentially leading to unbounded computation time. To ensure fair comparison, we implemented an early stopping criterion for RRT* when no path length improvement was achieved over 1000 consecutive iterations, providing a reasonable basis for computational time evaluation.

The computational efficiency analysis reveals distinct performance characteristics of different algorithms across various environmental scenarios. A* consistently maintains the lowest computation time due to its deterministic graph search characteristics and absence of kinematic constraints. However, it provides solutions with poor feasibility and smoothness, making them unsuitable for practical robotic systems. RRT* demonstrates the highest computational cost with significant variance across scenarios, aligning with its sampling-based stochastic nature and maintaining uncertainty even under defined termination criteria. Hybrid A* shows moderate planning time, reflecting the additional computational overhead required for kinematic feasibility. Our proposed method achieves a balance between computational efficiency and solution quality. Compared to Hybrid A*, our method demonstrates superior efficiency in complex scenarios, while compared to RRT*, it exhibits better temporal stability and predictability. Notably, in narrow scenarios, although RRT* shows shorter computation time, it generates paths with low feasibility, highlighting our method's advantages in ensuring solution quality.

Different algorithms exhibit distinct characteristics regarding computation time factors. A* computation time correlates almost solely with path length, while RRT* and Hybrid A* depend on both scenario complexity and path length. In contrast, our algorithm's computation time

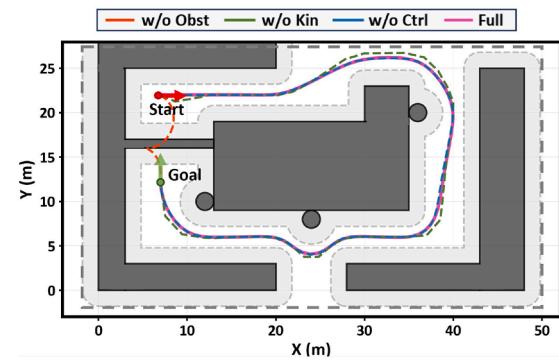


Fig. 8. Illustration of ablation experiment results for different model variants.

Table 6
Ablation study performance for model variants.

Methods	Length	Max curv.	Control Cost	Min Obs. Dist.	Feasibility
w/o Obst	12.689	—	—	—	—
w/o Kin	88.441	0.4346	9.4294	2.6879	98 %
w/o Ctrl	85.431	<u>0.3701</u>	<u>9.1825</u>	<u>2.7014</u>	100 %
Full	<u>85.405</u>	<u>0.3448</u>	<u>8.3172</u>	<u>2.6754</u>	100 %

relates primarily to scenario complexity, particularly narrow passages, with minimal impact from path length in open areas. This characteristic provides good scalability for large-scale environments. Furthermore, using superior initial solutions instead of A*-generated initial guesses could further improve solving speed, offering greater optimization potential for real-time applications.

4.2.4. Ablation experiments

To further evaluate the contribution of each module to the overall performance, we conducted an ablation study in the same environment as Scenario 3 from the spacious scene simulation experiments. Four variants were considered: the complete model (Full), removal of the obstacle potential field coefficient (w/o Obst), removal of the kinematic constraint metric (w/o Kin), and removal of the control cost metric (w/o Ctrl). The experiments were performed and several key performance metrics were recorded, with the outcomes illustrated in **Fig. 8** and the corresponding quantitative results summarized in **Table 6**. Notably, to ensure the positive definiteness of the metric tensor in the ablation settings, the removed sub-matrix was replaced with an identity matrix, while in the case of removing obstacle coefficients, the corresponding scaling factor was fixed to one.

The results of the ablation study clearly reveal the distinct contributions of each module to the overall performance. When the obstacle potential field coefficient is removed, the robot tends to ignore obstacles and directly approaches the goal, making indicators meaningless

in this setting. By contrast, the introduction of the obstacle module ensures consistent obstacle avoidance across all other variants, allowing safe operation. When the kinematic constraint is removed, the robot can still avoid obstacles and reach the target, but the resulting trajectories exhibit larger curvature and even occasional infeasible states, resembling A*-like behavior, which highlights the importance of this term in constraining path feasibility. In the absence of the control cost, the robot is still able to reach the goal safely and feasibly, yet the trajectory shows increased curvature and higher control effort, indicating that this module plays a role in suppressing unnecessary steering and improving smoothness. By integrating all components, the full model achieves a well-balanced performance, combining obstacle avoidance, feasibility, and trajectory smoothness, thereby demonstrating the complementary nature and necessity of each module.

4.3. Application deployment

To evaluate the real-world performance of the proposed algorithm, we deployed the system in a representative vehicle yard at Yantai Port and conducted on-site experiments. The feasibility of the generated paths was verified through path planning and tracking tests. Fig. 9 illustrates the operational environment and the monitoring system interface.

4.3.1. Experiment platform

The algorithm was deployed on a four-wheel independent steering and driving robot platform. The onboard control unit is equipped with an Intel Core i7-9700H CPU and a GeForce GTX 2060 GPU, running Ubuntu and ROS for efficient task execution.

As shown in Fig. 10, the robot is equipped with a range of onboard sensors including LiDAR, radar, and CCD cameras. High-precision global positioning is achieved through a real-time kinematic positioning module, ensuring both localization and planning accuracy.

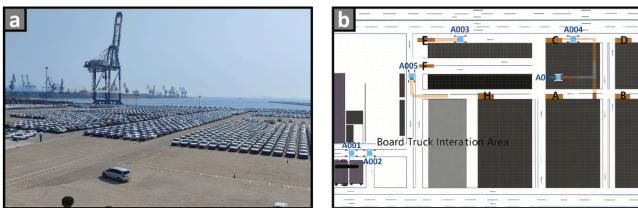


Fig. 9. Yard operation at Yantai port. (a) Physical experiment environment. (b) Remote monitoring interface for path visualization and robot status.



Fig. 10. Transfer robot hardware configuration and specifications.



Fig. 11. Real-world experiments in robot operation scenarios. (a) Vehicle pickup scenario: transfer robot moving from charging station to storage yard to retrieve commercial vehicles. (b) Vehicle drop-off scenario: transfer robot delivering commercial vehicles from storage yard to truck interaction area.

Table 7
Comparison of planning methods in real-world scenarios.

Scenarios	Methods	Length	Max curv.	Control Cost	Min Obs. Dist.	Feasibility
Vehicle Pickup	A*	153.63	0.5808	10.818	1.8185	97 %
	RRT*	150.12	0.3926	7.4745	1.8375	98 %
	Hybrid A*	179.16	0.3701	8.6380	2.5600	100 %
	Ours	157.19	0.2839	4.9410	3.5085	100 %
Vehicle Drop-off	A*	103.41	0.7447	7.8657	1.7133	95 %
	RRT*	101.84	0.6704	6.8460	1.8236	97 %
	Hybrid A*	111.91	0.3161	6.7822	2.5224	100 %
	Ours	106.60	0.2848	3.3258	3.8372	100 %

4.3.2. Field planning evaluation

In the actual operation scenario of the Yantai Port vehicle yard, the algorithm was validated for two typical tasks: vehicle pickup and drop-off. As shown in Fig. 11, the spatial correspondence between the planned paths and the real-world environment is clearly demonstrated. The robot first arrives at the outer roadway of the yard and enters the operation area by performing in-place rotation enabled by its 4WS system. After completing the pickup task, it exits the yard, performs a second rotational maneuver, and proceeds to the interaction zone to complete the vehicle drop-off.

During practical deployment, we observed that paths generated by A* and RRT* often suffer from poor executability, while Hybrid A* tends to produce redundant trajectories. The key performance metrics of each path planning algorithm in this scenario are summarized in Table 7. Overall, the proposed method consistently demonstrates superior performance in terms of executability and safety margin, effectively meeting the demands of real-world operations.

4.3.3. Tracking validation

To further validate the executability of the generated paths, a pure pursuit control algorithm was employed to guide the robot in performing path tracking tasks. The look-ahead distance was dynamically adjusted within a range of 1m to 2.5m based on the local path curvature, and the reference speed was set to 4m/s.

Fig. 12 presents the actual tracking performance in both vehicle pickup and drop-off scenarios. Quantitative evaluation of key tracking metrics shows that the maximum lateral deviation was 0.0752m in the pickup phase and 0.0809m in the drop-off phase, while the average deviation for both paths remained consistently at 0.01m. These tracking results indicate that the paths generated by the proposed algorithm exhibit stable curvature at control points and smooth control inputs, effectively meeting the high-precision operational requirements of robots in port environments.

Through real-world deployment and field testing, we validate the engineering practicality of the proposed method. By generating paths with enhanced safety margins and smoother curvature profiles, the approach effectively improves the reliability of autonomous execution for robots operating in complex yard environments, directly addressing critical operational challenges faced by existing automated vehicle systems in port facilities.

Unlike conventional methods that often produce theoretically optimal but practically difficult-to-execute trajectories, our approach prioritizes kinematic feasibility while maintaining planning efficiency, achieving more robust autonomous operations with reduced human intervention requirements. The successful field validation demonstrates the technical maturity for engineering applications, establishing a foundation for deployment in automated parking systems, intelligent warehousing, and other logistics facilities where precise maneuvering control is critically required.

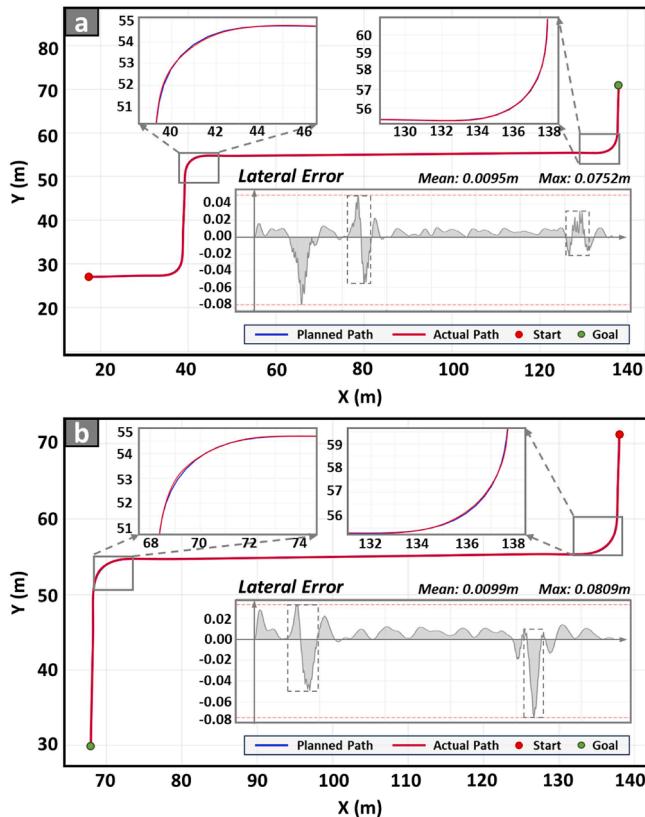


Fig. 12. Real-world robot path visualization and tracking error analysis. (a) Vehicle pickup scenario. (b) Vehicle drop-off scenario.

5. Conclusion

This paper proposes a geodesic-based path planning method on Riemannian manifolds, tailored for the scheduling of robots in structured port environments. The approach integrates robot heading, steering effort, and obstacle accessibility into the construction of a Riemannian metric tensor that reflects the local path cost structure. By formulating the planning task as a geodesic shortest path problem and solving it efficiently via GHF, the resulting trajectories naturally adhere to kinematic constraints while maintaining strong obstacle avoidance capability. Simulation and real-world experiments conducted in representative port yard scenarios demonstrate the practicality and robustness of the proposed method under complex spatial constraints and obstacle configurations. Compared to baseline approaches, our method achieves near-100 % path feasibility, while offering competitive path lengths and lower average curvatures. More importantly, it improves the minimum obstacle distance by over 25 % relative to other algorithms. Furthermore, the average planning time across all scenarios is less than 4 seconds, demonstrating a strong balance between computational efficiency and solution quality.

Future work will focus on extending the proposed method in several directions. First, to improve adaptability to diverse obstacle shapes, constructing obstacle potential fields based on convex hull representations may help reduce the loss of feasible space. Second, incorporating more detailed kinematic models together with uncertainty modeling of the robot's execution is expected to enhance the fidelity of trajectory feasibility and improve robustness during control execution. Finally, as the current approach is mainly designed for long-range global planning, extending the framework to local obstacle avoidance remains an important direction. In particular, constructing local potential fields and employing forward iterations such as the Runge–Kutta method may further improve computational efficiency and enable real-time applicability to local planning scenarios.

CRediT authorship contribution statement

Runjiao Bao: Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Software; **Junzheng Wang:** Writing – review & editing, Project administration, Funding acquisition; **Shoukun Wang:** Supervision, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62473044.

References

- Bao, R., Xu, Y., Xue, J., Yuan, H., Zhang, L., & Wang, S. (2025). A hierarchical target vehicle pose detection framework in ro-ro terminal environment. *IEEE Sensors Journal*, 25(14), 27001–27012. <https://doi.org/10.1109/JSEN.2025.3574091>
- Bonalli, R., Cauligi, A., Bylard, A., & Pavone, M. (2019). GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming. In *2019 International conference on robotics and automation (ICRA)* (pp. 6741–6747). <https://doi.org/10.1109/ICRA.2019.8794205>
- Chai, R., Chen, K., Hua, B., Lu, Y., Xia, Y., Sun, X.-M., Liu, G.-P., & Liang, W. (2024). A two phases multiobjective trajectory optimization scheme for multi-UGVs in the sight of the first aid scenario. *IEEE Transactions on Cybernetics*, 54(9), 5078–5091.

- Chai, R., Liu, D., Liu, T., Tsourdos, A., Xia, Y., & Chai, S. (2023). Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver. *IEEE Transactions on Automation Science and Engineering*, 20(3), 1633–1647. <https://doi.org/10.1109/TASE.2022.3183610>
- Chai, R., Tsourdos, A., Savvaris, A., Chai, S., & Xia, Y. (2019). Two-stage trajectory optimization for autonomous ground vehicles parking maneuver. *IEEE Transactions on Industrial Informatics*, 15(7), 3899–3909. <https://doi.org/10.1109/TII.2018.2883545>
- Colding, T., Minicozzi, W., Pedersen, E. (2015). Mean curvature flow. *Bulletin of the American Mathematical Society*, 52(2), 297–333.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2008). Practical search techniques in path planning for autonomous driving. AAAI Workshop - Technical Report.
- Govindarajan, V., Bhattacharya, S., & Kumar, V. (2016). Human-robot collaborative topological exploration for search and rescue applications. In *Distributed autonomous robotic systems: The 12th international symposium* (pp. 17–32). Springer.
- Guo, S., Gong, J., Shen, H., Yuan, L., Wei, W., & Long, Y. (2025). DBVSB-P-RRT*: A path planning algorithm for mobile robot with high environmental adaptability and ultra-high speed planning. *Expert Systems with Applications*, 266, 126123. <https://doi.org/10.1016/j.eswa.2024.126123>
- Guyer, J. E., Wheeler, D., & Warren, J. A. (2009). FiPy: Partial differential equations with python. *Computing in Science & Engineering*, 11(3), 6–15. <https://doi.org/10.1109/MCSE.2009.52>
- Han, R., Wang, S., Wang, S., Zhang, Z., Chen, J., Lin, S., Li, C., Xu, C., Eldar, Y. C., Hao, Q., & Pan, J. (2025). NeuPAN: Direct point robot navigation with end-to-end model-based learning. *IEEE Transactions on Robotics*, 41, 2804–2824. <https://doi.org/10.1109/TRO.2025.3554252>
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107. <https://doi.org/10.1109/TSSC.1968.300136>
- Huang, S. A., Wu, K.-L., & Liu, J.-S. (2015). UAV robust navigation via geodesic path planning and nonlinear mixed h₂/h_∞ control strategy. In *2015 IEEE 12th international conference on networking, sensing and control* (pp. 309–314). IEEE.
- Huo, F., Zhu, S., Dong, H., & Ren, W. (2024). A new approach to smooth path planning of Ackerman mobile robot based on improved ACO algorithm and b-spline curve. *Robotics and Autonomous Systems*, 175, 104655.
- Hönig, W., Ortiz-Haro, J., & Toussaint, M. (2022). DB-A*: Discontinuity-bounded search for kinodynamic mobile robot motion planning. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 13540–13547). <https://doi.org/10.1109/IROS47612.2022.9981577>
- Jeong, I.-B., Lee, S.-J., & Kim, J.-H. (2019). Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. *Expert Systems with Applications*, 123, 82–90. <https://doi.org/10.1016/j.eswa.2019.01.032>
- Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., & Schaal, S. (2011). Stomp: Stochastic trajectory optimization for motion planning. In *2011 IEEE international conference on robotics and automation* (pp. 4569–4574). <https://doi.org/10.1109/ICRA.2011.5980280>
- Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.
- Kasmaiee, S., Tadjfar, M., & Kasmaiee, S. (2024). Optimization of blowing jet performance on wind turbine airfoil under dynamic stall conditions using active machine learning and computational intelligence. *Arabian Journal for Science and Engineering*, 49(2), 1771–1795.
- Kasmaiee, S., Tadjfar, M., & Kasmaiee, S. (2025a). Pressure swirl nozzles with different discharge orifice shapes injecting into transverse airflow. *Physics of Fluids*, 37(1), 013357.
- Kasmaiee, S., Tadjfar, M., Kasmaiee, S., & Ahmadi, G. (2025b). Backward injection of liquid jets in crossflow. *Physics of Fluids*, 37(4), 043310.
- Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4), 566–580. <https://doi.org/10.1109/70.508439>
- Kim, S., Bhattacharya, S., Ghrist, R., & Kumar, V. (2013). Topological exploration of unknown and partially known environments. In *2013 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3851–3858). IEEE.
- Klein, H., Jaquier, N., Meixner, A., & Asfour, T. (2023). On the design of region-avoiding metrics for collision-safe motion generation on riemannian manifolds. In *2023 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2346–2353). IEEE.
- Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. millennium conference. IEEE international conference on robotics and automation. symposia proceedings (cat. no.00CH37065)* (pp. 995–1001 vol.2). (vol. 2). <https://doi.org/10.1109/ROBOT.2000.844730>
- Laux, M., & Zell, A. (2021). Robot arm motion planning based on geodesics. In *2021 IEEE international conference on robotics and automation (ICRA)* (pp. 7585–7591). IEEE.
- Laux, M., & Zell, A. (2023). Boundary conditions in geodesic motion planning for manipulators. In *2023 IEEE international conference on robotics and automation (ICRA)* (pp. 1558–1564). <https://doi.org/10.1109/ICRA48891.2023.10160843>
- LaValle, S. (1998). Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*.
- Li, Y., Li, G., & Wang, X. (2024). Research on trajectory planning of autonomous vehicles in constrained spaces. *Sensors*, 24(17), 5746.
- Liu, Z., Chen, J., Xia, H., & Lan, F. (2022). Human-like trajectory planning for autonomous vehicles based on spatiotemporal geometric transformation. *IEEE Transactions on Intelligent Transportation Systems*, 23(11), 20160–20176.
- Ratliff, N., Zucker, M., Bagnell, J. A., & Srinivasa, S. (2009). Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE international conference on robotics and automation* (pp. 489–494). <https://doi.org/10.1109/ROBOT.2009.5152817>
- Sahin, A., & Bhattacharya, S. (2024). Topo-geometrically distinct path computation using neighborhood-augmented graph, and its application to path planning for a tethered robot in 3D. *IEEE Transactions on Robotics*, 41, 20–41.
- Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., & Abbeel, P. (2014). Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9), 1251–1270. <https://doi.org/10.1177/0278364914528132>
- Wang, B., Ju, D., Xu, F., & Feng, C. (2023). Bi-RRT*: An improved bidirectional RRT* path planner for robot in two-dimensional space. *IEEE Transactions on Electrical and Electronic Engineering*, 18(10), 1639–1652.
- Webb, D. J., & van den Berg, J. (2013). Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In *2013 IEEE international conference on robotics and automation* (pp. 5054–5061). <https://doi.org/10.1109/ICRA.2013.6631299>
- Xie, Z., & Dames, P. (2023). Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles. *IEEE Transactions on Robotics*, 39(4), 2700–2719. <https://doi.org/10.1109/TRO.2023.3257549>
- Xu, Y., Bao, R., Zhang, L., Wang, J., & Shoukun, W. (2025). Embodied intelligence in RO/RO logistic terminal autonomous intelligent transportation robot architecture. *Science China Information Sciences*, 68, 150210.
- Yang, H., Yao, C., Liu, C., & Chen, Q. (2023). RMRL: Robot navigation in crowd environments with risk map-based deep reinforcement learning. *IEEE Robotics and Automation Letters*, 8(12), 7930–7937. <https://doi.org/10.1109/LRA.2023.3322093>
- Yusheng, Z., Xiangrong, W., & Zaihua, W. (2020). On the nonholonomic constraints and motion control of wheeled mobile structures1. *Chinese Journal of Theoretical and Applied Mechanics*, 52(4), 1143–1156. <https://doi.org/10.6052/0459-1879-19-257>
- Zhang, L., Xu, Y., Si, J., Bao, R., An, Y., Wang, S., & Wang, J. (2025a). Autonomous transfer robot system for commercial vehicles at ro-ro terminals. *Expert Systems with Applications*, 289, 128347. <https://doi.org/10.1016/j.eswa.2025.128347>
- Zhang, Y., Zhou, Q., & Yang, X.-S. (2025b). An RRT* algorithm based on riemannian metric model for optimal path planning. *arXiv preprint arXiv:2507.0169*.