GMPPO: Guided Model Predictive Path Optimization for Mobile Robots in Obstacle-Constrained Environments

Lin Zhang, Wenhao Zhang, Runjiao Bao, Tianwei Niu, Shoukun Wang, and Junzheng Wang

***Abstract*—Autonomous navigation of industrial mobile robots in obstacle-constrained environment, such as manufacturing floors, warehouse storage systems, and factory inspection scenarios, poses significant challenges in ensuring path feasibility, safety, and real-time performance. This article proposes a guided model predictive path optimization (GMPPO) framework to address these issues. First, an enhanced convex feasible set algorithm is employed to generate robust initial paths by integrating decoupled dimensionality reduction and a dual-layer fault-tolerant mechanism. Second, based on this reference path, robot geometry, and real-time obstacle information, dynamic safety corridors are constructed to characterize local feasible spaces. Finally, these reference paths and corresponding safety corridors guide a high-fidelity model predictive optimization process, yielding smooth, dynamically feasible, and collision-free trajectories under nonholonomic constraints. Theoretical analysis establishes the resolution completeness and feasibility guarantee of the proposed method. Extensive simulations, real-world robotic experiments, and engineering deployments demonstrate that GMPPO significantly improves planning success rates, trajectory smoothness, and operational safety, while maintaining high computational efficiency and broad applicability.**

***Index Terms*—Enhanced convex feasible set, mobile robot navigation, model predictive optimization, obstacle-constrained environments, safe corridor.**

I. Introduction

THE advancement of Industry 4.0 has underscored the pivotal role of mobile robots in flexible manufacturing, warehouse logistics, and operations in hazardous environments [1], [2]. However, generating feasible paths in cluttered industrial settings while ensuring collision avoidance, dynamic feasibility, and real-time performance remains a critical challenge. In this study, we focus on obstacle constrained environments, which refer to industrial spaces where densely arranged equipment and structural elements limit the available free space and form narrow, cluttered, and difficult to navigate corridors for mobile robots. These characteristics exacerbate the difficulty of achieving safe and smooth navigation, especially for nonholonomic robots, i.e., platforms whose motions are limited by nonintegrable velocity constraints and cannot move instantaneously in arbitrary directions, operating in real-world industrial layouts.

Although sampling-based planners perform well in open spaces, their solution quality significantly deterioratesin obstacle-constrained environments, leading to high failure rates. Optimization-based methods can produce smooth trajectories but often become trapped in local minima, especially in high-curvature scenarios [3], [4]. Hybrid approaches, typically adopting a "global search + local refinement" framework, offer partial improvements but still face three key limitations. First, they lack structural awareness of obstacle-constrained spaces, often treating them as generic obstacles while overlooking their limited connectivity and high collision risk. Second, kinematic and dynamic constraints, such as minimum turning radius and acceleration limits, are frequently imposed post hoc, compromising the feasibility of the resulting trajectory. Third, the decoupling between global and local modules fragments the solution space and reduces computational efficiency. In complex industrial environments with obstacle-constrained spaces, there is a pressing need for a planning framework that simultaneously ensures dynamically feasible, collision-free, and smooth trajectories while maintaining computational efficiency.

A. Related Work

Path planning is a fundamental task in autonomous navigation, aiming to generate feasible trajectories from start to goal while ensuring safety and dynamic feasibility [5], [6]. Existing path planning methods fall into four main categories [7]: graph search, sampling-based, optimization-based, and learning-based approaches. These paradigms have been widely studied, including explorations in obstacle-constrained environments.

Received 25 June 2025; revised 17 December 2025; accepted 9 March 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62473044, and in part by BIT Research and Innovation Promoting Project under Grant 2024YCXZ007. *(Corresponding authors: Tianwei Niu; Shoukun Wang.)*

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: bit.zhanglin@bit.edu.cn; 3120220901@bit.edu.cn; 3120230765@bit.edu.cn; ntwbit@bit.edu.cn; bitwsk@bit.edu.cn; wangjz@bit.edu.cn).

Digital Object Identifier 10.1109/TIE.2026.3675199

Dijkstra and A* are representative graph search algorithms, commonly applied when the configuration space is discretized into grids or graphs [8]. While effective in static environments, these methods generate discrete paths that lack continuity, leading to tracking oscillations or degraded control stability. Several studies have proposed enhancements [9], and Zhang et al. [10] proposed a hybrid A*-DWA method for unmanned vessels. However, these methods still exhibit limited adaptability for real-world navigation in obstacle-constrained environments.

Rapidly-exploring random tree (RRT) and probabilistic roadmap (PRM) are classical sampling-based planning algorithms that construct connectivity graphs in high-dimensional configuration spaces through random sampling [11]. OpenPlanner [12], a notable variant, integrates sampling strategies, constraint handling, and cost optimization, and has demonstrated strong performance in autonomous driving applications. However, its adaptability to obstacle-constrained and topologically complex environments remains limited. To improve applicability in obstacle-constrained environments, several sampling-based methods have been proposed by researchers [13]. For instance, Zhong et al. [14] introduced GVDF-RRT*, which employs generalized Voronoi diagrams to guide sparse sampling and direct node connections, improving connectivity and efficiency in constrained spaces. Guo et al. [15] proposed a grid-based bridge detection method to construct free-space graphs for accurate path planning in narrow passages. Zhang et al. [16] developed RA-RRTV*, which combines local "vine-like" expansion with chance-constrained optimization to balance path quality and risk under localization uncertainty. Despite these advances, most methods approximate the robot as a holonomic point mass, neglecting nonholonomic kinematic constraints. As a result, trajectory feasibility and quality often degrade in real-world deployments.

Optimization-based algorithms are widely used to generate smooth trajectories that satisfy kinematic and dynamic constraints [17]. The timed elastic band (TEB) method, a widely used method, optimizes a time-parameterized elastic band to generate smooth, dynamically feasible, and collision-avoiding trajectories [18]. Huo et al. [19]. integrated steering angle constraints into an ant colony optimization framework to enhance convergence and feasibility. Chai et al. [20] proposed a decoupled fuzzy multiobjective strategy combining a collision-free 3-D tunnel planner with fuzzy inference to balance competing goals. Furthermore, Li et al. [21] developed a hybrid A* algorithm with adaptive expansion for autonomous vehicles in constrained spaces, improving trajectory efficiency and comfort. Kulathunga et al. [22] proposed a perception-driven navigation framework for robots in narrow spaces, achieving improved accuracy and control stability in experiments. Zhang et al. [23] proposed a multisegment path planning method for autonomous vehicles to enable parking in narrow environments. Despite these advances, dense obstacles and limited feasible regions in narrow environments still pose challenges, causing local minima, high computational costs, and trade-offs between smoothness and safety. Further improvements in efficiency and robustness are essential for navigation in complex environments.

Learning-based methods have rapidly advanced by leveraging deep neural networks to parameterize policies or value functions. These data-driven approaches [24], [25] optimize cumulative rewards through trial-and-error, enabling end-to-end or hierarchical trajectory generation from raw sensory inputs. To address navigation in confined and complex spaces, Denesh et al. [26] proposed a perceptron-Q learning fusion model for robot navigation in such environments, which enables dynamic obstacle detection and decision-making. Nakamura et al. [27] proposed a deep Q-network planner with a reward function incorporating velocity smoothing and steering constraints, achieving robust navigation in narrow corridors across varied scenes. Li et al. [28] integrated deep learning with Monte Carlo tree search in a hierarchical framework to jointly optimize path search and velocity profiles, enhancing efficiency, success rates, and ride comfort in complex environments. Despite their potential, learning-based methods face challenges such as slow convergence, high computational demands, and limited interpretability. Industrial narrow-space applications further suffer from scarce real-world data and a reality gap between simulation and deployment, limiting model generalization and safety.

B. Motivation and Contribution

To address these fundamental challenges, we propose a guided model predictive path optimization (GMPPO) framework based on a unified kinodynamic and collision-avoidance model. This framework systematically integrates critical navigation constraints for nonholonomic systems operating in obstacle-constrained spaces, ensuring safety, dynamic feasibility, and computational efficiency, while providing a universal solution applicable to diverse nonholonomic systems. The main contributions are summarized as follows.

1) We propose an enhanced convex feasible set (ECFS) algorithm with decoupled dimensionality reduction and a dual-layer fault tolerance mechanism. Designed for narrow environments, it hierarchically optimizes geometric constraints between the robot and obstacles, improving planning robustness and adaptability while reducing collision risk.
2) We develop a dynamic safety corridor-guided model predictive optimization algorithm. Leveraging a dual-disc robot model and real-time obstacle data, it generates dynamic safety corridors within a rolling-horizon scheme to jointly satisfy kinematic constraints, obstacle avoidance, and trajectory smoothness.
3) We provide theoretical guarantees for the algorithm's feasibility and validate it through simulations, field tests with physical robots, and industrial deployments. Results confirm its feasibility, robustness, and strong applicability in obstacle-constrained spaces.

The article is organized as follows. Section II covers preliminaries. Section III presents the method. Sections IV and V show simulation and experimental results. Section VI concludes and discusses future work.

II. Preliminaries

A. Problem Statement

Consider a 2-D Euclidean workspace $\mathcal{W} \subseteq \mathbb{R}^2$, where the obstacle region $\mathcal{O}_\mathcal{W} = \mathcal{O}^p_\mathcal{W} \cup \mathcal{O}^o_\mathcal{W}$ consists of prior-known and newly perceived obstacles, with dynamic obstacles not considered in this work. The free workspace is given by $\mathcal{F}_\mathcal{W} = \mathcal{W} \setminus \mathcal{O}_\mathcal{W}$. The robot pose is denoted as $\mathbf{s} = [x, y, \theta]^\top$, belonging to the configuration space $\mathcal{C} = \mathbb{SE}(2)$. The region occupied by the robot at pose $\mathbf{s}$ is denoted as $\mathcal{R}(\mathbf{s}) \subset \mathcal{W}$. Accordingly, the free configuration space is defined as $\mathcal{C}_{\text{free}} = \{\mathbf{s} \in \mathcal{C} \mid \mathcal{R}(\mathbf{s}) \subseteq \mathcal{F}_\mathcal{W}\}$, representing all collision-free configurations.

The objective of this study is to generate a feasible path for mobile robots navigating in obstacle-constrained and complex environments. Given a start configuration $\mathbf{s}_{\text{start}} \in \mathcal{C}_{\text{free}}$ and a goal configuration $\mathbf{s}_{\text{goal}} \in \mathcal{C}_{\text{free}}$, the planner aims to compute a discrete path $\mathcal{P} = \{\mathbf{s}_k\}_{k=0}^{N_p}$, where $\mathbf{s}_0 = \mathbf{s}_{\text{start}}$ and $\mathbf{s}_{N_p} = \mathbf{s}_{\text{goal}}$, subject to the following constraints

1) *Kinematic Feasibility*: Each transition $\mathbf{s}_k \to \mathbf{s}_{k+1}$ must satisfy the robot's kinematic constraints.
2) *Collision Avoidance*: All configurations must lie within the free configuration space, i.e., $\mathbf{s}_k \in \mathcal{C}_{\text{free}}, \quad \forall k \in \{0, \ldots, N_p\}$.

To improve path quality, curvature minimization is incorporated as a smoothing objective rather than a strict constraint, ensuring geometric continuity while still allowing tight turns in constrained regions.

B. Kinematic Model

To ensure kinematic feasibility, the robot's kinematic model must be explicitly incorporated as a constraint within the path optimization formulation. This study considers a four-wheel independently driven/steered (4WIDS) mobile platform, which offers a reduced turning radius and high maneuverability, making it well-suited for navigation in narrow and complex industrial environments [29]. The robot state and control vectors are defined as $\mathbf{x} = [x, y, v, \theta, \delta]^\top$ and $\mathbf{u} = [a, \omega]^\top$, respectively. The corresponding kinematic model is given by

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{B}\mathbf{u} \tag{1}$$

$$f(\mathbf{x}) = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ 0 \\ v\frac{2\tan(\delta)}{L_\text{b}} \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0,0 \\ 0,0 \\ 1,0 \\ 0,0 \\ 0,1 \end{bmatrix} \tag{2}$$

where (x, y) denote the coordinates of the robot's center in the global frame, v is the linear velocity, θ is the heading angle, δ is the steering angle, and L_b is the wheelbase. The control inputs are $[a, \omega]$, where a is the linear acceleration and ω is the steering angular velocity.

The control constraints are defined as follows:

$$a \in [a^{\min}, a^{\max}], \quad \omega \in [\omega^{\min}, \omega^{\max}] \tag{3}$$

where $a^{\min}$ and $a^{\max}$ correspond respectively to the minimum and maximum feasible acceleration of the robot, $\omega^{\min}$ and $\omega^{\max}$ represent the minimum and maximum allowable, velocities.

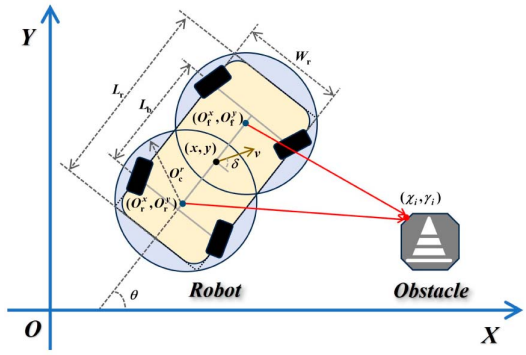


Fig. 1. Collision model between robots and obstacles.

C. Collision Model

To mitigate overfitting in constrained environments while balancing computational complexity and resource utilization, the robot's geometry is approximated using a dual-disc model. This simplification facilitates efficient collision checking during motion. As illustrated in Fig. 1, the centers and radii of the two circles are defined as follows:

$$\begin{aligned} O^x_\text{f} &= x + \frac{L_\text{r}}{4}\cos\theta, \quad O^y_\text{f} = y + \frac{L_\text{r}}{4}\sin\theta \\ O^x_\text{r} &= x - \frac{L_\text{r}}{4}\cos\theta, \quad O^y_\text{r} = y - \frac{L_\text{r}}{4}\sin\theta \end{aligned} \tag{4}$$

$$O^r_\text{c} = \sqrt{\left(\frac{L_\text{r}}{4}\right)^2 + \left(\frac{W_\text{r}}{2}\right)^2} \tag{5}$$

where $(O^x_\text{f}, O^y_\text{f})$ and $(O^x_\text{r}, O^y_\text{r})$ denote the centers of the front and rear circles, respectively, while O^r_c represents their radius. L_r and W_r correspond to the robot's length and width, respectively.

Consequently, the robot–obstacle collision detection is simplified to bipartite circle–obstacle checks, formulated as the following constraint:

$$\begin{aligned} &(O^x_\text{f} - \chi_i)^2 + (O^y_\text{f} - \gamma_i)^2 \geq O^{r\,2}_\text{c}, \quad i \in \{1, \ldots, N\} \\ &(O^x_\text{r} - \chi_i)^2 + (O^y_\text{r} - \gamma_i)^2 \geq O^{r\,2}_\text{c}, \quad i \in \{1, \ldots, N\} \end{aligned} \tag{6}$$

where (χ_i, γ_i) denotes a discrete point on the boundary of an environmental obstacle, and N is the total number of sampled boundary points.

D. Obstacle Detection

In unstructured industrial environments, mobile robot navigation is challenged by highly nonconvex obstacles and the trade-off between computational complexity and real-time performance. To mitigate these issues, nonconvex problems are commonly transformed into convex formulations. In this study, range filtering is first applied to remove excessively high or distant points. Then, point cloud segmentation and feature extraction are performed using the open-source LeGO-LOAM algorithm [30], effectively eliminating ground points and clustering nonground points. Finally, the clustered 3-D point clouds are projected online onto a 2-D grid map and approximated as

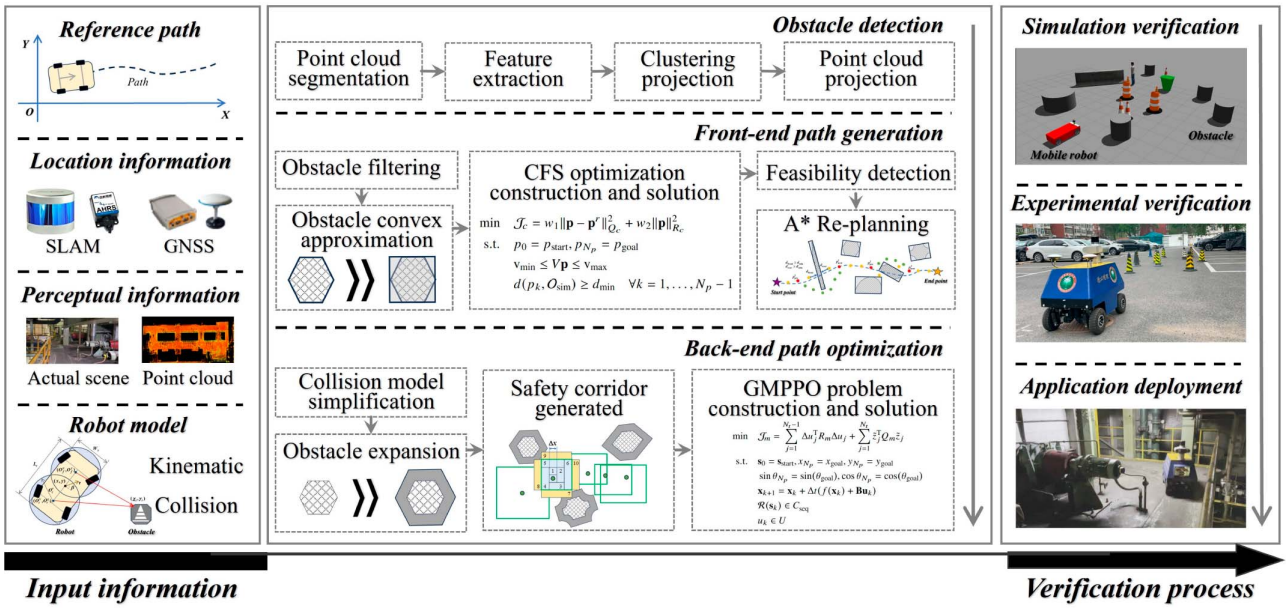


Fig. 2. Overall architecture diagram of the GMPPO algorithm.

convex polygons, thereby completing the obstacle detection process and facilitating subsequent planning. Parameters for point cloud segmentation and feature extraction are chosen based on the robot's dimensions, and the size of the projected 2-D grid map is set according to the workspace scale, smaller for narrow spaces and larger for expansive environments.

III. Method

This study proposes GMPPO, a unified framework that integrates front-end path generation with back-end optimization to enable efficient, feasible, and collision-free navigation in obstacle-constrained environments. The framework is shown in Fig. 2.

A. Front-End Path Generation

In obstacle-constrained, complex environments, reference paths are prone to collisions. The classical convex feasible set (CFS) algorithm models motion constraints and obstacles as convex sets, using projections to generate feasible paths with mathematical rigor, global optimality, and multiobjective support [31]. However, it suffers from high computational costs in high-dimensional spaces, and convex approximations of nonconvex obstacles can shrink feasible regions, leading to path omissions and reduced real-time performance and robustness.

To improve CFS algorithm practicality in complex environments, this article proposes a path generation method based on an ECFS algorithm integrating decoupled dimensionality reduction and a two-level fault-tolerant repair mechanism. By decoupling geometric modeling of the robot and obstacles stepwise, computational complexity is greatly reduced. When local optimization fails, a A* search restores path connectivity. This method combines the strict constraints of convex optimization with the fast response of heuristic search, ensuring path safety while enhancing robustness and adaptability. The pseudocode is shown in Algorithm 1.

Algorithm 1: The ECFS Algorithm.

Require: Reference path $\mathcal{P}_{\text{ref}}$, obstacle set $\mathcal{O}_{\mathcal{W}}$, maximum boundary $L_{\max}$;
Ensure: Front-end path $\mathcal{P}_{\text{f}}$;
$\mathcal{O}_{\text{sim}} \leftarrow \textbf{\textit{SimplifyObstacles}}(\mathcal{O}_{\mathcal{W}}, \mathcal{P}_{\text{ref}}, L_{\max})$; {Remove distant obstacles; approximate convex polygons >4 vertices with bounding rectangles.}
$\mathcal{P}_{\text{cfs}} \leftarrow \textbf{\textit{SolveCFSPath}}(\mathcal{P}_{\text{ref}}, \mathcal{O}_{\text{sim}})$; {Point-mass model optimization.}
$\mathcal{P}_{\text{coll}} \leftarrow \textbf{\textit{DetectCollisionSegments}}(\mathcal{P}_{\text{cfs}}, \mathcal{O}_{\text{sim}})$; {Dense interpolation collision check.}
if $\mathcal{P}_{\text{coll}} = \emptyset$ **then**
  $\mathcal{P}_{\text{f}} \leftarrow \mathcal{P}_{\text{cfs}}$;
else
  for each collision segment $\mathcal{P}_i \in \mathcal{P}_{\text{coll}}$ **do**
    $p_{\text{pre}} \leftarrow \textbf{\textit{PreviousWaypoint}}(\mathcal{P}_i)$;
    $p_{\text{next}} \leftarrow \textbf{\textit{NextWaypoint}}(\mathcal{P}_i)$;
    $\mathcal{P}_{\text{astar}} \leftarrow \textbf{\textit{AStarSearch}}(p_{\text{pre}}, p_{\text{next}})$;
    $\mathcal{P}_{\text{ecfs}} \leftarrow \textbf{\textit{ReplaceSegment}}(\mathcal{P}_{\text{cfs}}, \mathcal{P}_i, \mathcal{P}_{\text{astar}})$;
  end for
  $\mathcal{P}_{\text{f}} \leftarrow \mathcal{P}_{\text{ecfs}}$;
end if
return $\mathcal{P}_{\text{f}}$

To improve computational efficiency and reduce optimization dimensionality, two decoupled strategies are applied during the front-end path generation. First, the robot is approximated as a point mass, simplifying kinematic constraints. Second, the environment is processed via the

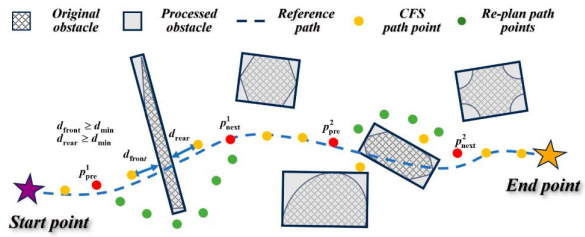


Fig. 3. Fault-tolerant repair mechanism of the ECFS.

obstacle simplification function, ***SimplifyObstacles*** (Algorithm 1, Line 1). This function filters out obstacles beyond a distance threshold $L_{\max}$, which is set according to the size of the local 2-D grid map after perception processing, to focus on local constraints, and replaces complex polygons (with more than four vertices) with their minimum bounding rectangles, thereby reducing constraint complexity. Based on the simplified obstacle set $\mathcal{O}_{\text{sim}}$ and reference path $\mathcal{P}_{\text{ref}}$, a CFS optimization problem is then formulated as follows:

$$\begin{aligned} \min \quad & \mathcal{J}_c = w_1 \|\mathbf{p} - \mathbf{p}^r\|_{Q_c}^2 + w_2 \|\mathbf{p}\|_{R_c}^2 \\ \text{s.t.} \quad & p_0 = p_{\text{start}}, p_{N_p} = p_{\text{goal}} \\ & \mathrm{v}_{\min} \le V\mathbf{p} \le \mathrm{v}_{\max} \\ & d(p_k, \mathcal{O}_{\text{sim}}) \ge d_{\min}, \quad \forall k = 1, \ldots, N_p - 1, \end{aligned} \tag{7}$$

where $\mathcal{J}c$ includes path deviation and length terms weighted by w_1 and w_2, with Q_c and R_c being positive definite matrices. The optimized path is denoted by $\mathbf{p} = \{p_k\}_{k=0}^{N_p}$, where $p_k = [x_k, y_k]^\top$ is the kth path point, and $\mathbf{p}^r$ represents the reference path. The matrix V is a first-order forward difference operator applied to $\mathbf{p}$ to penalize excessive path variation. The vectors $\mathbf{v}_{\min}, \mathbf{v}_{\max} \in \mathbb{R}^{N_p}$ specify the lower and upper bounds of the decision variables. The function $d(p_k, \mathcal{O}_{\text{sim}})$ denotes the minimum Euclidean distance between p_k and the obstacle set $\mathcal{O}_{\text{sim}}$. In practice, w_1 and w_2 balance tracking accuracy and smoothness, Q_c and R_c are scaled according to variable magnitudes, and the minimum safety distance $d_{\min}$ is determined by the robot geometry and safety margin in the local 2-D grid map.

To address solution failures and path oscillations in CFS optimization (see Fig. 3), we propose a two-layer fault tolerance and repair mechanism. First, we use the ***DetectCollisionSegments*** function (Algorithm 1, Line 3) to perform dense interpolation and collision checks on the initial path $\mathcal{P}_{\text{cfs}}$, identifying a set of potential conflicting segments ($\mathcal{P}_{\text{coll}}$). Then, for each element $\mathcal{P}_i \in \mathcal{P}_{\text{coll}}$, a local A* search is performed on the projected 2-D grid map using its endpoints (see Algorithm 1, lines 7–13), generating a collision-free alternative path $\mathcal{P}_{\text{astar}}$. These are iteratively merged into an updated path $\mathcal{P}_{\text{ecfs}}$, ensuring topological connectivity and geometric safety, and enhancing robustness in obstacle-constrained environments.

B. Back-End Path Optimization

In obstacle-constrained spaces, complex constraints, cluttered obstacles, and high-precision tracking demands challenge safe path planning. Traditional methods often struggle with multidimensional constraints and real-time adaptability, leading to collision-prone or infeasible trajectories. Model predictive optimization offers an effective solution by enabling real-time handling of dynamic constraints, integration of multiphysics constraints, and holistic trajectory optimization.

Algorithm 2: Safe Corridor Based Model Predictive Path Optimization.

Require: Front-end reference path $\mathcal{P}_{\text{f}}$, robot geometry $\mathcal{G}_{\text{rob}}$, obstacle set $\mathcal{O}_{\mathcal{W}}$, expansion step Δx, maximum expansion distance d_{cmax};

Ensure: Optimized collision-free path $\mathcal{P}_{\text{opt}}$;

1: $\mathcal{P}_{\text{cf}}, \mathcal{P}_{\text{cr}} \leftarrow$ ***CirclePathGen***$(\mathcal{P}_{\text{f}}, \mathcal{G}_{\text{rob}})$; {Compute front and rear center paths based on robot geometry.}
2: $\mathcal{O}_{\text{exp}} \leftarrow$ ***InflateObstacles***$(\mathcal{O}_{\mathcal{W}}, \mathcal{G}_{\text{rob}})$; {Inflate obstacles by the robot's circumscribed radius.}
3: $\Omega_{\text{seq}} \leftarrow \emptyset$;
4: **for each** point pair $(\mathbf{s}_k^f, \mathbf{s}_k^r) \in (\mathcal{P}_{\text{cf}}, \mathcal{P}_{\text{cr}})$ **do**
5: **for each** direction $d \in \{+y, +x, -y, -x\}$ **do**
6: $d_{\text{exp}} \leftarrow 0$; {Initialize expansion distance.}
7: **while** ***IsFree***$(\Omega \oplus d \cdot \Delta x, \mathcal{O}_{\text{exp}})$ **and** $d_{\text{exp}} < d_{\text{cmax}}$ **do**
8: $\Omega \leftarrow \Omega \oplus d \cdot \Delta x$;
9: $d_{\text{exp}} \leftarrow d_{\text{exp}} + \Delta x$;
10: **end while**
11: **end for**
12: $\Omega_{\text{seq}} \leftarrow \Omega_{\text{seq}} \cup \{\Omega\}$;
13: **end for**
14: $\mathcal{C}_{\text{seq}} \leftarrow$ ***MergeRectangles***(Ω_{seq}); {Construct the complete safe corridor by merging adjacent rectangles.}
15: $\mathcal{P}_{\text{opt}} \leftarrow$ ***SolveGMPPO***$(\mathcal{P}_{\text{f}}, \mathcal{C}_{\text{seq}}, \mathcal{G}_{\text{rob}})$; {Use Model Predictive to solve the optimal path within safe corridor.}
16: **return** $\mathcal{P}_{\text{opt}}$

This article proposes a dynamic safety corridor construction method based on robot geometry, which guides the selection of feasible and collision-free waypoints along the path. Using a bi-disc model to approximate the robot contour, front and rear center trajectories are extracted. Combined with local obstacle data, an adaptive safety region defines the navigable space. On this basis, we develop a nonlinear model predictive approach tailored to safe-corridor path planning. The safe corridor is imposed as a hard constraint within a kinematic optimization framework, allowing dynamic trajectory adaptation through a receding-horizon scheme. This formulation guarantees obstacle avoidance, dynamic feasibility, and smoothness of the resulting trajectories. The overall procedure is summarized in Algorithm 2.

In our study, obstacles are first modeled as convex polygons and then inflated by a radius O_c^r, determined by (6), to account for the robot's size and safety margin. Based on the front-end generated path and the bi-disc geometric model [see (4) and (5)], front-center and rear-center trajectories ($\mathcal{P}_{\text{cf}}$ and $\mathcal{P}_{\text{cr}}$) are then constructed, along which dynamic safety corridors are generated.

Specifically, as shown in Fig. 4, our method adaptively generates rectangular regions based on waypoint proximity, allowing

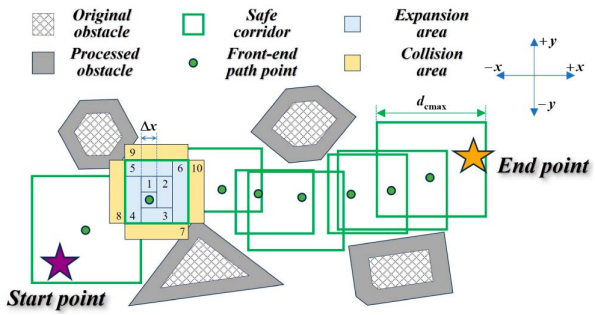


Fig. 4. Construction process of the safety corridor.

shared regions for nearby points. For each waypoint s_k, a square of length Δx centered at s_k is initialized and iteratively expanded along the $+y$, $+x$, $-y$, and $-x$ axes (Algorithm 2, Line 3–13). The Δx depends on the workspace scale, with larger values in expansive areas to reduce computation and smaller values in confined spaces to better utilize the space. Each expansion triggers immediate collision checking: if no collision is detected, the region is expanded; otherwise, growth in that direction stops. The expansion process terminates once the growth along any axis reaches the predefined maximum limit d_{cmax} or a potential collision is detected. The resulting bounds along each direction define the maximal feasible safety corridor for the corresponding waypoint. After all waypoints are processed, the overall safety corridor $\mathcal{C}_{\text{seq}}$ is obtained by assembling the individual rectangular regions using $\boldsymbol{MergeRectangles}(\Omega_{\text{seq}})$ function (Algorithm 2, Line 14), which fuses adjacent rectangles with overlapping intervals into a continuous corridor, ensuring that every waypoint $s_k^f \in \mathcal{P}_{\text{cf}}$ and $s_k^r \in \mathcal{P}_{\text{cr}}$ lies strictly within its assigned corridor segment.

To enable time-parametric modeling, each point on the reference path $\mathcal{P}_{\text{f}} = \{s_k\}_{k=0}^{N_p}$ is assigned a uniform timestamp $t_k = k\Delta t$, where $\Delta t = T/h$, with T as the total planning time and h the number of segments. This yields a time-annotated trajectory $\mathcal{T} = \{\hat{s}_k = (s_k, t_k)\}_{k=0}^{N_p}$, which serves as the basis for time-domain optimization. A model predictive trajectory optimization framework is constructed by discretizing the robot's nonlinear kinematic model and incorporating safety constraints. The objective minimizes tracking error and control variation, while terminal, kinematic, corridor, and input constraints ensure geometric and dynamic feasibility. The final optimization problem is formulated as follows:

$$\begin{aligned}
\min \quad & \mathcal{J}_m = \sum_{j=1}^{N_t-1} \Delta u_j^{\mathrm{T}} R_m \Delta u_j + \sum_{j=1}^{N_t} \tilde{z}_j^{\mathrm{T}} Q_m \tilde{z}_j \\
\text{s.t.} \quad & \mathbf{s}_0 = \mathbf{s}_{\text{start}}, x_{N_p} = x_{\text{goal}}, y_{N_p} = y_{\text{goal}} \\
& \sin\theta_{N_p} = \sin(\theta_{\text{goal}}), \cos\theta_{N_p} = \cos(\theta_{\text{goal}}) \\
& \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t(f(\mathbf{x}_k) + \mathbf{B}\mathbf{u}_k) \\
& \mathcal{R}(\mathbf{s}_k) \in C_{\text{seq}} \\
& u_k \in U,
\end{aligned} \tag{8}$$

where N_t is the prediction horizon, $R_m = \text{diag}[r_a, r_\omega]$ and $Q_m = \text{diag}[q_x, q_y, q_\theta]$ are diagonal weighting matrices that penalize control input changes and state deviations, respectively. The diagonal form allows for independent tuning of each penalty. In cluttered or narrow environments, increasing q_x and q_y tightens positional constraints, enhancing geometric feasibility and obstacle avoidance. In curved sections, a larger q_θ ensures smoother heading transitions and mitigates abrupt steering. In this work, the weighting matrices R_m and Q_m are initially selected based on standard model predictive control tuning rules. They are then further adjusted through iterative experiments in representative cluttered environments to achieve a balanced trade-off between control smoothness and state tracking accuracy.

Although the objective function is quadratic, the nonlinear corridor boundaries and collision-avoidance constraints render the problem a nonlinear optimization task. The function $\boldsymbol{SolveGMPPO}(\mathcal{P}_{\text{f}}, \mathcal{C}_{\text{seq}}, \mathcal{G}_{\text{rob}})$ (Algorithm 2, Line 15) assembles these components into the final constrained optimization problem and employs the interior-point solver IPOPT, together with CppAD for automatic differentiation, to efficiently compute the optimal collision-free path within the safe corridor.

C. Completeness and Feasibility Analysis

The proposed GMPPO framework comprises a front-end ECFS algorithm (Algorithm 1) and a back-end safe corridor-based model predictive path optimization algorithm (Algorithm 2). We establish the theoretical guarantees of the framework through two complementary properties: resolution completeness of the front-end path generation and feasibility guarantee of the back-end trajectory optimization.

Theorem 1 (Resolution Completeness of the Front-end Algorithm): Under the following conditions, Algorithm 1 is resolution complete.

1) *Conservative Obstacle Approximation*: The simplified obstacle set satisfies

$$\mathcal{O}_{\text{sim}} \subseteq \mathcal{O}_{\mathcal{W}} \oplus \mathcal{B}(r_{\text{cons}}), \quad r_{\text{cons}} \leq \frac{\rho_{\min}}{4} \tag{9}$$

where $\oplus$ denotes the Minkowski sum, $\rho_{\min}$ is the minimum collision-free passage width in the environment, and $\mathcal{B}(r)$ represents a closed ball of radius r.

2) *Replanning Resolution Condition*: The grid resolution Δs of the A* search satisfies

$$\Delta s \leq \frac{\rho_{\min} - 2r_{\text{cons}}}{2\sqrt{2}} \tag{10}$$

ensuring that feasible regions remain connected in the discretized space for 2-D planning.

3) *Path Connectivity*: There exists a continuous collision-free path $\mathcal{P}^* \subset \mathcal{C}_{\text{free}}$ connecting the initial and goal configurations.

Proof 1 (Resolution Completeness of the Front-end Algorithm): The resolution completeness of Algorithm 1 is established as follows.

1) *Connectivity Preservation Analysis*: Under Condition 1), approximating obstacles conservatively with radius

$r_{\text{cons}} \leq \rho_{\min}/4$ ensures a minimum passage width in the simplified free space of

$$\rho_{\text{sim}} = \rho_{\min} - 2r_{\text{cons}} \geq \frac{\rho_{\min}}{2}. \tag{11}$$

Combined with Condition 2), where the grid resolution satisfies $\Delta s \leq \rho_{\text{sim}}/(2\sqrt{2})$, the maximum discretization error from any continuous point to the nearest grid node is bounded by

$$r_{\text{disc}} = \frac{\sqrt{2}\Delta s}{2} \leq \frac{\rho_{\text{sim}}}{4}. \tag{12}$$

This ensures that feasible narrow passages remain connected in the discretized space and are not broken due to overly coarse sampling.

2) *A* Completeness Guarantee*: The ECFS algorithm employs A* search for local replanning when collision segments are detected. Since A* is a systematic graph search algorithm and given that Conditions 1) and 2) preserve connectivity in the discretized space, A* will find a solution path for each collision segment when Condition 3) is satisfied.

As the grid resolution $\Delta s \to 0$, the discretized solution converges to a continuous feasible path

$$\lim_{\Delta s \to 0} d(\mathcal{P}_{\text{ecfs}}, \mathcal{P}^*) = 0 \tag{13}$$

where $d(\cdot)$ is a suitable path distance metric. Hence, the algorithm satisfies the condition of resolution completeness.

Theorem 2 (Feasibility Guarantee of the Back-End Algorithm): Given a collision-free reference path $\mathcal{P}_{\text{f}} \subset \mathcal{C}_{\text{free}}$ generated by the front-end, Algorithm 2 satisfies

1) Safety Corridor Construction: The rectangular expansion procedure constructs a finite sequence of convex corridors $\Omega_{\text{seq}} = \{\Omega_k\}_{k=1}^{N}$ such that

$$\mathcal{P}_{\text{f}} \oplus \mathcal{B}(r_{\text{rob}}) \subseteq \bigcup_{k=1}^{N} \Omega_k \subseteq \mathcal{C}_{\text{free}} \tag{14}$$

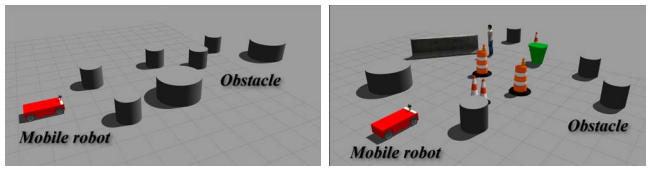
where $\mathcal{B}(r_{\text{rob}})$ is a closed ball of radius r_{rob}, and $\oplus$ denotes the Minkowski sum.

2) Feasibility of GMPPO: Constraining the optimization within the constructed safety corridor, the GMPPO problem

$$\begin{aligned} \min_{\mathcal{P}_{\text{opt}}} \quad & \|\mathcal{P}_{\text{opt}} - \mathcal{P}_{\text{f}}\|_{\mathcal{H}^2} \\ \text{s.t.} \quad & \mathcal{P}_{\text{opt}} \in \bigcup_{k=1}^{N} \Omega_k \end{aligned} \tag{15}$$

admits at least one feasible solution.

Proof 2: According to the rectangular expansion procedure in Algorithm 2, each constructed corridor Ω_k is convex and collision-free. By inflating each waypoint with clearance radius r_{rob}, the union of corridors covers the swept volume: $\mathcal{P}_{\text{f}} \oplus \mathcal{B}(r_{\text{rob}}) \subseteq \bigcup_{k=1}^{N_p} \Omega_k \subseteq \mathcal{C}_{\text{free}}$, establishing property 1. Since $\mathcal{P}_{\text{f}}$ lies entirely within the constructed safety corridor by construction, it constitutes a feasible solution to the GMPPO problem, ensuring feasibility and establishing property 2.



(a) (b)

Fig. 5. Simulation scenario: (a) scene one; and (b) scene two.

TABLE I
FRONT-END ALGORITHM SIMULATION RESULTS

Scenario	Algorithm	Safety	Consumption Time (ms, Mean ± SD)	Maximum Curvature (m^{-1})	Average Curvature (m^{-1})	Path Length (m)
Scenario One	Ref	Yes	–	2.87	0.24	15.01
	A*	Yes	8.67±0.46	3.77	0.38	15.45
	IRRT*	Yes	582.46±7.54	0.90	0.25	15.35
	CFS	Yes	14.19±1.29	0.65	0.07	14.59
	ECFS	Yes	6.87±1.43	0.65	0.07	14.59
Scenario Two	Ref	No	–	1.81	0.23	14.98
	A*	Yes	7.41±1.42	4.96	0.50	15.57
	IRRT*	Yes	485.48±5.15	3.19	0.33	15.13
	CFS	No	14.53±0.98	0.12	0.02	9.03
	ECFS	Yes	7.37±1.21	2.49	0.25	15.01

IV. BENCHMARK AND SIMULATIONS

To verify the feasibility and effectiveness of the proposed method, extensive simulations were performed in obstacle-constrained environments.

A. Simulation Settings

To evaluate the proposed method, simulations were conducted on a laptop with Ubuntu 20.04, an Intel Core i7-8565U CPU, and an NVIDIA GeForce MX150 GPU. Two scenarios with varying planning complexity were created, featuring randomly generated obstacle shapes and positions, as illustrated in Fig. 5.

For comparative evaluation, considering the stringent requirements of safety, real-time performance, and stability in industrial applications, this study selected widely adopted, industry-validated algorithms as baselines. Independent performance assessments were conducted for the front-end coarse planner, and the complete planning framework. To ensure fairness and reduce randomness, all tests were performed under a unified robotic architecture with standardized perception, localization, and control modules. Each algorithm was run 50 times, with stage-wise computation time recorded and key metrics statistically analyzed to evaluate overall performance.

B. Front-End Algorithm Comparison and Verification

We compare the proposed front-end algorithm (ECFS) with widely used front-end planners, A*, IRRT* and CFS [31], to highlight its incremental improvements in feasibility and path quality. Trajectories are shown in Fig. 6, and quantitative metrics are summarized in Table I.

Simulation results show the proposed ECFS algorithm outperforms the classical CFS in computational efficiency, reducing average planning time by 50.4% compared to CFS and

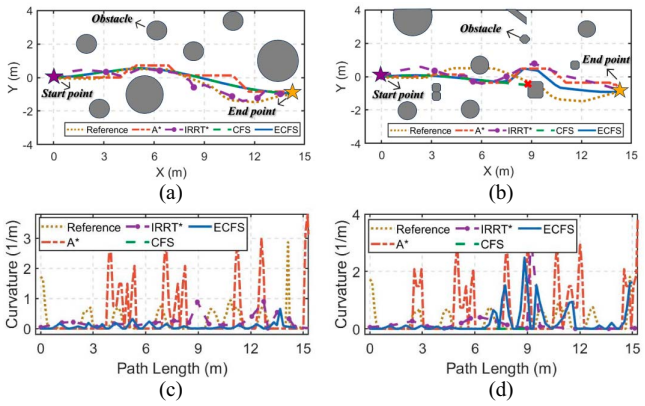


Fig. 6. Front-end algorithm simulation results: (a) path curve of scene one; (b) path curve of scene two; (c) path curvature of scene one; and (d) path curvature of scene two.

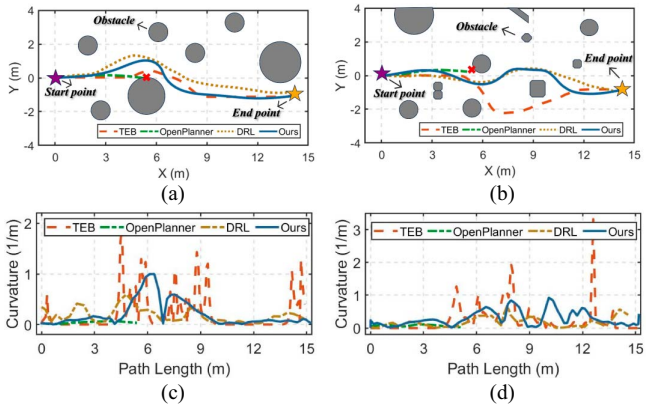


Fig. 7. Overall algorithm simulation results: (a) path curve of scene one; (b) path curve of scene two; (c) path curvature of scene one; and (d) path curvature of scene two.

TABLE II
TIME CONSUMPTION OF THE PROPOSED ALGORITHM

Scenario	Stage	Maximum (ms)	Minimum (ms)	Average (ms, mean±SD)	Confidence Intervals (ms)
Scenario One	Path generation	10.52	4.52	6.87±1.43	95%CI [6.46, 7.27]
	Corridor generation	15.01	6.23	9.80±2.25	95%CI [9.16, 10.44]
	Path optimization	18.25	16.49	17.16±0.40	95%CI [17.04, 17.27]
Scenario Two	Path generation	9.58	5.11	7.37±1.21	95%CI [7.03, 7.71]
	Corridor generation	14.49	6.13	9.59±2.03	95%CI [9.01, 10.16]
	Path optimization	24.54	19.99	21.51±0.95	95%CI [21.24, 21.78]

TABLE III
OVERALL ALGORITHM SIMULATION RESULTS

Scenario	Algorithm	Safety	Maximum Curvature (m^{-1})	Average Curvature (m^{-1})	Path Length (m)
Scenario One	TEB	Yes	1.94	0.22	14.88
	OpenPlanner	No	0.07	0.04	5.40
	DRL	Yes	0.60	0.18	14.96
	Ours	Yes	1.00	0.19	15.37
Scenario Two	TEB	Yes	3.32	0.25	15.57
	OpenPlanner	No	0.11	0.07	5.10
	DRL	Yes	0.68	0.19	14.58
	Ours	Yes	0.93	0.29	15.19

10.7% compared to A*. ECFS also produces smoother, shorter paths with lower curvature. In contrast, although IRRT* yields higher-quality paths than A* and is comparable to CFS, its inherent sampling-based nature leads to significantly higher computational cost, limiting its applicability in real-time navigation. Moreover, in complex obstacle environments where CFS fails to find a feasible solution, ECFS successfully completes the planning process and refines the initial path into a safer and more feasible trajectory.

C. Overall Algorithm Verification

To evaluate the proposed planning framework, we measured the average computation time, variance, and confidence interval for path generation, corridor construction, and trajectory optimization (see Table II).

As shown in Table II, the proposed algorithm completed collision-free path planning in 33.83 and 38.47 ms for the two validation scenarios, respectively. Although execution time increased slightly with environmental complexity, the overall performance remained well within real-time requirements. The optimization stage dominated the computational cost due to the simultaneous handling of kinematic constraints and safety guarantees. In both scenarios, the variance and 95% confidence intervals were small, indicating stable performance and high statistical reliability of the results.

To comprehensively evaluate the performance of the proposed method, we conducted a systematic comparison with three representative planners: TEB [18], OpenPlanner [12], and a deep reinforcement learning (DRL) [32] approach. It should be emphasized that the DRL method must be retrained specifically for the two selected scenarios in order to complete the planning tasks. Fig. 7 presents the resulting trajectories and curvature profiles, while the quantitative results are summarized in Table III.

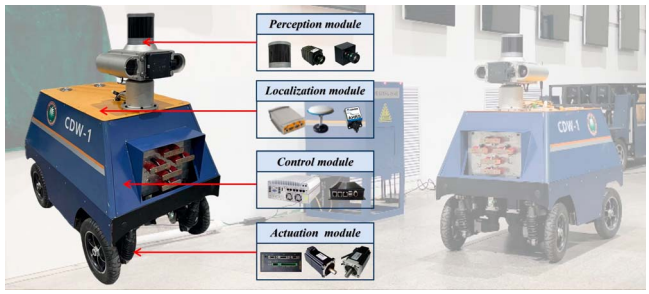


Fig. 8. Hardware components of the robot.

The results show that although OpenPlanner achieves smooth curvature, it often fails in narrow passages, causing frequent stalls near obstacles. TEB generates paths with much higher curvature, leading to greater control effort and potential instability. Our method reduces the maximum curvature by 60.2% compared with TEB, producing smoother and safer trajectories. While path lengths vary across scenarios, TEB tends to stay closer to obstacles and exhibits unnecessary oscillations,

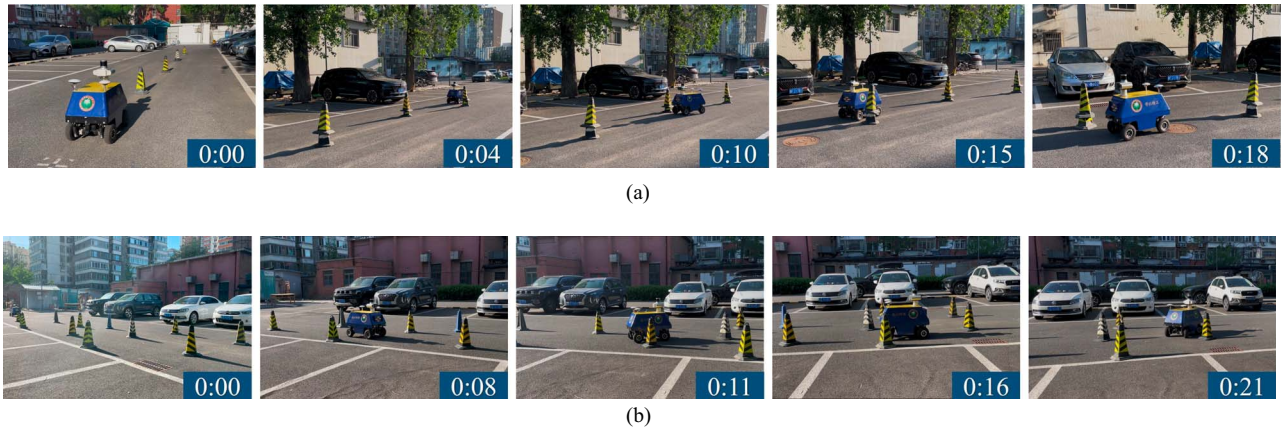


Fig. 9. Experimental verification process: (a) experimental process of scene one; and (b) experimental process of scene two.

TABLE IV
ROBOT PARAMETERS

Description	Value
Length × Width × Height (cm)	100 × 62 × 97
Wheelbase (cm)	68
Track width (cm)	52
Turning radius (cm)	34
Max linear speed (m/s)	5
Max steering angle (°)	45

TABLE V
EXPERIMENTAL RESULT

Scenario	Algorithm	Safety	Maximum Curvature (m^{-1})	Average Curvature (m^{-1})	Path Length (m)	Completion Time (s)
Scenario One	Ref	No	0.51	0.14	18.32	-
	TEB	Yes	1.82	0.39	19.12	20.8
	Ours	Yes	1.30	0.26	19.56	21.4
Scenario Two	Ref	No	0.72	0.13	21.34	-
	TEB	Yes	2.00	0.35	20.69	23.4
	Ours	Yes	0.81	0.22	21.27	24.1

increasing collision risk. In contrast, the DRL-based method performs well quantitatively but requires scenario-specific tuning and retraining, limiting its generalizability. Overall, the proposed approach provides a more balanced and practical solution.

V. EXPERIMENTS AND DEPLOYMENT

To validate the method's practical feasibility and engineering adaptability, tests were first conducted in a lab environment and then verified in a real-world application scenario.

A. Experimental Settings

To evaluate the method's applicability in real-world scenarios, experiments were conducted in two representative environments: a sparse-obstacle setting and a dense-obstacle setting. The experiments were carried out using a custom four-wheel independent drive and steering robot (see Fig. 8). The robot integrates perception, localization, control, and actuation modules. Its perception system comprises a LiDAR and RGB/IR cameras. Localization is achieved through a GPS unit and an IMU, supporting both GPS-based and SLAM-based positioning with real-time obstacle detection. The control system includes an on-board computer and a low-level controller, while the actuation module consists of four steering motors and four drive motors with corresponding drivers. Detailed parameters of the platform are provided in Table IV.

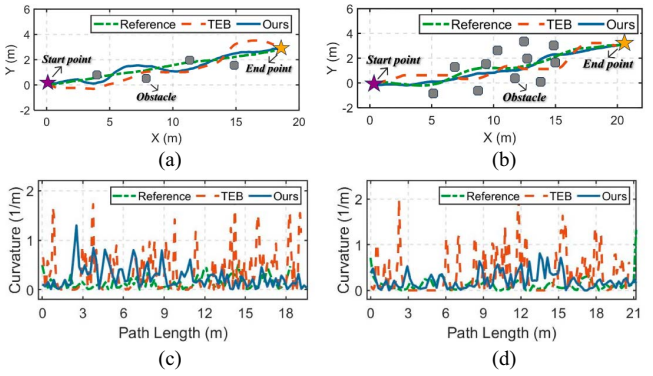


Fig. 10. Algorithm experiment verification results: (a) path curve of scene one; (b) path curve of scene two; (c) path curvature of scene one; and (d) path curvature of scene two.

B. Experimental Verification

The proposed algorithm was evaluated in both sparse and dense obstacle environments. Localization was provided by a GPS-based system, and path tracking was performed using a pure pursuit controller, with both localization and tracking accuracy meeting practical requirements. Experimental setups and results are presented in Figs. 9, 10, and Table V, respectively.

Experimental results show that in both scenarios, the reference paths, although low in curvature, resulted in collisions and failed to yield feasible trajectories. In contrast, both the

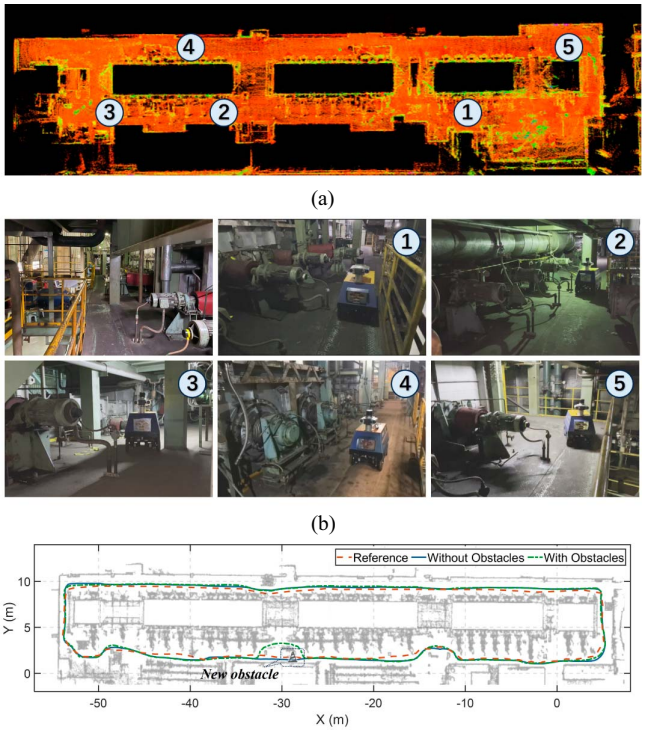


Fig. 11. Application results of algorithm deployment: (a) point cloud map; (b) experimental process; and (c) path curve.

proposed algorithm and TEB completed the tasks without collisions. TEB produced slightly shorter paths, but its maximum curvature was higher, and its average curvature exceeded that of the proposed method by 50% and 59%, respectively, indicating reduced smoothness. This led to frequent sharp turns, increased control effort, and compromised safety due to closer proximity to obstacles.

C. Application Deployment

The proposed algorithm and robotic system were deployed for real-world inspection in the narrow and enclosed annealing furnace area at Baosteel, Shanghai. The robot employed LIO-SAM [33] for localization and a pure pursuit controller for tracking. A reference path manually generated from a point cloud map was used, albeit with significant drawing errors. Additionally, artificial obstacles were placed along the route to evaluate obstacle avoidance capabilities. The inspection environment and trajectory are illustrated in Fig. 11.

As shown in Fig. 11, the manually drawn reference path overlaps with multiple on-site structures and lacks smoothness due to sparse waypoints, rendering it unsuitable under robot kinematic constraints. In contrast, the paths generated by the proposed algorithm, both with and without obstacles, enabled successful, collision-free navigation and inspection. The obstacle-free path measured 131.93 m, while the obstacle-avoidance path extended slightly to 134.62 m due to detours. Despite the marginal increase in length, the generated trajectories were smooth, exhibited continuous curvature, and satisfied kinematic constraints, demonstrating the algorithm's robustness and adaptability in complex real-world environments.

VI. Conclusion

This article tackles the path planning problem for mobile robots in obstacle-constrained environments by proposing a GMPPO framework. The front-end decouples kinematic and obstacle constraints, integrating an A*-based repair mechanism to ensure path connectivity and feasibility. The back-end constructs dynamic safety corridors using a bi-disc robot model and real-time obstacle data, which, combined with the reference path, guide model predictive optimization to ensure nonholonomic and dynamic feasibility. The method is theoretically proven to be probabilistically complete and has been extensively validated through simulations and real-world experiments. Results show clear advantages in feasibility, smoothness, safety, and efficiency over mainstream methods, highlighting its strong engineering practicality and deployment potential.

Future work will focus on handling moving obstacles and incorporating perception uncertainty in dynamic environments. Learning-based methods will also be explored for real-time safety corridor generation and adaptive obstacle abstraction to improve robustness and efficiency in complex scenarios.

References

[1] R. Chand, B. Sharma, and S. A. Kumar, "Systematic review of mobile robots applications in smart cities with future directions," *J. Ind. Inf. Integration*, vol. 45, 2025, Art. no. 100821.
[2] H. Hu, Z. Shen, and C. Zhuang, "A pinn-based friction-inclusive dynamics modeling method for industrial robots," *IEEE Trans. Ind. Electron.*, vol. 72, no. 5, pp. 5136–5144, May 2025.
[3] L. Liu, X. Wang, X. Yang, H. Liu, J. L, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst. Appl.*, vol. 227, 2023, Art. no. 120254.
[4] Z. Zhang, G. Cheng, Z. Guo, H. R. Karimi, and Y. Lu, "Parallel parking path planning and trajectory tracking in narrow environments for autonomous unmanned vehicles," *Optim. Control Appl. Methods*, vol. 45, pp. 341–352, 2024.
[5] S. Zhang, R. Cui, W. Yan, and Y. Li, "Dual-layer path planning with pose slam for autonomous exploration in gps-denied environments," *IEEE Trans. Ind. Electron.*, vol. 71, no. 5, pp. 4976–4986, May 2023.
[6] R. Zhang, R. Chai, S. Chai, Y. Xia, and A. Tsourdos, "Design and practical implementation of a high efficiency two-layer trajectory planning method for AGV," *IEEE Trans. Ind. Electron.*, vol. 71, no. 2, pp. 1811–1822, Feb. 2023.
[7] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Rob. Auton. Syst.*, vol. 174, Art. no. 104630, 2024.
[8] A. Sahin and S. Bhattacharya, "Topo-geometrically distinct path computation using neighborhood-augmented graph, and its application to path planning for a tethered robot in 3d," *IEEE Trans. Robot.*, vol. 3, pp. 421–437, Mar. 2024.
[9] Y. Liu et al., "Hybrid path planning method for USV based on improved A-Star and DWA," *J. Mar. Sci. Eng.*, vol. 13, no. 5, 2025, Art. no. 934.
[10] J. Zhang, H. Ling, Z. Tang, W. Song, and A. Lu, "Path planning of USV in confined waters based on improved A* and DWA fusion algorithm," *Ocean Eng.*, vol. 322, 2025, Art. no. 120475.
[11] Y. Su, J. Xin, and C. Sun, "Dynamic path planning for mobile robots based on improved RRT* and DWA algorithms," *IEEE Trans. Ind. Electron.*, vol. 36, no. 4, 2025, Art. no. 14.
[12] X. Yao, Y. Bai, B. Zhang, D. Xu, G. Cao, and Y. Bian, "Autonomous navigation and adaptive path planning in dynamic greenhouse environments utilizing improved LeGO-LOAM and openplanner algorithms," *J. Field Rob.*, vol. 41, no. 7, pp. 2427–2440, 2024.
[13] N. Hiraoka, H. Ishida, T. Hiraoka, K. Kojima, K. Okada, and M. Inaba, "Sampling-based global path planning using convex polytope approximation for narrow collision-free space of humanoid," *Int. J. Humanoid Rob.*, vol. 21, no. 6, 2024, Art. no. 2450005.
[14] Y. Zhong, X. Zhu, J. Huang, Y. Ye, Z. Wang, and Y. Wang, "GVDF-RRT*: An improved F-RRT* path planning algorithm based on gen-

eralized Voronoi diagram," *IEEE Access*, vol. 13, pp. 78968–78981, 2025.
[15] P. Guo, C. Sun, and Q. Li, "Obstacle avoidance path planning in unstructured environment with narrow passages," *IEEE Trans. Intell. Veh.*, vol. 8, no. 11, pp. 4632–4643, Nov. 2023.
[16] S. Zhang, R. Cui, W. Yan, and Y. Li, "RA-RRTV*: Risk-averse RRT* with local vine expansion for path planning in narrow passages under localization uncertainty," *IEEE Robot. Automat. Lett.*, vol. 10, no. 9, pp. 2072–2079, Sep. 2025.
[17] X. Hai, Z. Zhu, Y. Liu, A. W. Khong, and C. Wen, "Resilient real-time decision-making for autonomous mobile robot path planning in complex dynamic environments," *IEEE Trans. Ind. Electron.*, vol. 72, no. 11, pp. 11551–11562, Nov. 2025.
[18] Z. Zhu, Q. Zhang, Y. Song, Y. Yang, and J. Liu, "STC-TEB: Spatial-temporally complete trajectory generation based on incremental optimization," *IEEE Robot. Automat. Lett.*, vol. 21, no. 5, pp. 422–430, May 2024.
[19] F. Huo, S. Zhu, H. Dong, and W. Ren, "A new approach to smooth path planning of Ackerman mobile robot based on improved ACO algorithm and b-spline curve," *Rob. Auton. Syst.*, vol. 175, 2024, Art. no. 104655.
[20] R. Chai et al., "A two phases multiobjective trajectory optimization scheme for multi-UGVS in the sight of the first aid scenario," *IEEE Trans. Cybern.*, vol. 23, no. 5, pp. 434–444, May 2024.
[21] Y. Li, G. Li, and X. Wang, "Research on trajectory planning of autonomous vehicles in constrained spaces," *Sensors*, vol. 24, no. 17, 2024, Art. no. 5746.
[22] G. Kulathunga et al., "Navigating narrow spaces: A comprehensive framework for agricultural robots," *IEEE Robot. Automat. Lett.*, vol. 10, no. 9, pp. 9296–9303, Sep. 2025.
[23] Z. Zhang, G. Cheng, Z. Guo, H. R. Karimi, and Y. Lu, "Parallel parking path planning and trajectory tracking in narrow environments for autonomous unmanned vehicles," *Optim. Control Appl. Methods*, vol. 46, no. 3, pp. 1235–1248, 2025.
[24] Z. Zhang, H. Fu, J. Yang, and Y. Lin, "Deep reinforcement learning for path planning of autonomous mobile robots in complicated environments," *Complex Intell. Syst.*, vol. 11, no. 6, pp. 1–15, 2025.
[25] B. Xue, F. Zhou, C. Wang, M. Gao, and L. Yin, "Robot mapless navigation in VUCA environments via deep reinforcement learning," *IEEE Trans. Ind. Electron.*, vol. 23, no. 6, pp. 456–463, Jun. 2024.
[26] M. Denesh Babu, C. Maheswari, and B. M. Priya, "Dynamic robot navigation in confined indoor environment: unleashing the perceptron-Q learning fusion," *Sensors*, vol. 25, no. 20, 2025, Art. no. 6384.
[27] T. Nakamura, M. Kobayashi, and N. Motoi, "Path planning for mobile robot considering turnabouts on narrow road by deep q-network," *IEEE Access*, vol. 11, pp. 19111–19121, 2023.
[28] H. Li, P. Chen, G. Yu, B. Zhou, Y. Li, and Y. Liao, "Trajectory planning for autonomous driving in unstructured scenarios based on deep learning and quadratic optimization," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 4886–4903, Apr. 2023.
[29] Y. Xu, L. Zhang, Z. Liu, S. Wang, and J. Wang, "Design and development of a new autonomous transportation robot for finished vehicles docking transportation in RO/RO logistics terminal," *Adv. Eng. Inf.*, vol. 66, 2025, Art. no. 103391.
[30] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Piscataway, NJ, USA: IEEE Press, 2018, pp. 4758–4765.
[31] C. Liu, C.-Y. Lin, Y. Wang, and M. Tomizuka, "Convex feasible set algorithm for constrained trajectory smoothing," in *Proc. Am. Control Conf. (ACC)*. Piscataway, NJ, USA: IEEE Press, 2017, pp. 4177–4182.
[32] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 1140–1146.
[33] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Piscataway, NJ, USA: IEEE Press, 2020, pp. 5135–5142.

Lin Zhang received the B.S. and M.S. degrees in engineering from Yanshan University, Qinhuangdao, China, in 2020 and 2023, respectively. He is currently working toward the Ph.D. degree in control science and engineering with the School of Automation, Beijing Institute of Technology, Beijing, China.

He is a member of the State Key Laboratory of Intelligent Control and Decision of Complex Systems. His research interests include multiagent cooperative planning and the control of special unmanned systems.

Wenhao Zhang received the B.Eng. degree in engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2022. He is currently working toward the M.S. degree in control science and engineering with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, China.

His research interests include motion and control of unmanned systems.

Runjiao Bao received the B.Eng. degree in engineering from Jilin University, Changchun, China, in 2023. He is currently working toward the M.S. degree in control science and engineering with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, China.

His research interests include trajectory planning and control.

Tianwei Niu received the B.S. degree from the College of Mechanical Engineering and Mechanics, Ningbo University, Ningbo, China, in 2018, and the M.S. degree from the College of Mechanical Engineering, University of Science and Technology Beijing, Beijing, China, in 2022, both in vehicle engineering. He is currently working toward the Ph.D. degree in control science and engineering with the School of Automation, Beijing Institute of Technology, Beijing.

He is a member of the State Key Laboratory of Intelligent Control and Decision of Complex Systems. His research interests include localization and motion planning for field robots.

Shoukun Wang received the B.S., M.S., and Ph.D. degrees in automation from Beijing Institute of Technology, Beijing, China, in 1999, 2002, and 2004, respectively.

He is currently a Professor and a Ph.D. Supervisor with the School of Automation, Beijing Institute of Technology. He was a Visiting Scholar with the Department of Electronics and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests include electro-hydraulic control algorithms, robot locomotion control, and visual servo systems.

Junzheng Wang received the Ph.D. degree in control science and engineering from Beijing Institute of Technology, Beijing, China, in 1994.

He is currently a Professor and a Ph.D. Supervisor and serves as the Deputy Director of the Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology. His research interests include motion control, performance testing of electric and electro-hydraulic servo systems, and dynamic target detection and tracking based on image processing.