processes

MDPI

Article

A Predictive and Risk-Adaptive Dynamic Window Approach for Robot Navigation in Constrained Environments

Yongtao Qiu, Zheng Zhang *, Yong Zeng and Ruhong Ma

School of Mechanical Engineering, Yancheng Institute of Technology, Yancheng 224051, China; qiuyongtao@ycit.edu.cn (Y.Q.); zengyong@ycit.edu.cn (Y.Z.); 310011@ycit.edu.cn (R.M.)
* Correspondence: zz020808zz@163.com; Tel.: +86-17714926007

Abstract

Reliable robot navigation in spatially constrained dynamic environments requires coordinated global guidance, local planning, and obstacle prediction. This study proposes a predictive and risk-adaptive Dynamic Window Approach (DWA) to address fixed evaluation weights and limited anticipation of obstacle motion. An asymmetric feasible region is constructed along the global reference path from inflated static obstacles and map boundaries to constrain local targets and candidate trajectories. Robot trajectories and obstacle motions are assessed at synchronized future instants. Predicted safety margins support trajectory screening, while a composite risk measure incorporating approximate time-to-collision adjusts evaluation weights. Passage decisions coordinate waiting, safe traversal, and global replanning under temporary or persistent blockage. Simulations included mechanism validation and comparisons with five DWA-based baselines. The comprehensive experiment comprised 30 randomized trials per method. The proposed method achieved an 80.0% success rate, compared with 56.7% for Time-Aligned Prediction (TAP). Among successful collision-free runs, the proposed method achieved a mean tracking error of 0.269 m, a mean minimum dynamic safety margin of 0.541 m, and a mean arrival time of 75.80 s; TAP's corresponding mean arrival time was 56.86 s. These results indicate a trade-off between task-completion robustness and traversal efficiency under the tested conditions.

Keywords: mobile robot navigation; dynamic window approach; asymmetric feasible region; dynamic obstacle prediction; risk-adaptive evaluation; global replanning

1. Introduction

Mobile robots are increasingly used in intelligent manufacturing, warehouse logistics, autonomous delivery, and automated inspection, where navigation environments are becoming more dynamic and spatially constrained [1,2]. In such scenarios, robots must follow a feasible global route while responding online to moving obstacles, temporary blockage, and local changes in traversable space. These challenges are particularly significant in narrow passages and obstacle-dense regions, where insufficient local decision-making may lead to premature stopping, excessive detouring, oscillation, or navigation failure.

Mobile-robot path planning generally consists of global and local planning. Global planners such as A*, Dijkstra, and Rapidly-exploring Random Tree (RRT) provide macroscopic route guidance [3–6], whereas local planners generate real-time motion commands according to the surrounding environment.

Learning-based methods have also been investigated for robot path planning in environments with obstacles. Tang and German [7] enhanced tabular Q-learning for grid

Check for updates

Received: 10 October 2026
Revised: 13 November 2026
Accepted: 15 November 2026
Published: 18 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

navigation through directional initialization, an annealed exploration policy, and potential-based reward shaping. Their study provides a complementary perspective on improving learning efficiency and path quality in grid environments.

Among representative local planning methods [8–11], the Dynamic Window Approach (DWA) is widely used because it directly searches feasible linear and angular velocities while accounting for robot kinematic and dynamic constraints [11]. However, conventional DWA still has several limitations in complex dynamic environments. Its obstacle evaluation mainly relies on current or short-horizon environmental information, fixed evaluation weights provide limited adaptability to changing risk levels, and the feasible space surrounding the global reference path is not explicitly represented. Consequently, the robot may respond late to approaching obstacles, behave overly conservatively, or fail to exploit available lateral space in constrained regions.

To improve anticipatory obstacle avoidance, recent studies have increasingly incorporated dynamic-obstacle prediction into DWA. Missura and Bennewitz [12] introduced predicted obstacle motion into DWA trajectory evaluation, while Kobayashi and Motoi [13] combined predicted obstacle positions with virtual manipulators. Hahn [14] incorporated short-term obstacle-motion prediction and future-distance information, and Tao et al. [15] proposed Time-Aligned Prediction (TAP) to evaluate robot and obstacle trajectories at synchronized future instants. Chen et al. [16] further introduced probabilistic collision prediction to account for motion uncertainty, whereas Motoi and Ueda [17] incorporated collision prediction time into local collision assessment. These studies highlight the importance of predicting robot–obstacle interactions over time. For navigation in constrained spaces, an additional planning requirement is to relate the predicted interaction risk to the available lateral space and use both when evaluating candidate trajectories.

Considerable attention has also been devoted to improving the adaptability of DWA trajectory evaluation. Sun et al. [9] used fuzzy logic to adjust trajectory-evaluation weights according to environmental conditions. Cao and Nor [18] improved DWA for dynamic obstacle avoidance, while Yang et al. [19] introduced a risk-degree function to adaptively regulate trajectory evaluation. Zhai et al. [20] developed a safety-distance matrix with an adaptive weight-adjustment strategy for real-time planning. Subsequent studies further incorporated application-specific dynamic obstacle avoidance, fuzzy decision mechanisms, energy-related considerations, and multi-strategy enhancement into DWA [21–23]. These studies support the use of environmental risk to adapt trajectory evaluation. This motivates the joint consideration of geometric constraints, spatial clearance, and temporal interaction risk, so that weight adjustment is coordinated with candidate-trajectory screening and local-target selection.

Global–local planning coordination and navigation recovery have likewise been widely investigated. Hou et al. [24], Zhou et al. [25], and Lu and Da [26] combined DWA with global optimization or search strategies to improve global–local coordination. Lv et al. [27] considered safety-oriented path planning in real-time dynamic environments, while Qin et al. [28] integrated an enhanced A* algorithm with predictive DWA. In addition, Jiang et al. [29] improved look-ahead target selection for path tracking, and Tonola et al. [30] developed an informed multi-path replanning strategy for changing environments. These studies demonstrate the value of coordinating global guidance, local planning, and navigation recovery. For dynamically blocked passages, the present study further considers how to select among waiting, traversal, and global replanning according to predicted reference-path occupancy and the persistence of blockage.

These developments motivate a closer examination of how spatial constraints, predicted interactions, and recovery decisions influence one another. When moving obstacles occupy a reference path in a constrained area, a robot must assess both the remaining lateral space and

the timing of a possible interaction. A locally admissible motion may also fail to restore task progress under persistent blockage. Coordinating these spatial, temporal, and task-level decisions therefore forms the research problem addressed in this study.

The objective of this study is to improve navigation reliability, dynamic safety, and path-tracking performance in spatially constrained environments with moving obstacles. Specifically, the study aims to make effective use of asymmetric free space around the global reference path, anticipate interactions with moving obstacles, and maintain task progress when the reference route is temporarily or persistently blocked. To achieve these objectives, a predictive and risk-adaptive DWA is developed by integrating asymmetric feasible-region constraints, local-target selection, synchronized obstacle-motion prediction, risk-adaptive trajectory evaluation, and passage and replanning decisions.

The main contributions of this work are summarized as follows:

1. An asymmetric feasible region and a target-selection strategy within this region are developed. The feasible widths on both sides of the global reference path are independently determined according to static-obstacle distribution and map boundaries. Combined with corner-aware and risk-driven target adjustment, the method enables the robot to exploit lateral free space while maintaining global-path guidance.

2. A predictive and risk-adaptive trajectory-evaluation strategy is developed for DWA. Robot trajectories and dynamic-obstacle motion are evaluated at synchronized future instants, while dynamic safety margin and ATTC characterize spatial and temporal interaction risk. The resulting risk is used to adapt the relative importance of safety, efficiency, and motion stability during local trajectory selection.

3. A dynamic passage and navigation-recovery mechanism is introduced for constrained passages. Passage-entry admission and safe-passage-window detection are used to handle temporary blockage, whereas a waiting–replanning decision mechanism triggers global route updating when the original path remains persistently unavailable.

2. Fundamentals of the Dynamic Window Approach

The Dynamic Window Approach (DWA) is a real-time local planning method that searches feasible linear and angular velocity commands while considering robot kinematic, dynamic, and braking-safety constraints [11]. Candidate trajectories are predicted over a short horizon, and the velocity pair with the highest evaluation score among the admissible candidates is selected as the control command.

For a differential-drive mobile robot moving in a two-dimensional plane, the state is represented as $X = [x,y,\theta]^T$, and the control input is u $= [v,\omega]^T$. Within a sampling interval Δt, the robot motion is described by

$$\begin{cases} x(t+\Delta t) = x(t) + v(t)\cos\theta(t)\Delta t \\ y(t+\Delta t) = y(t) + v(t)\sin\theta(t)\Delta t \\ \theta(t+\Delta t) = \theta(t) + \omega(t)\Delta t \end{cases}. \quad (1)$$

Based on the current velocity state, the admissible velocity set is obtained by combining the physical velocity limits V_s, the acceleration-constrained reachable set V_d, and the braking-safe set V_a:

$$V = V_s \cap V_d \cap V_a. \quad (2)$$

Velocity pairs in the admissible set are sampled to generate short-horizon trajectories. After unsafe candidates are removed, the remaining trajectories are evaluated using the conventional DWA objective function:

$$G(v,\omega) = \alpha H(v,\omega) + \beta D(v,\omega) + \gamma S(v,\omega), \quad (3)$$

where $H(v,\omega)$ represents the heading consistency with the local target, $D(v,\omega)$ represents obstacle clearance, and $S(v,\omega)$ represents motion velocity. The coefficients α, β, and γ determine the relative importance of the three criteria.

Although conventional DWA provides favorable real-time performance, its application in complex dynamic environments is limited by insufficient anticipation of future obstacle motion, fixed trajectory-evaluation weights, and the lack of explicit representation of the feasible space surrounding the global reference path. These limitations motivate the predictive and risk-adaptive DWA with feasible-region constraints developed in Section 3.

3. Predictive and Risk-Adaptive DWA with Feasible-Region Constraints

To improve navigation in spatially constrained dynamic environments, this study develops a predictive and risk-adaptive DWA with feasible-region constraints. Rather than using the global reference path only as a centerline for local target guidance, the proposed method expands the path into an asymmetric traversable region according to the surrounding static obstacles and map boundaries. The resulting feasible region is subsequently integrated with look-ahead target selection, dynamic-obstacle prediction, candidate-trajectory screening, risk-adaptive evaluation, and passage-state decision-making. In this way, global-path guidance and local obstacle avoidance are coordinated within a unified planning process.

3.1. Construction of an Asymmetric Feasible Region

An asymmetric feasible region is constructed around the global reference path to represent the available lateral space for local planning in the two-dimensional workspace. Figure 1 illustrates the construction of the asymmetric feasible region along the global reference path. Its left and right widths are determined independently from inflated static obstacles and map boundaries and vary along the reference path. The region provides geometric constraints for local target selection and candidate-trajectory screening. Candidate trajectories remain subject to the robot motion constraints and separate static-collision, dynamic-collision, and terminal stopping-safety checks. The reference path is first resampled at approximately uniform spatial intervals to improve the continuity of the cross-sections of the feasible region. Static obstacles are then inflated according to the robot size and the prescribed safety clearance. For the j-th obstacle, the inflated radius is defined as

$$r_j^{inf} = r_j + r_r + d_s, \tag{4}$$

where r_j is the obstacle radius, r_r is the equivalent robot radius, and d_s is the additional safety distance. After obstacle inflation, the robot can be approximated as a point during the subsequent feasible-region construction process.

For each resampled reference-path point q_i, a unit tangent vector t_i is estimated from the neighboring path points, and the corresponding unit normal vector is defined as

$$\boldsymbol{n}_i = [-t_{i,y}, t_{i,x}]^T. \tag{5}$$

The positive and negative directions of n_i are regarded as the left and right sides of the reference path, respectively. Starting from q_i, collision-free space is searched independently in both directions until an inflated obstacle, the map boundary, or the prescribed maximum search width is reached. The resulting feasible widths are

$$W_i^L = \max\{\xi \in [0, W_{\max}] \mid \boldsymbol{q}_i + s\boldsymbol{n}_i \notin O^{\inf} \text{ for all } s \in [0, \xi)\}, \tag{6}$$

$$W_i^R = \max\{\xi \in [0, W_{\max}] \mid \boldsymbol{q}_i - s\boldsymbol{n}_i \notin O^{\inf} \text{ for all } s \in [0, \xi)\}, \tag{7}$$

where O^{inf} denotes the set consisting of all inflated static-obstacle regions together with the non-traversable regions outside the map boundaries, and $W_{\max}$ is the prescribed maximum one-sided search width.

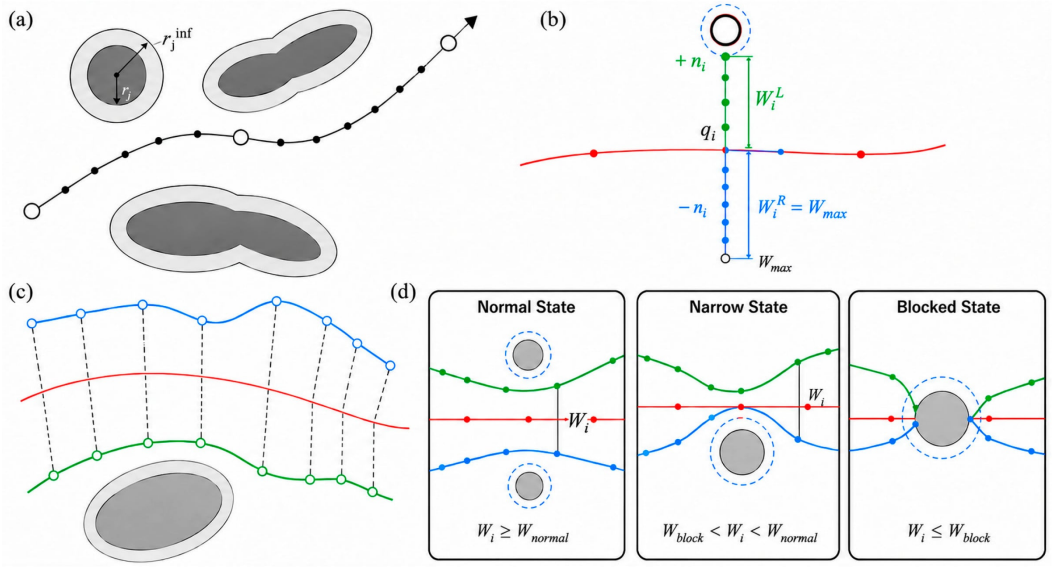


Figure 1. Schematic of asymmetric feasible-region construction: (**a**) reference-path processing and obstacle inflation; (**b**) independent searches for the left and right widths; (**c**) boundary continuity correction; (**d**) classification based on local region width. Note: The reference path is shown in black in (**a**) and in red in (**b**–**d**). Green and blue indicate the left and right sides of the feasible region, respectively. Gray filled regions represent static obstacles, and the surrounding outlines indicate the inflated obstacle boundaries. The dashed lines in (**c**) indicate local cross-sections.

In the discrete implementation, each side is sampled outward from the reference path point at 0.20 m intervals. The search stops at the first infeasible sample and retains the distance to the last feasible sample as the initial width. If the starting point is infeasible, the width is set to zero.

Because abrupt changes in obstacle geometry, map boundaries, or path curvature may produce discontinuous widths of the feasible region, the obtained width sequences W_i^L and W_i^R are locally smoothed. Sudden outward expansion is restricted, whereas rapid contraction caused by nearby obstacles is preserved to prioritize safety. After smoothing, the endpoint and the sampled points along each one-sided cross-section are checked again. If any checked point is infeasible, the width is reduced and the check is repeated. The final effective widths to the left and right of the global reference path are denoted by $\hat{W}_i^L$ and $\hat{W}_i^R$, respectively.

Accordingly, the final left and right boundaries of the feasible region are expressed as

$$\boldsymbol{B}_i^L = \boldsymbol{q}_i + \hat{W}_i^L \boldsymbol{n}_i, \tag{8}$$

$$\boldsymbol{B}_i^R = \boldsymbol{q}_i - \hat{W}_i^R \boldsymbol{n}_i. \tag{9}$$

Connecting the corrected boundary points along the reference path forms the asymmetric feasible region, whose left and right widths vary independently with the surrounding environment.

To support local planning and passage-state decisions, the total width of the feasible region at q_i is defined as

$$W_i = \hat{W}_i^L + \hat{W}_i^R. \tag{10}$$

The local width-based state of the feasible region is defined as

$$S_i = \begin{cases} 1, & W_i \geq W_{normal} \\ 2, & W_{block} < W_i < W_{normal} \\ 3, & W_i \leq W_{block} \end{cases}, \quad (11)$$

where W_{normal} and W_{block} are the prescribed width thresholds separating the normal and narrow states and the narrow and blocked states, respectively, with $W_{\text{block}} < W_{\text{normal}}$. Both thresholds are measured in meters and are applied to the total effective width Wi defined in Equation (10). In the comprehensive navigation experiments, $W_{\text{block}} = 2.0$ m and $W_{\text{normal}} = 4.4$ m. The state labels $S_i = 1$, $S_i = 2$, and $S_i = 3$ denote normal, narrow, and blocked states, respectively. Normal regions retain nominal planning, narrow regions impose reduced velocity and stricter safety requirements, and blocked regions trigger waiting or global replanning.

The resulting feasible region provides both geometric constraints for candidate-trajectory screening and spatial-state information for subsequent target-selection and passage-decision mechanisms.

3.2. Look-Ahead Target Selection Within the Feasible Region

Conventional DWA commonly selects a point on the global reference path as the local target, which may cause delayed turning near sharp corners and underutilization of lateral free space when the centerline is temporarily affected by dynamic obstacles. To address this issue, a look-ahead target-selection strategy within the feasible region is developed by jointly considering path progress, upcoming curvature, available space within the feasible region, and predicted dynamic-obstacle risk.

As illustrated in Figure 2, the mechanism first determines a baseline look-ahead target, then adjusts it near significant corners or shifts it laterally when the predicted dynamic risk near the global reference path becomes high.

To prevent the local target from jumping backward when the robot moves around a corner, a nondecreasing path-progress index is maintained as

$$i_r(t) = \max\{i_r(t-1), i_n(t)\}, \quad (12)$$

where $i_n(t)$ denotes the index of the reference-path point nearest to the current robot position. Starting from $i_r(t)$, the baseline target g_0 is selected at a prescribed accumulated distance along the global reference path.

For a candidate path point q_i, the local turning angle is estimated from the path directions before and after the point. A point is regarded as a significant corner when the turning angle exceeds a prescribed threshold. As the robot approaches such a corner, the look-ahead distance is continuously adjusted according to the distance to the corner and the turning angle:

$$L_c = L_{\min} + (L_{\max} - L_{\min})\rho_d(1 - \kappa_\varphi \rho_\varphi), \quad (13)$$

where $L_{\min}$ and $L_{\max}$ are the minimum and maximum look-ahead distances, $\rho_d \in [0,1]$ represents the normalized distance to the upcoming corner, $\rho_\varphi \in [0,1]$ represents the normalized turning angle, and κ_φ controls the influence of path curvature. When the robot enters the predefined early-switching region, the baseline target is moved to a point on the post-corner path segment. This enables the robot to begin changing its motion direction before physically reaching the corner and reduces turning lag.

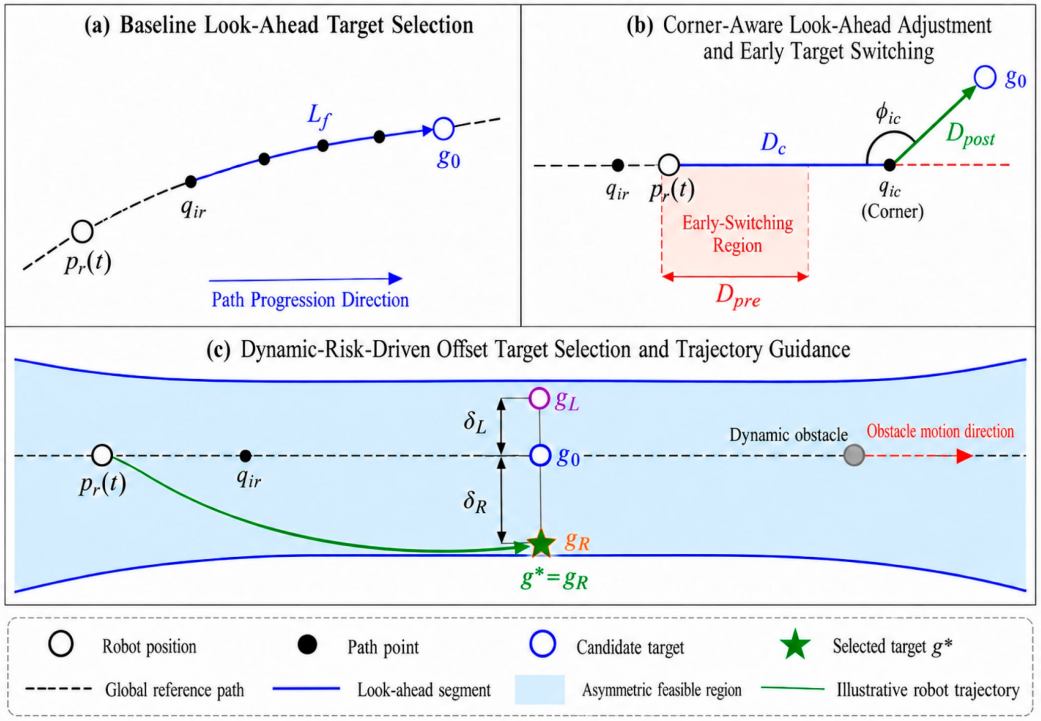


Figure 2. Schematic of look-ahead target selection within the feasible region: (**a**) baseline look-ahead target selection; (**b**) corner-aware look-ahead adjustment and early target switching; (**c**) dynamic-risk-driven offset target selection and trajectory guidance.

Under low dynamic risk, the baseline target g_0 remains on the global reference path. When the predicted dynamic safety around g_0 falls below a prescribed activation threshold, lateral target candidates are generated using the effective widths of the feasible region. The left and right candidates are defined as

$$g_L = g_0 + \delta_L \boldsymbol{n}_{i_f}, \tag{14}$$

$$g_R = g_0 - \delta_R \boldsymbol{n}_{i_f}, \tag{15}$$

where n_f denotes the normal vector to the global reference path at the baseline target, and δ_L and δ_R are constrained by the corresponding effective widths of the feasible region $\hat{W}_f^L$ and $\hat{W}_f^R$, while retaining a prescribed clearance from the boundaries of the feasible region.

Using the dynamic-obstacle predictions described in Section 3.3, each valid target candidate $g_m \in \{g_0, g_L, g_R\}$ is evaluated according to its predicted dynamic safety, remaining clearance from the feasible-region boundaries, and lateral deviation from the global reference path:

$$J_g(\boldsymbol{g}_m) = \alpha_g D_{dyn}(\boldsymbol{g}_m) + \beta_g M_{cor}(\boldsymbol{g}_m) - \gamma_g |e_m|, \tag{16}$$

where $D_{dyn}(g_m)$ denotes the predicted dynamic safety margin of the candidate target, $M_{cor}(g_m)$ denotes its remaining clearance from the feasible-region boundaries, and e_m denotes its lateral offset from the global reference path.

To avoid unnecessary switching among the center, left, and right targets, an offset target is adopted only when its score exceeds that of the baseline target on the global reference path by a prescribed threshold:

$$\boldsymbol{g}^* = \begin{cases} \boldsymbol{g}_b, & J_g(\boldsymbol{g}_b) > J_g(\boldsymbol{g}_0) + \Delta J_{\min} \\ \boldsymbol{g}_0, & \text{otherwise} \end{cases}, \tag{17}$$

where g_b is the highest-scoring valid candidate. Consequently, the robot follows the global reference path under low-risk conditions, adjusts its target in advance near significant

corners, and shifts toward the safer side of the feasible region when predicted dynamic obstacles affect the vicinity of the global reference path. In this way, the proposed strategy combines global-path tracking, corner anticipation, and lateral feasible-space utilization within a unified local-target-selection mechanism.

3.3. Dynamic Obstacle Motion Prediction and Safety Assessment

Conventional DWA mainly evaluates obstacles from their current states and may therefore respond late to future interactions. To improve anticipatory avoidance, robot and dynamic-obstacle motions are predicted over a common horizon and evaluated at synchronized future instants. Dynamic safety margin, approximate time-to-collision (ATTC), and terminal stopping safety are then used for candidate assessment.

Assume that the environment contains N_d dynamic obstacles. The state of the j-th obstacle is represented by its position, velocity, and equivalent radius as

$$O_j(t) = \left[o_j^T(t), v_j^T(t), r_j\right]^T, \tag{18}$$

where the position and velocity terms denote the obstacle position and velocity, respectively, and r_j denotes its equivalent radius.

Because the DWA prediction horizon is short, an ordinary dynamic obstacle is assumed to maintain constant velocity. Its future position is estimated as

$$\hat{\boldsymbol{o}}_j(t+\tau) = \boldsymbol{o}_j(t) + \boldsymbol{v}_j(t)\tau. \tag{19}$$

For obstacles moving reciprocally within a prescribed interval, let c_j denote the center of the reciprocal-motion range, d_j the corresponding unit motion direction, and A_j the maximum one-sided displacement. Their predicted positions are expressed as

$$\hat{o}_j(t+\tau) = c_j + \mathcal{R}_{A_j}\left[s_j(t) + v_j^s(t)\tau\right]d_j, \tag{20}$$

where the mirror-reflection operator confines the predicted displacement to $[-A_j, A_j]$ and reverses the motion direction at the interval endpoints.

For each candidate velocity pair (v,ω), the corresponding robot trajectory is predicted using the kinematic model described in Section 2. Let p_k denote the predicted robot position at the k-th prediction instant

$$\tau_k = k\Delta t, \tag{21}$$

where Δt is the prediction interval. The dynamic obstacles are predicted at the same time instants. The minimum dynamic safety margin associated with candidate (v,ω) is then defined as

$$d_{\min}^{\mathrm{dyn}}(v,\omega) = \min_{\substack{k=0,1,\ldots,N_p \\ j=1,2,\ldots,N_d}} \left[\left\|\mathbf{p}_k - \hat{\mathbf{o}}_j(t+\tau_k)\right\|_2 - r_j - r_r - d_s\right], \tag{22}$$

where N_p is the number of prediction steps, r_r is the equivalent robot radius, and d_s denotes the nominal additional clearance. Candidate trajectories are screened using the minimum predicted dynamic safety margin and the prescribed safety condition.

The same parameter d_s is used for static-obstacle inflation in Section 3.1 and remains fixed within each run. The risk-adaptive trajectory evaluation in Section 3.4 adjusts the evaluation weights online according to the composite risk measure; d_s itself is not updated online.

Because the minimum dynamic safety margin does not indicate the temporal urgency of a potential conflict, ATTC is defined as the first prediction instant at which the margin falls below a warning threshold d_{ttc}:

$$T_{\mathrm{ATTC}}(v,\omega) = \min\left\{\tau_k \middle| \min_j d_{j,k}^{\mathrm{dyn}} \le d_{\mathrm{ttc}}\right\}, \tag{23}$$

where $d_{j,k}^{dyn}$ is defined as

$$d_{j,k}^{dyn} = \left\| \boldsymbol{p}_k - \hat{\boldsymbol{o}}_j(t + \tau_k) \right\|_2 - r_r - r_j - d_s. \tag{24}$$

If no prediction instant satisfies this condition, T_{ATTC} is set to $+\infty$.

A smaller T_{ATTC} indicates a more urgent potential interaction, whereas $T_{\mathrm{ATTC}} = +\infty$ indicates that no significant near-term conflict is predicted within the current horizon. This quantity is referred to as an approximate time-to-collision because it is obtained from discretely predicted trajectories rather than from an analytical continuous collision-time solution.

A trajectory may remain safe within the prediction horizon but terminate at a position that becomes unsafe shortly afterward. Therefore, terminal stopping safety is checked over an additional stopping horizon T_s. Let $_{\mathrm{pe}}$ denote the terminal position of the candidate trajectory and T_p its prediction horizon. The minimum stopping safety margin is defined as

$$d_{\min}^{\mathrm{stop}} = \min_{\substack{j = 1, \ldots, N_d \\ 0 \le \tau \le T_s}} \left[\| \mathbf{p}_e - \hat{\mathbf{o}}_j(t + T_p + \tau) \|_2 - r_j - r_r - d_s - d_{\mathrm{stop}} \right], \tag{25}$$

where d_{stop} is an additional margin reserved for terminal stopping safety. The candidate terminal position is regarded as safe only when $d_{\min}^{\mathrm{stop}} \ge 0$. This check assumes that the robot remains at the predicted terminal position and evaluates its separation from moving obstacles over the additional horizon. It does not simulate an explicit braking trajectory and therefore does not establish safe stopping under finite deceleration or prediction error.

During normal tracking, candidate trajectories must pass both synchronized dynamic-collision screening and terminal-position occupancy screening. The resulting dynamic safety margin and T_{ATTC} are further used in the risk-adaptive trajectory evaluation described in Section 3.4. The mode-dependent changes to the screening conditions are described in Section 3.5.

3.4. Candidate Trajectory Generation and Risk-Adaptive Evaluation with Feasible-Region Constraints

The local state of the feasible region, local target, and predicted dynamic risk are incorporated into DWA candidate generation and evaluation. The admissible velocity range is first adjusted according to the local state of the feasible region, unsafe trajectories are removed by hard constraints, and the remaining candidates are evaluated using risk-adaptive safety, efficiency, and motion-stability criteria.

Let S_r denote the local state of the feasible region corresponding to the current robot position. A state-dependent velocity-scaling coefficient is defined as

$$\eta_s(S_r) = \begin{cases} \eta_{normal}, & S_r = 1 \\ \eta_{narrow}, & S_r = 2 \\ \eta_{block}, & S_r = 3 \end{cases}. \tag{26}$$

The locally allowable maximum linear velocity is then

$$v_{\max}^{\mathrm{loc}} = \eta_s(S_r) v_{\max}. \tag{27}$$

Combining this limit with the current velocity and acceleration constraints yields the locally admissible velocity window. Thus, the allowable speed is reduced according to the local width-based state of the feasible region to improve motion stability and response capability.

For each sampled velocity pair (v,ω), a short-horizon trajectory is generated using the robot kinematic model. During normal tracking, static-collision, synchronized dynamic-collision, terminal-position occupancy, and feasible-region boundary checks are applied as hard filters.

For a predicted trajectory point p_k, let q_{ik} denote the nearest point on the global reference path and n_{ik} the corresponding normal vector. Its signed lateral offset is

$$e_k = \left(\boldsymbol{p}_k - \boldsymbol{q}_{i_k}\right)^T \boldsymbol{n}_{i_k}. \tag{28}$$

With a reserved boundary clearance d_c, every predicted point must satisfy

$$-\hat{W}_{i_k}^R + d_c \le e_k \le \hat{W}_{i_k}^L - d_c. \tag{29}$$

Under this nominal boundary rule, any trajectory violating Equation (29) is discarded.

For trajectories satisfying all hard constraints, the clearances from the feasible-region boundaries, static obstacles, and dynamic obstacles are combined into the safety component

$$J_{safe} = \lambda_C C(v, \omega) + \lambda_O O(v, \omega) + \lambda_D D(v, \omega), \tag{30}$$

where $C(v, \omega)$ represents safety with respect to the feasible-region boundaries, $O(v, \omega)$ represents static-obstacle safety, and $D(v, \omega)$ represents dynamic-obstacle safety.

The trajectory-efficiency component jointly considers local-target approach, translational velocity, forward progress, and progress along the reference-path topology:

$$J_{eff} = \lambda_G G(v, \omega) + \lambda_V V(v, \omega) + \lambda_P P(v, \omega) + \lambda_T T(v, \omega), \tag{31}$$

where $G(v, \omega)$ evaluates the consistency of the candidate trajectory with the selected local target, $V(v, \omega)$ promotes motion efficiency, $P(v, \omega)$ rewards effective forward displacement, and $T(v, \omega)$ encourages progress along the reference path and suppresses backward rematching near corners. Motion stability is characterized by a smoothness term J_{smooth}, which penalizes abrupt changes in the selected angular velocity.

The resulting trajectory score is expressed as

$$J(v, \omega) = J_{safe} + J_{eff} + J_{smooth} - J_{pen}, \tag{32}$$

where J_{pen} is an additional penalty used to suppress unnecessary stopping under low-risk conditions, excessive velocity near the local target, and candidate motions inconsistent with the current passage state.

To adapt trajectory evaluation to changing environmental risk, a composite risk factor is constructed from static-obstacle proximity, dynamic-obstacle proximity, ATTC, local state of the feasible region, and proximity to the feasible-region boundaries:

$$R = a_1 R_{\text{sta}} + a_2 R_{\text{dyn}} + a_3 R_{\text{ATTC}} + a_4 R_{\text{cor}} + a_5 R_{\text{mar}}, \ \sum_{i=1}^{5} a_i = 1, \tag{33}$$

with $0 \le R \le 1$.

In the comprehensive navigation experiments, the coefficients $(a_1, a_2, a_3, a_4, a_5)$ in Equation (33) were $(0.24, 0.32, 0.22, 0.14, 0.08)$, corresponding to static-obstacle proximity, dynamic-obstacle proximity, ATTC, feasible-region state, and boundary proximity, respectively. The static and dynamic clearance scales were both 7.0 m, the ATTC scale was 2.5 s with a signed dynamic-margin warning threshold of 0.30 m, and the boundary-clearance scale was 1.0 m. The width-based risk component was 0.10, 0.45, or 0.85 for normal, narrow, or blocked states, respectively.

The dynamic-obstacle component is determined from the minimum dynamic safety margin, while the ATTC component increases as the predicted conflict becomes more urgent. The risk components associated with the feasible region reflect its local width-based state and the remaining distance to its boundaries. Therefore, R approaches a larger value when the robot operates close to static or dynamic obstacles, moves through a narrow part of the feasible region, or faces an imminent predicted interaction.

Let λ_k^0 denote the nominal coefficient of the k-th evaluation term. Its risk-dependent adjustment, excluding the additional scaling factors associated with the width-based state and navigation mode, is expressed as

$$\lambda_k = \lambda_k^0(1 + \mu_k R), \tag{34}$$

where μ_k is a signed coefficient controlling the risk-dependent adjustment: positive values increase the corresponding coefficient as R rises, negative values decrease it, and zero leaves it unchanged. For a fixed width-based state and navigation mode, increasing R increases the static-obstacle, dynamic-obstacle, and motion-smoothness coefficients, while reducing the goal-approach, feasible-region boundary, velocity, and progress coefficients. The topology coefficient remains unchanged. Additional scaling factors associated with the width-based state and navigation mode are applied separately. For the goal approach, feasible-region boundary, static-obstacle safety, dynamic-obstacle safety, velocity, forward progress, motion smoothness, and reference-path topology terms, respectively, the nominal coefficients were $\lambda^0 = (1.15, 0.65, 1.20, 1.45, 0.85, 1.35, 0.18, 0.65)$, and the risk-adaptation coefficients were $\mu = (-0.20, -0.50, 1.20, 1.60, -0.55, -0.35, 0.45, 0)$.

After all valid candidate trajectories have been evaluated, the optimal velocity command is selected as

$$(v^*, \omega^*) = \arg \max_{(v,\omega)\in U_{\text{valid}}} J(v, \omega), \tag{35}$$

where V_{valid} denotes the candidate velocity set that satisfies all hard safety constraints.

Through this procedure, the proposed method combines velocity regulation based on the local state of the feasible region, hard geometric and dynamic safety screening, and risk-adaptive trajectory evaluation. The robot therefore favors efficient motion in open and low-risk regions while adopting more conservative and stable behavior when dynamic interactions or spatial constraints become significant.

3.5. Dynamic Passage and Global Replanning Strategy

Short-horizon local planning may be insufficient when a dynamic obstacle repeatedly or persistently occupies a constrained passage. To distinguish temporary from persistent blockage, a passage and navigation-recovery mechanism is introduced to coordinate passage-entry admission, safe-passage-window detection, short-term constrained traversal, waiting, and global replanning. As illustrated in Figure 3, the robot either waits before the affected region, traverses it when a complete safe passage window is available, or triggers replanning when the original route remains persistently unavailable.

To determine whether the robot should enter a dynamically affected passage, the future occupancy of the reference path is first evaluated using the dynamic-obstacle predictions from Section 3.3. A reference-path point q_i is regarded as dynamically occupied at prediction time τ when

$$\left\| \boldsymbol{q}_i - \hat{\boldsymbol{o}}_j(t + \tau) \right\|_2 - r_j - r_r - d_s < d_{\text{occ}}, \tag{36}$$

where d_{occ} is the prescribed distance threshold for identifying dynamic occupancy of a reference-path point. The first and last predicted occupancy times define the dynamic-obstacle occupancy interval.

Let D_b denote the path distance from the current robot position to the affected region and D_e the length of the region that must be traversed. Using an estimated traversal velocity v_e, the predicted entry and exit times are approximated as

$$\tau_{\text{in}} = \frac{D_b}{v_e}, \ \tau_{\text{out}} = \tau_{\text{in}} + \frac{D_e}{v_e} + T_{\text{buf}}, \tag{37}$$

where T_{buf} is an additional passage-safety buffer. In the comprehensive experiments, the entrance gate used $d_{occ} = 0.25$ m, $v_e = 1.75$ m/s, and $T_{\text{buf}} = 0.55$ s. The gate examined

reference-path points within 13.0 m of the robot, with a configured prediction horizon of 4.2 s and a sampling interval of 0.25 s. When entry was blocked, the holding point was selected using a nominal upstream offset of 2.0 m, with a stopping tolerance of 0.45 m.

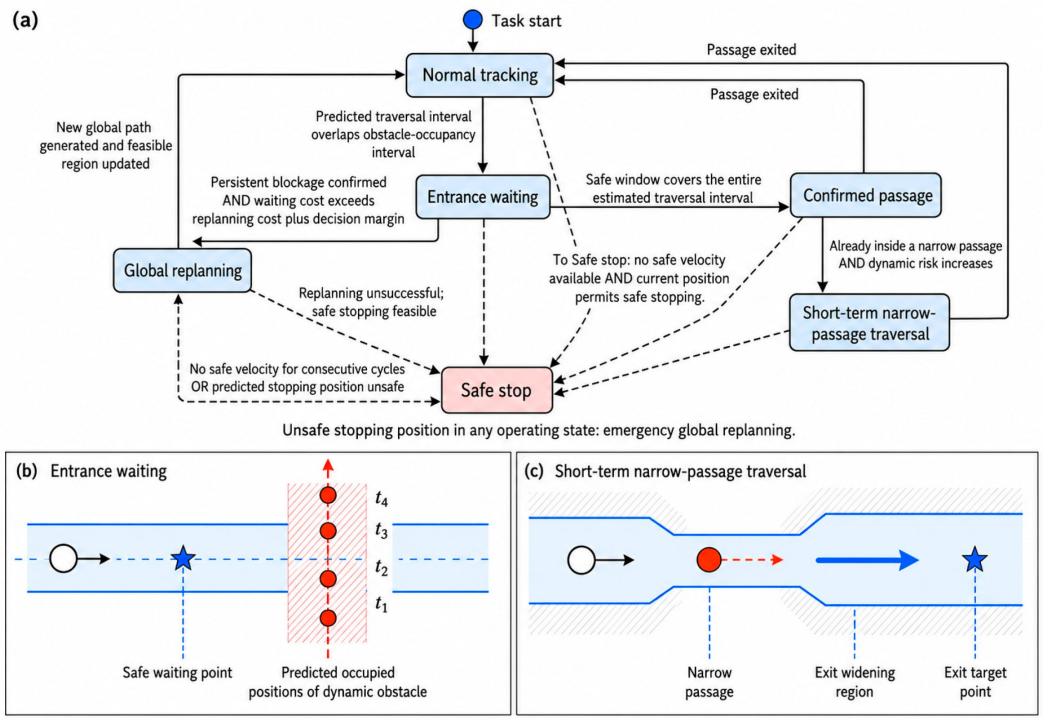


Figure 3. Schematic of the dynamic passage and navigation-recovery mechanism: (**a**) state-transition diagram; (**b**) entrance waiting under predicted occupancy of the affected reference-path segment; (**c**) short-term narrow-passage traversal and exit-target guidance. Note: In (**a**), solid arrows indicate the main state transitions, and dashed arrows indicate transitions to or from the safe-stop state. In (**b**,**c**), black arrows indicate the robot's direction of motion, and red arrows indicate the dynamic obstacle's direction of motion. The thick blue arrow in (**c**) shows the direction toward the exit target.

If the estimated traversal interval $[\tau_{\text{in}}, \tau_{\text{out}}]$ overlaps the predicted obstacle-occupancy interval, the robot is not permitted to enter the affected region. Instead, a safe waiting point is selected upstream of the blockage location, and the local target is temporarily replaced by this waiting point. This prevents the robot from stopping inside the future swept region of the dynamic obstacle.

While waiting, the algorithm continuously evaluates whether a complete safe passage window becomes available. Using the affected-region length D_e and the estimated traversal velocity v_e defined above, the required passage time is

$$T_{\text{pass}} = \frac{D_e}{v_e} + T_{\text{buf}}. \tag{38}$$

The clearance criterion for a complete traversal is expressed as

$$\|q(\tau) - \hat{o}_j(t+\tau)\|_2 - r_r - r_j - d_s \geq d_{\text{pass}},\ 0 \leq \tau \leq T_{\text{pass}}, \tag{39}$$

for all relevant dynamic obstacles, where d_{pass} denotes the additional safety margin required for passage confirmation.

In the comprehensive experiments, passage confirmation used a signed clearance threshold of 0.18 m and a forecast capped at 5.0 s, sampled at 0.20 s intervals. The assumed traversal speed and time buffer were 1.85 m/s and 0.45 s, respectively. In confirmed-passage mode, the local target was moved forward and the terminal-position occupancy horizon was shortened to 0.50 s.

If the robot has already entered a narrow passage when dynamic pressure increases, returning to the entrance waiting point may be undesirable because sufficient stopping space may no longer be available. In this case, a short-term narrow-passage traversal state is activated. The planner searches forward for a wider and dynamically safer path region and uses it as an exit target. In the comprehensive implementation, this mode uses a shortened prediction horizon, disables terminal-position occupancy screening, and permits a limited deviation from the feasible-region boundary. Sampled static- and dynamic-obstacle clearance checks remain active.

When entry remains blocked, the planner further determines whether continued waiting is preferable to global replanning. The estimated waiting cost and replanning cost are expressed as

$$J_{\text{wait}} = T_w + \frac{D_{\text{rem}}}{v_e} + \lambda_r T_b, \tag{40}$$

$$J_{\text{replan}} = T_c + \eta_d \frac{D_{\text{rem}}}{v_e}, \tag{41}$$

where T_w is the estimated remaining waiting time, T_b represents the predicted blockage duration, D_{rem} is the remaining path length to the goal, T_c is the equivalent cost of global path recomputation and motion-state transition, and λ_r and η_d are the corresponding correction coefficients.

In the comprehensive experiments, the cost parameters were $\lambda_r = 0.60$, $T_c = 0.50$ s, and $\eta_d = 1.12$. During entrance waiting, the gate could request replanning within 0.95 m of the holding point if the estimated waiting time exceeded 2.8 s or if the waiting cost exceeded the replanning cost by more than 0.30 s. Emergency replanning could be requested after two consecutive control cycles without an admissible candidate when a bottleneck was detected or the path ahead was blocked.

When no candidate velocity satisfies the safety constraints, the algorithm checks whether remaining at the current position satisfies the predicted clearance conditions. If so, the command $v^* = 0$, $\omega^* = 0$ is temporarily applied and the environment is reevaluated at the next control cycle. If the no-safe-velocity condition persists or the current stopping position becomes threatened by predicted dynamic-obstacle motion, emergency replanning is triggered.

During replanning, the predicted future motion of each dynamic obstacle is converted into a temporary swept region by placing inflated obstacle regions along its predicted trajectory. The replanning map is therefore constructed as

$$O_{replan} = O_{static}^{inf} \cup O_{dynamic}^{temp}, \tag{42}$$

where O_{static}^{inf} denotes the inflated static-obstacle set and $O_{dynamic}^{temp}$ denotes the temporary swept regions generated from predicted dynamic-obstacle motion.

The global planner is then invoked using the current robot position as the new starting point and the original navigation goal as the destination. After a feasible route is obtained, the reference path is simplified and validated, the feasible region described in Section 3.1 is reconstructed, and the path-progress information is reset according to the updated route. If replanning fails, a zero-velocity command is maintained, and the environment is reevaluated in subsequent control cycles.

Through this mechanism, temporary and persistent blockage are handled differently. Temporary blockage is addressed through safe entrance waiting and complete passage-window detection, whereas persistent blockage is handled through cost- and duration-based global replanning. Consequently, the robot can avoid premature passage entry, exploit temporary safe traversal opportunities, and recover task progress when the original reference path becomes persistently unavailable.

3.6. Overall Navigation Workflow

The overall navigation workflow of the proposed method is illustrated in Figure 4. The global reference path and feasible region are first constructed to provide geometric guidance for local planning. During navigation, the target-selection mechanism within the feasible region determines the local target, while candidate robot trajectories and dynamic-obstacle motions are predicted over a common horizon. Candidate trajectories are screened using the checks enabled for the active navigation mode, and the retained trajectories are evaluated using the risk-adaptive objective function. According to the predicted passage state, the robot either continues normal tracking, waits for a safe passage opportunity, traverses a dynamically constrained passage, or triggers global replanning when the original route becomes persistently unavailable. After replanning, the reference path and feasible region are updated, and local navigation resumes along the new route. This closed-loop process coordinates global guidance, local trajectory optimization, dynamic-risk assessment, and navigation recovery within a unified planning process.

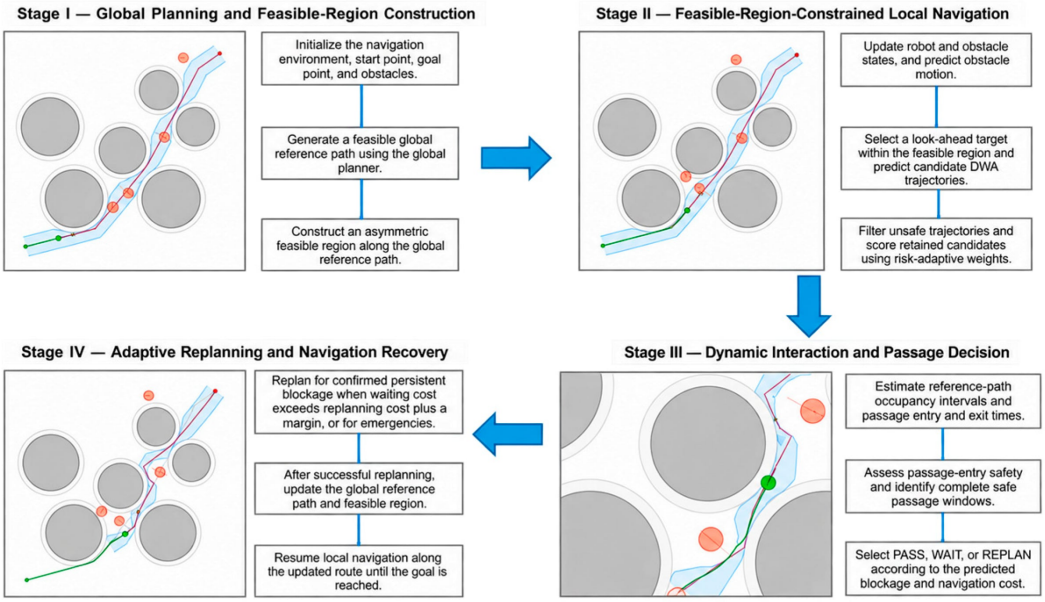


Figure 4. Overall navigation workflow of the proposed method. Note: Gray circles represent static obstacles, orange circles represent dynamic obstacles, and the light-blue shaded bands represent the asymmetric feasible regions. The large blue arrows indicate the sequence of the four navigation stages.

4. Experiments

4.1. Experimental Environment and Parameter Settings

All simulation experiments were conducted in MATLAB R2023a on a computer equipped with an AMD Ryzen 7 4800H processor and 32 GB RAM. The experimental maps were defined within a 100 × 100 workspace. A differential-drive mobile robot with a control period of 0.1 s was adopted. The equivalent robot radius was 1.2 m, and an additional safety clearance of 0.8 m was used for obstacle inflation and collision checking.

In a fixed-map sensitivity test, ten matched trials were conducted at each nominal additional clearance of 0.2, 0.4, 0.6, and 0.8 m using the same initial reference path. The numbers of successful collision-free runs were 10, 8, 7, and 9, respectively. Across the nine pairs in which both the 0.2 and 0.8 m runs succeeded, the 0.8 m setting increased the mean minimum sampled dynamic physical clearance by 0.693 m and the mean arrival time by 11.92 s. The nominal setting of 0.8 m was retained to favor greater obstacle separation under the tested conditions.

Five experimental groups were designed to evaluate different components of the proposed method: feasible-region construction and offset-target selection, dynamic-obstacle prediction and risk-adaptive evaluation, corner-aware look-ahead target selection, dynamic passage and global replanning, and comprehensive navigation performance. Because the five experimental groups have different validation objectives, the maximum velocity, prediction horizon, and look-ahead distance were configured accordingly, as summarized in Table 1.

Table 1. Main parameter settings for the five experimental scenarios.

Experimental Scenario	Maximum Linear Velocity (m/s)	Maximum Angular Velocity (°/s)	Prediction Horizon (s)	Look-Ahead Distance (m)
Dynamic-prediction Feasible-region	2.60	65	3.50	8.00
Corner-aware look-ahead	2.80	80	2.80	8.00
Narrow-passage	3.30	80	2.80	2.80–8.00
Comprehensive navigation	2.80	75	2.60	7.00
	3.00	90	2.80	3.20–7.00

The mechanism-validation comparisons were conducted under matched scenario conditions. Within each scenario, all compared methods used the same initial robot state, reference path, obstacle configuration, dynamic-obstacle motion parameters, and termination conditions. The comparisons examined the respective mechanism implementations, with differences in speed selection and path-progress handling considered when interpreting the results. Unrelated auxiliary functions were disabled where necessary.

For the comprehensive experiment, Traditional DWA, Hahn-DWA, Grasshopper Optimization Algorithm-based DWA (GOA-DWA) [31], Predictive DWA, TAP, and the proposed method were compared under the same map, robot model, initial reference path, dynamic-obstacle motion rules, and basic motion constraints. To avoid unfair differences caused solely by route-recovery capability, all baseline methods were provided with the same basic RRT-based recovery mechanism using the current obstacle configuration only. In contrast, the proposed method retained its complete set of mechanisms for feasible-region construction, predictive risk assessment, passage decisions, and replanning.

Thirty trials were conducted for each method in the comprehensive scenario using the same randomized dynamic-obstacle initial phases. Navigation success was calculated over all trials, whereas continuous indicators were computed over successful collision-free runs and reported as mean ± standard deviation. The evaluation considered navigation success, dynamic safety margin, arrival time, path length, tracking error, waiting or replanning behavior, and computation time per control cycle. The five-component risk measure in Equation (33) is used in the comprehensive navigation implementation. The isolated dynamic-crossing experiment in Section 4.2.2 uses a scenario-specific risk configuration based on the predicted dynamic safety margin and ATTC.

Additional paired simulations were conducted within fixed scenarios to examine mechanism repeatability, risk and weight evolution, and sensitivity to state-switching thresholds, obstacle-motion deviations, and execution limits. A separate comparison at closely matched executed speeds assessed the influence of speed on tracking performance.

4.2. Simulation Results and Analysis Under Different Scenarios

4.2.1. Effect of the Asymmetric Feasible Region

To evaluate the ability of the asymmetric feasible region to represent asymmetric feasible space and support lateral obstacle avoidance, a dynamic crossing scenario was constructed in which a moving obstacle crossed the global reference path near x = 42. Static obstacles were arranged mainly above the path, producing a narrower feasible region on the upper side and a wider region below. Four methods were compared: Centerline-guided DWA, Fixed-width-region DWA, Adaptive-region DWA, and Adaptive-region–offset-target DWA. Passage-entry waiting and global replanning were disabled to isolate the effect of the feasible-region and offset-target mechanisms.

As shown in Figure 5, the feasible region is narrower near the upper static obstacles while retaining greater free space on the lower side, thereby explicitly representing the asymmetric traversable region around the reference path. The trajectories of Centerline-guided DWA, Fixed-width-region DWA, and Adaptive-region DWA remain close to the path centerline, indicating that the feasible region alone mainly acts as a geometric feasibility constraint and does not necessarily induce lateral avoidance.

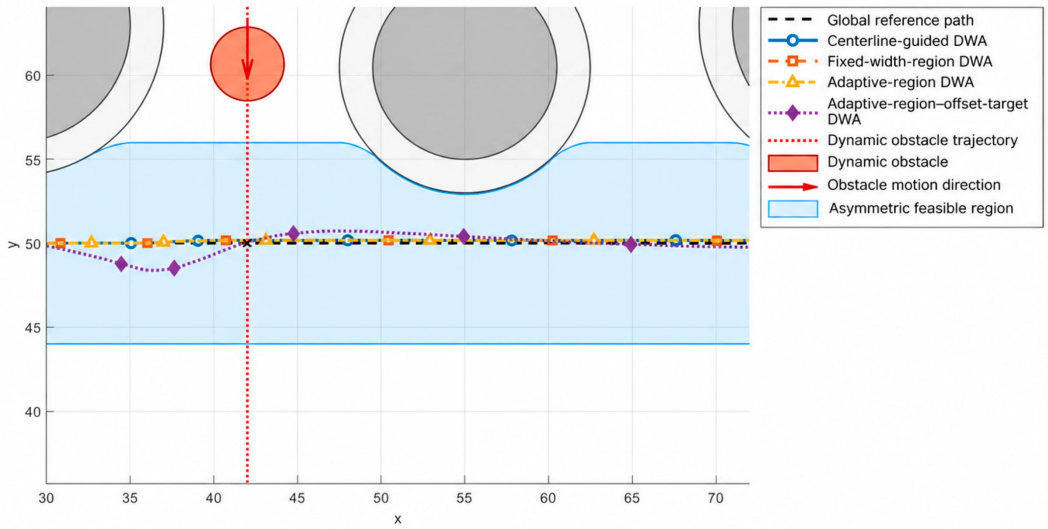


Figure 5. Local trajectories of different methods in an environment with an asymmetric feasible region. Note: The gray filled circles represent static obstacles, and their outer outlines mark the boundaries of the inflated obstacles. The orange circle represents the dynamic obstacle.

In contrast, Adaptive-region–offset-target DWA shifts gradually toward the less constrained side before the dynamic obstacle reaches the interaction region and subsequently returns toward the reference path. The trajectory remains inside the feasible region throughout the maneuver, demonstrating that the offset-target mechanism converts the available lateral space into explicit avoidance guidance.

Figure 6 further shows that only Adaptive-region–offset-target DWA produces a nonzero lateral target offset during the interaction. The robot responds smoothly, with a maximum lateral displacement of approximately 1.6 m, while maintaining a safe separation from the moving obstacle. These results indicate that feasible-region construction and offset-target selection play complementary roles: the former defines the locally feasible region, whereas the latter enables the robot to actively exploit that region when the global-path centerline is temporarily affected by a dynamic obstacle.

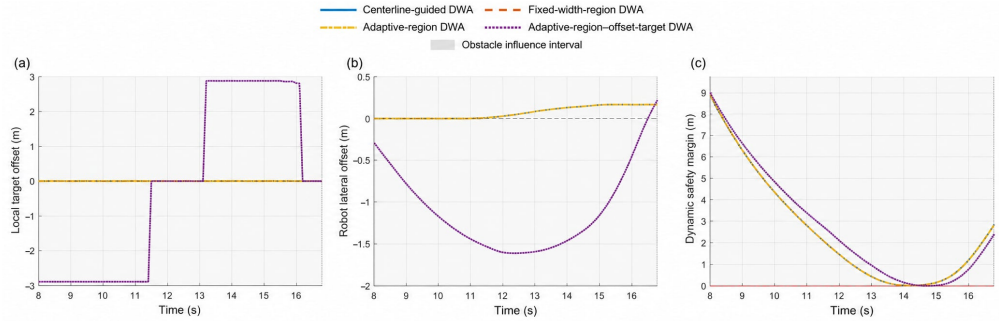


Figure 6. Evolution of target selection, robot lateral response, and dynamic safety during obstacle interaction. (**a**) Local target offset. (**b**) Robot lateral offset. (**c**) Dynamic safety margin. The gray shading indicates the obstacle influence interval. The gray dashed horizontal reference line marks zero offset.

4.2.2. Effect of Dynamic-Obstacle Prediction and Risk-Adaptive Evaluation

To evaluate the effects of synchronized dynamic-obstacle prediction and risk-adaptive trajectory evaluation, an open-area crossing scenario was constructed in which a moving obstacle crossed the robot's reference path. Four methods were compared: Current-position DWA, Synchronized-prediction DWA, Prediction-and-TTC DWA, and the proposed method. The compared methods differed only in the treatment of future obstacle motion, temporal conflict information, and risk-adaptive weighting.

As shown in Figure 7, Current-position DWA reacts relatively late and maintains safety mainly by delaying passage through the conflict region. Synchronized-prediction DWA and Prediction-and-TTC DWA respond earlier, but both approach the prescribed dynamic safety boundary closely and remain in the high-risk region for approximately 3.2 s. In contrast, the proposed method begins a moderate lateral adjustment before entering the main conflict region and maintains the prescribed safety threshold throughout the interaction.

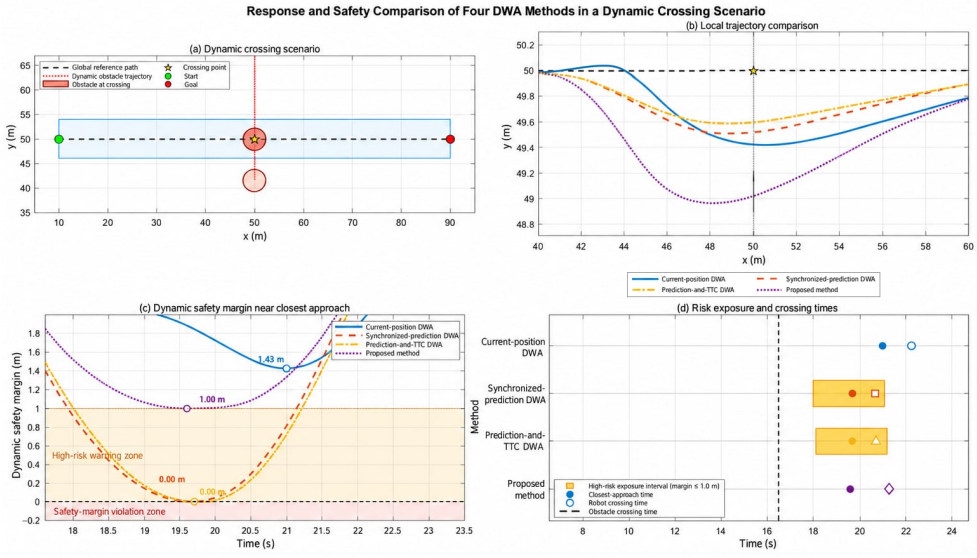


Figure 7. Dynamic-obstacle response and safety comparison of four DWA methods in the crossing scenario: (**a**) dynamic crossing scenario; (**b**) local trajectory comparison; (**c**) safety clearance near closest approach; (**d**) risk exposure and crossing times. Note: The stars in panels (**a**,**b**) mark the crossing point. The open circles in panel (**c**) mark the minimum dynamic safety margins. In panel (**d**), filled markers indicate the times of closest approach, open markers indicate the robot crossing times, and the vertical dashed line marks the obstacle crossing time.

The quantitative results are listed in Table 2.

Table 2. Performance comparison of dynamic-obstacle prediction and risk-adaptive evaluation methods.

Method	Arrival Time (s)	Path Length (m)	Minimum Dynamic Safety Margin (m)	High-Risk Exposure Time (s)	Mean Computation Time per Cycle (ms)
Current-position DWA	43.0	78.570	1.4284	0	4.973
Synchronized-prediction DWA	41.4	78.530	0.0024	3.2	5.143
Prediction-and-TTC DWA	41.5	78.540	0.00003	3.2	5.132
Proposed method	42.2	78.640	1.0011	0	5.094

All four methods complete the task without physical collision, and their path lengths differ by less than 0.11 m. Synchronized-prediction DWA and Prediction-and-TTC DWA achieve the shortest arrival times, but their minimum dynamic safety margins are close to zero. The proposed method reaches the goal in 42.2 s with a minimum dynamic safety margin of 1.0011 m and zero high-risk exposure time, providing a better balance between traversal efficiency and dynamic safety. The mean computation time was 5.094 ms per cycle, below the 0.1 s control period, indicating a low average computational cost in this scenario.

These results show that synchronized prediction improves anticipatory interaction assessment, whereas temporal conflict information alone is insufficient to guarantee an adequate safety margin. By further adapting the trajectory-evaluation weights according to local risk, the proposed method achieves a more balanced trade-off between safety and efficiency.

To assess sensitivity to obstacle-motion mismatch, the actual obstacle speed was modulated sinusoidally by 0%, 10%, or 20% with a 6 s period in the crossing scenario. In a separate near-reversal scenario, the actual upper reversal boundary was shifted inward by 0, 0.5, or 1.0 m. Four methods used ten matched trials per scenario and setting, yielding 240 runs, with exact current obstacle states and nominal motion parameters supplied to the planner. At the 3.5 s forecast horizon, mean position RMSEs were 0.3023 and 0.6003 m for the two nonzero speed modulations, and 0.3598 and 0.6799 m for the two nonzero boundary shifts. All runs completed without a sampled physical collision or nominal-clearance violation. For the proposed method, the 1.0 m shift reduced the mean minimum sampled dynamic physical clearance from 2.4308 to 1.9732 m.

4.2.3. Effect of the Corner-Aware Look-Ahead Target

To evaluate the effect of the corner-aware look-ahead mechanism on path-tracking performance, a continuous S-shaped reference path with multiple consecutive turns was constructed. Four methods were compared: Fixed-look-ahead DWA, Speed-adaptive-look-ahead DWA, Threshold-based corner-look-ahead DWA, and the proposed Corner-aware-look-ahead DWA. All methods used the same robot model and reference path, with the target-selection, path-progress, and speed-selection rules of their respective implementations.

As shown in Figure 8, Fixed-look-ahead DWA exhibits noticeable outward deviation near corners because the target direction changes relatively late. Speed-adaptive and threshold-based strategies reduce part of this deviation, but the latter may still produce large instantaneous errors because of discrete look-ahead switching. In contrast, the proposed method continuously adjusts the look-ahead distance according to the distance to the upcoming corner and the turning angle, while switching the target toward the post-corner segment in advance. This results in smoother corner entry and smaller trajectory deviation.

The quantitative results are listed in Table 3.

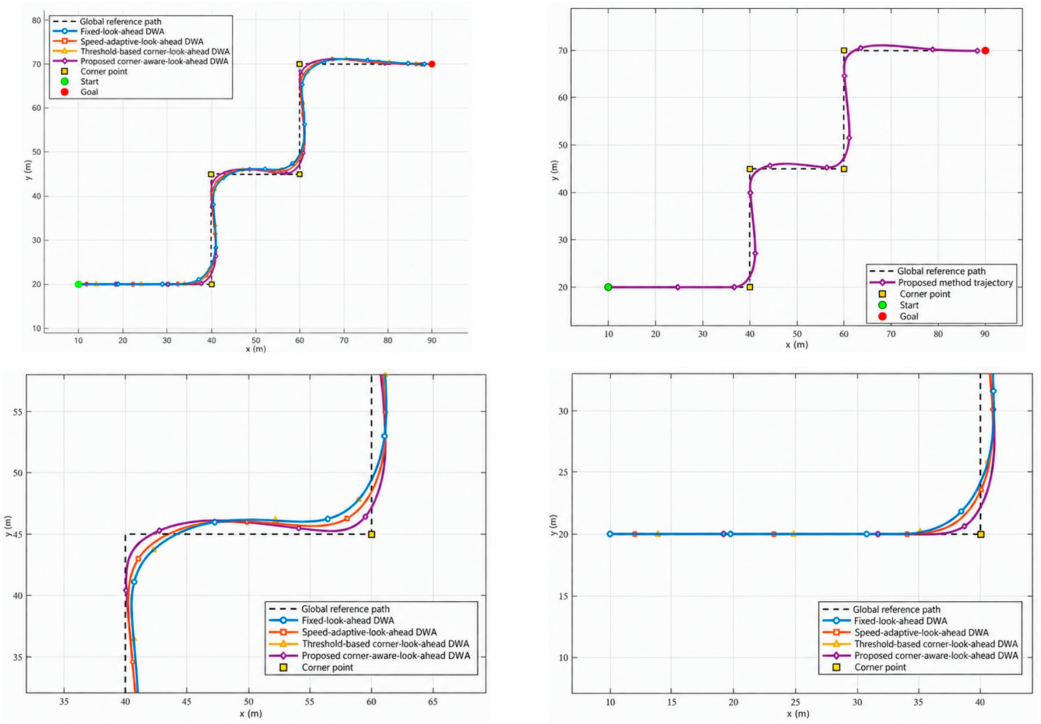


Figure 8. Path-tracking performance of different look-ahead target-selection methods on the S-shaped reference path.

Table 3. Experimental results for different look-ahead target-selection methods.

Method	Arrival Time (s)	Mean Tracking Error (m)	Maximum Tracking Error (m)	Mean Maximum Corner Error (m)
Fixed-look-ahead DWA	50.9	0.670	1.940	1.797
Speed-adaptive-look-ahead DWA	57.1	0.564	1.535	1.453
Threshold-based corner-look-ahead DWA	56.9	0.549	2.060	1.659
Proposed corner-aware-look-ahead DWA	67.5	0.464	1.122	1.094

The proposed method achieves the lowest mean tracking error, maximum tracking error, and mean maximum corner error, with values of 0.464 m, 1.122 m, and 1.094 m, respectively. Compared with Fixed-look-ahead DWA, these errors are reduced by 30.7%, 42.2%, and 39.1%, respectively. However, its arrival time increases to 67.5 s, indicating that the improvement in tracking accuracy is achieved at the cost of reduced traversal efficiency. The speed-adaptive and proposed corner-aware implementations were each tested at their naturally selected speeds and under a common speed plan with a cruise speed of 2.1 m/s. The four configurations used the same ten initial perturbations, yielding 40 runs. Under the common plan, their mean executed speeds were 2.0726 and 2.0730 m/s, their mean arrival times were 58.97 and 58.96 s, and their mean tracking errors were 0.5841 and 0.5103 m, respectively. The proposed implementation had lower tracking error in all ten matched-speed pairs, with a 12.64% reduction in mean error relative to the speed-adaptive implementation. Each implementation retained its own path-progress rules.

Overall, the results confirm that combining continuous corner-aware look-ahead adjustment with early target switching is more effective than fixed or discretely switched look-ahead strategies in reducing turning lag and improving path-tracking accuracy on paths with consecutive corners.

4.2.4. Effect of Dynamic Passage and Global Replanning

To evaluate the dynamic passage and navigation-recovery mechanisms, two narrow-passage scenarios were considered: short-term periodic blockage and persistent occupancy. Three strategies were compared: Predictive DWA, a Conservative waiting strategy, and the proposed complete strategy. All methods used the same robot, obstacle, and termination conditions, while only the passage-state and replanning mechanisms were changed.

In the short-term periodic-blockage scenario, a dynamic obstacle repeatedly crosses the passage, creating temporary traversal opportunities. Predictive DWA enters the passage whenever a locally safe velocity is available, whereas the Conservative waiting strategy remains near the entrance until the blockage disappears. The proposed strategy instead determines whether a complete safe passage window is available before allowing entry.

As shown in Figure 9, the proposed method detects a safe passage window at 10.4 s, enters the passage at 12.4 s, and exits at 22.1 s, resulting in a passage traversal time of 9.7 s. Predictive DWA requires 12.9 s to traverse the same constrained region, whereas the Conservative waiting strategy fails to reach the goal within the 72.0 s termination time. These results indicate that complete passage-window verification improves motion continuity while avoiding unnecessarily conservative waiting.

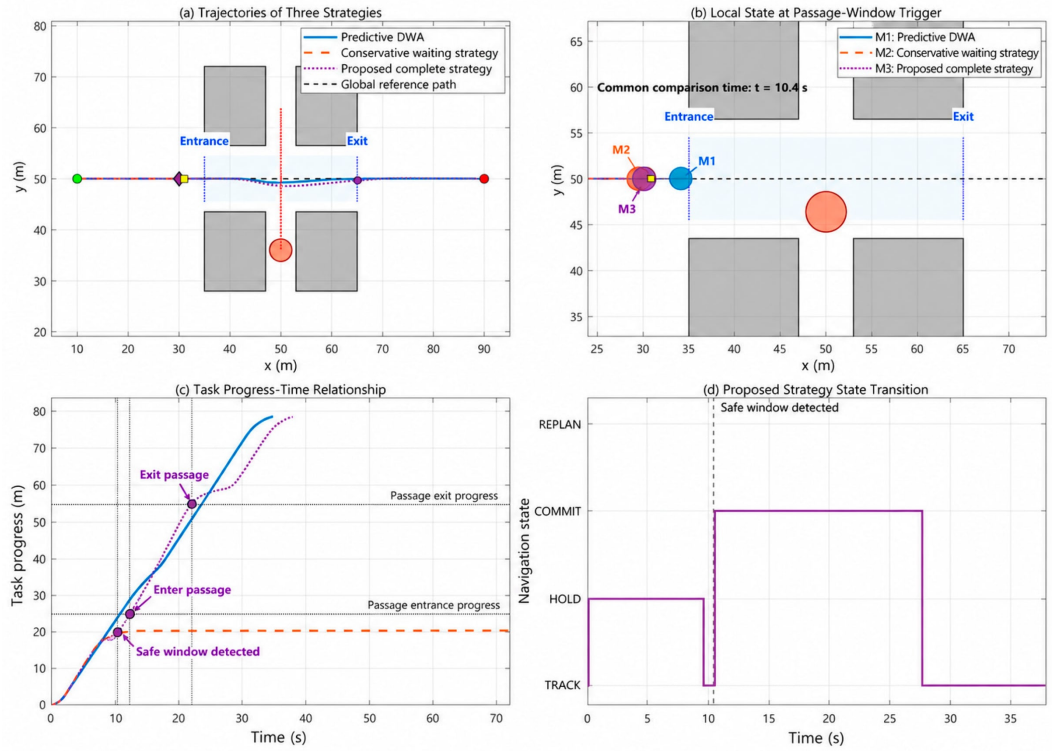


Figure 9. Comparison of passage-handling strategies and the proposed strategy state transition: (**a**) trajectories of three strategies; (**b**) local state at passage-window trigger; (**c**) task progress-time relationship; (**d**) proposed strategy state transition. Note: The yellow square marks the entrance holding point, the gray blocks represent static obstacles, and the orange obstacle circle represents the dynamic obstacle. In (**b**), the colored circles labeled M1, M2, and M3 show the robot positions under the three strategies at 10.4 s.

In the persistent-occupancy scenario, the dynamic obstacle remains near the original reference path for an extended period. Under this condition, local velocity selection or continued waiting alone cannot restore task progress. The proposed method therefore compares the estimated costs of waiting and replanning while also requiring the blockage to persist for a prescribed confirmation period.

As shown in Figure 10, the proposed strategy triggers global replanning at 15.0 s after the persistent-blockage condition is confirmed. At that instant, the estimated waiting cost is 75.3 s, whereas the replanning cost plus the decision margin is 31.6 s. The robot subsequently follows a detour and reaches the goal at 54.2 s. In contrast, Predictive DWA becomes stalled inside the passage, and the Conservative waiting strategy remains near the entrance; neither completes the task before the common 92.0 s termination time.

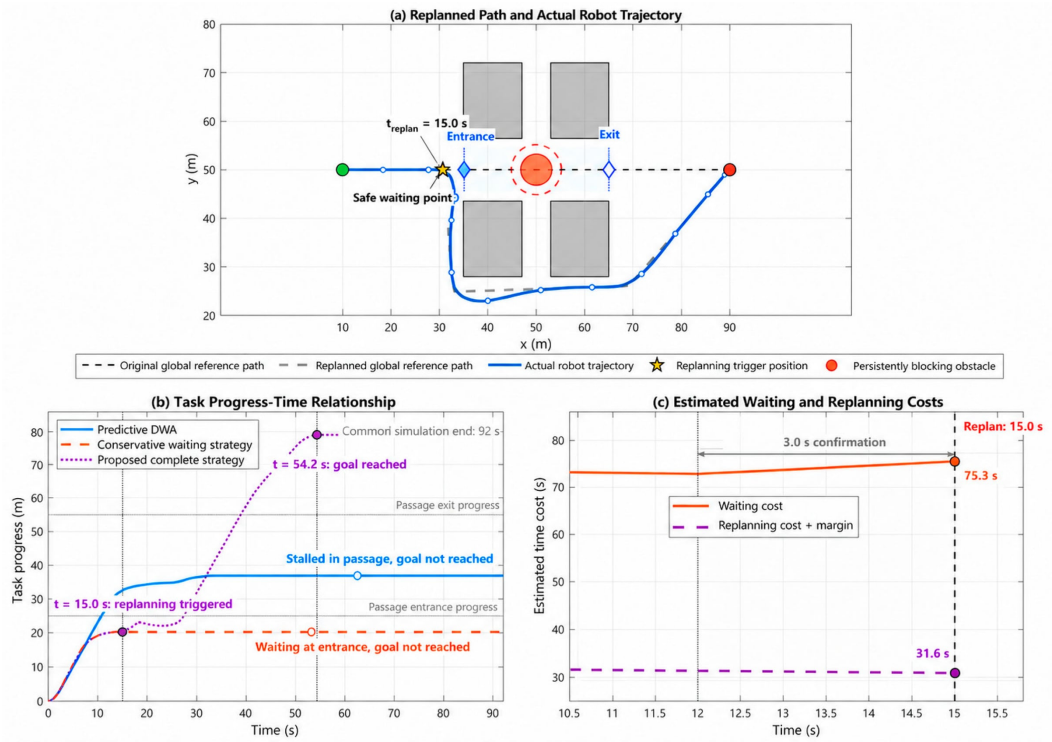


Figure 10. Navigation behavior under persistent reference-path blockage: (**a**) replanned path and actual robot trajectory; (**b**) task progress-time relationship; (**c**) estimated waiting and replanning costs. Note: In (**a**), the yellow star marks the replanning trigger position. The filled and open blue diamonds mark the passage entrance and exit, respectively. The red dashed circle marks the inflated boundary of the blocking obstacle.

The main quantitative results are summarized in Table 4. For unsuccessful runs, the reported path length denotes the distance traveled before the termination time.

The two scenarios demonstrate that temporary and persistent blockage require different navigation decisions. Safe-passage-window detection enables the robot to exploit temporary traversal opportunities, whereas the waiting–replanning mechanism restores task progress when the original route remains persistently unavailable. No physical collision was observed in either of the two tested scenarios.

The passage-decision tests used ten matched trials per setting with a fixed nominal additional clearance of 0.8 m. The nominal passage-admission threshold was 0.25 m in signed dynamic margin. In the nominal persistent-blockage test, replanning required at least 3.0 s of continuous blockage confirmation while the robot was within 0.90 m of the holding point as well as a waiting cost more than 0.40 s above the estimated replanning cost. In temporary blockage, entry-clearance thresholds of 0.20, 0.25, and 0.30 m yielded mean passage-commitment times of 10.13, 10.21, and 10.30 s, respectively. In persistent blockage, confirmation durations of 2.4, 3.0, and 3.6 s yielded mean replanning-trigger times of 14.42, 15.01, and 15.71 s, respectively. All 60 runs reached the goal without a recorded physical collision or nominal-clearance violation.

A separate execution-limit test used ten matched trials per model in each of the two blockage scenarios, with two instantaneous-execution settings and three finite braking rates of 0.55, 1.10, and 2.20 m/s^2. All 60 finite-rate runs reached the goal, while all 20 instantaneous-execution runs in persistent blockage timed out. Across the 100 runs, no physical collision was recorded at the 0.01 s sampling interval. The inflated static-obstacle check flagged contact or entry in all 30 finite-rate persistent-blockage runs.

Table 4. Experimental results for dynamic-passage and replanning strategies.

Scenario	Method	Goal Reached	Arrival Time (s)	Path Length (m)	Entry Waiting Time (s)
Short-term periodic blockage	Predictive DWA	Yes	34.8	78.598	0
Short-term periodic blockage	Conservative waiting strategy	No	—	20.618	57.2
Short-term periodic blockage	Proposed complete strategy	Yes	37.9	78.683	0
Persistent occupancy	Predictive DWA	No	—	39.619	0
Persistent occupancy	Conservative waiting strategy	No	—	20.509	77.9
Persistent occupancy	Proposed complete strategy	Yes	54.2	113.244	0.9

4.2.5. Comprehensive Navigation Performance in a Dynamic Environment

To evaluate the overall navigation capability of the proposed method, a comprehensive dynamic environment containing multiple static obstacles, moving obstacles, consecutive turns, and locally constrained regions was constructed. Six methods were compared: Traditional DWA, Hahn-DWA, GOA-DWA, Predictive DWA, TAP, and the proposed method. All methods used the same robot model, initial global reference path, dynamic-obstacle motion rules, and basic motion constraints. In addition, the baseline methods were provided with the same basic RRT-based recovery mechanism to avoid failures caused solely by the absence of route recovery.

Thirty trials were conducted for each method using the same randomized dynamic-obstacle initial phases. Navigation success was calculated over all trials, while the continuous performance indicators were reported as mean ± standard deviation over successful collision-free runs.

As shown in Figure 11, the baseline methods exhibit different levels of anticipatory obstacle avoidance and local recovery capability. Traditional DWA remains largely reactive, whereas Hahn-DWA, GOA-DWA, Predictive DWA, and TAP improve local decision-making through prediction or enhanced velocity evaluation. However, these methods may still suffer from prolonged waiting or incomplete task recovery when dynamic obstacles repeatedly constrain the reference path. In contrast, the proposed method combines adaptive feasible-space representation, predictive risk assessment, target selection within the feasible region, and passage/replanning decisions, allowing navigation continuity to be maintained under a wider range of dynamic conditions.

The statistical results are summarized in Table 5.

Table 5. Comprehensive navigation performance over 30 trials.

Method	Successful Runs/30	Arrival Time (s)	Path Length (m)	Minimum Dynamic Safety Margin (m)	Mean Tracking Error (m)
Traditional DWA	2	114.8 ± 14.42	162.90 ± 12.15	0.176 ± 0.063	0.309 ± 0.097
Hahn-DWA	6	71.20 ± 8.23	138.10 ± 10.63	0.189 ± 0.275	0.560 ± 0.126
GOA-DWA	8	133.70 ± 15.13	139.24 ± 14.64	0.275 ± 0.154	0.663 ± 0.321
Predictive DWA	14	104.3 ± 10.80	144.47 ± 34.82	−0.028 ± 0.403	1.271 ± 2.367
TAP	17	56.86 ± 11.83	125.50 ± 7.68	0.082 ± 0.103	1.174 ± 0.960
Proposed method	24	75.80 ± 10.89	122.278 ± 4.27	0.5413 ± 0.442	0.2691 ± 0.126

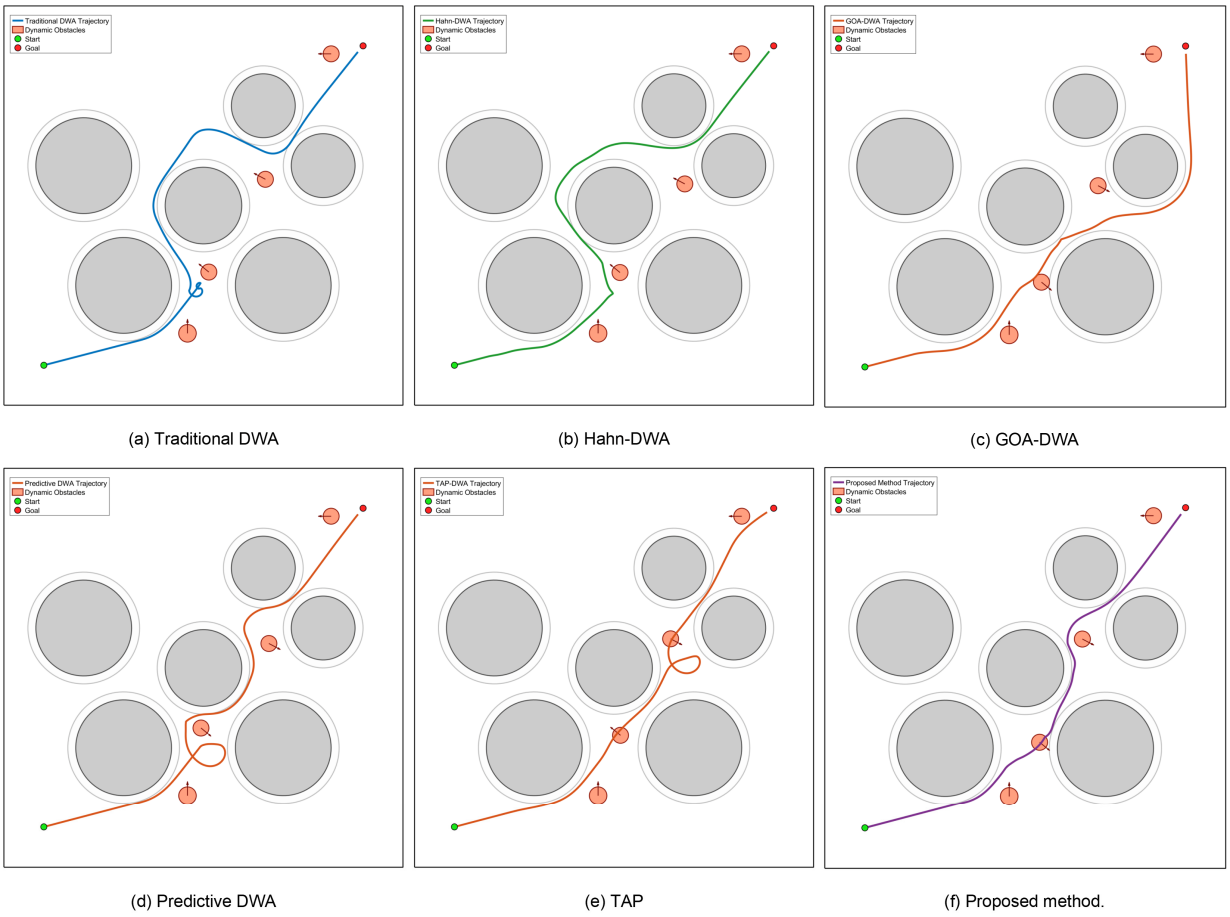


Figure 11. Representative navigation trajectories in the comprehensive dynamic environment: (**a**) Traditional DWA; (**b**) Hahn-DWA; (**c**) GOA-DWA; (**d**) Predictive DWA; (**e**) TAP; (**f**) Proposed method. Note: The gray filled circles represent static obstacles, their light-gray outlines mark the inflated boundaries, and the orange circles and arrows indicate dynamic obstacles and their directions of motion.

Because the continuous performance metrics were calculated only over successful collision-free runs, these metrics should be interpreted together with the corresponding navigation success rates.

The proposed method achieves 24 successful runs out of 30, corresponding to a success rate of 80.0%, which is higher than TAP (56.7%), Predictive DWA (46.7%), GOA-DWA (26.7%), Hahn-DWA (20.0%), and Traditional DWA (6.7%). This result indicates that the main advantage of the proposed method lies in its improved task-completion robustness under varying dynamic-obstacle conditions. All six unsuccessful runs reached the 180 s simulation time limit without reaching the goal, and none had a recorded collision. These failures were associated with prolonged waiting, repeated absence of feasible local velocity commands, and recovery actions that did not restore sufficient task progress.

In addition, the proposed method achieves the shortest mean path length of 122.278 m, the largest mean minimum dynamic safety margin of 0.5413 m, and the lowest mean tracking error of 0.2691 m among the six methods. A negative dynamic safety margin indicates that the prescribed safety-clearance margin was violated rather than that physical collision necessarily occurred.

The proposed method does not achieve the shortest arrival time. TAP obtains the lowest mean arrival time of 56.86 s, compared with 75.80 s for the proposed method. However, TAP succeeds in only 17 of the 30 trials, whereas the proposed method succeeds in 24.

Therefore, arrival time should be interpreted together with task-completion rate rather than as an isolated metric.

Overall, the proposed method achieved the highest observed navigation success rate in the tested scenario. Among successful collision-free runs, it also achieved the shortest mean path length, the largest mean minimum dynamic safety margin, and the lowest mean tracking error. These results should be interpreted alongside its longer mean arrival time than TAP.

5. Discussion

The comprehensive results indicate a trade-off between task-completion robustness and traversal efficiency. The proposed method completed 24 of 30 trials (80.0%), compared with 17 of 30 trials (56.7%) for TAP, while the corresponding mean arrival times were 75.80 and 56.86 s, respectively (Table 5). The mechanism-validation results suggest that passage-window verification and recovery from persistent blockage can help maintain task progress. Waiting for suitable passage opportunities and cautious corner handling may also increase traversal time. However, arrival times were calculated over each method's successful collision-free runs, which comprise different subsets of trials. The observed time difference should therefore be interpreted alongside success rates and should not be treated as a paired estimate of the time cost of individual mechanisms.

Time-aligned prediction in TAP [15] evaluates robot–obstacle separation at matching future instants. Yang et al. [19] introduce collision-risk assessment and use fuzzy control to adjust DWA evaluation weights. Zhai et al. [20] incorporate a safety-distance matrix into A* planning and combine it with adaptive-weight DWA. The present method coordinates synchronized prediction, asymmetric feasible-region constraints, lateral target selection, and passage and replanning decisions. The mechanism-validation results help explain the roles of these components: feasible-region constraints alone did not produce the same lateral avoidance as offset-target selection in Figures 5 and 6, while the proposed method resumed task progress after replanning in the persistent-blockage scenario shown in Figure 10. These observations illustrate the roles of target selection and navigation recovery under the specific conditions examined. The quantitative comparisons in Table 5 concern the baseline implementations used in this study, all provided with the same basic RRT-based recovery mechanism. Differences in maps, obstacle-motion assumptions, robot constraints, and evaluation metrics limit direct numerical comparisons with results reported in the original papers.

The validation was conducted in MATLAB using prescribed obstacle-motion rules and known obstacle states. Additional simulations examined selected deviations from the assumed obstacle-motion models and finite acceleration and deceleration limits. However, sensing errors, localization uncertainty, communication delays, and realistic actuator dynamics were not explicitly evaluated. Robustness to these untested factors and transferability to physical robots therefore remain unverified. In addition, the reported mean computation times do not establish per-cycle deadline guarantees, particularly during global replanning.

The separate clearance sensitivity study was limited to ten paired trials per setting in a fixed map and initial reference-path configuration. The retained 0.8 m value therefore represents an empirical engineering choice under the tested conditions. At the nominal clearance of 0.8 m, all six replanning attempts in trial 02 were rejected because the candidate routes exceeded the detour-length or path-deviation limits. The robot remained stationary from 56.1 s as a moving obstacle approached. The sampled physical clearance decreased from 0.0150 m at 76.0 s to −0.0046 m at 76.1 s, when a collision was recorded. In trial 06, the robot remained stationary during a sequence of control cycles with no admissible local DWA candidate. A reciprocating obstacle passed the robot, reducing the sampled

physical clearance to 0.7505 m at 45.3 s, 0.0495 m below the nominal clearance. The robot subsequently resumed motion and reached the goal at 98.8 s. These cases show that physical clearance can fall below the nominal planning clearance while the robot holds position and an obstacle continues to move. Furthermore, safety metrics were evaluated at discrete 0.1 s intervals, so continuous-time safety is not established.

Future work will investigate uncertainty-aware obstacle prediction and the calibration of the nominal clearance and passage-decision parameters under sensing and localization errors. Particular attention will be given to waiting-position selection and recovery decisions when no safe local velocity is available. These developments will be assessed using executed obstacle separation, collision occurrence, task completion, and traversal efficiency. Validation will also be extended to additional maps, obstacle densities, and motion patterns using independent trials. Computational profiling will measure the maximum observed computation time per control cycle and the cost of global replanning. Physical-robot experiments will evaluate performance under realistic perception and actuation conditions. A further objective is to reduce unnecessary waiting and conservative corner handling while maintaining appropriate obstacle separation.

6. Conclusions

This study proposed a predictive and risk-adaptive DWA with feasible-region constraints for robot navigation in spatially constrained dynamic environments. The method combines an asymmetric feasible region, target selection informed by available space and upcoming corners, synchronized dynamic-obstacle prediction, approximate time-to-collision assessment, risk-adaptive trajectory evaluation, and a dynamic passage and replanning strategy. These mechanisms enable the robot to exploit locally available free space, respond to predicted dynamic risks, and recover navigation progress when the original reference path becomes temporarily or persistently unavailable.

Mechanism-validation experiments demonstrated the roles of the proposed components in the tested scenarios involving asymmetric feasible regions, dynamic crossing interactions, consecutive turns, and blocked passages. In the comprehensive experiment, the proposed method achieved the highest observed success rate among the six compared methods, completing 24 of 30 trials (80.0%). Among successful collision-free runs, it achieved the shortest mean path length of 122.278 m, the largest mean minimum dynamic safety margin of 0.5413 m, and the lowest mean tracking error of 0.2691 m. Its mean arrival time was 75.80 s, compared with 56.86 s for TAP, indicating a trade-off between task-completion robustness and traversal efficiency under the tested conditions.

The mean computation time per cycle reported in Table 2 was below the 0.1 s control period. This result indicates a low average computational cost in that experiment but does not establish that every control cycle of the complete navigation system meets the deadline. Future work will focus on more complex and uncertain dynamic-obstacle motion, improving traversal efficiency under conservative corner handling, and validating the proposed method through physical mobile-robot experiments in real dynamic environments.

Author Contributions: Conceptualization, Y.Q. and Y.Z.; methodology, Y.Q. and Z.Z.; software, Z.Z.; validation, Z.Z. and Y.Q.; formal analysis, Z.Z.; investigation, Z.Z.; resources, Y.Q. and R.M.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Y.Q., Y.Z. and R.M.; visualization, Z.Z.; supervision, Y.Q., Y.Z. and R.M.; project administration, Y.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jiangsu Province Industry-University-Research Collaboration Project, grant number BY20221001, and the School-Level Research Project of Yancheng Institute of Technology, grant number xjr2021013.

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

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

References

1. Fragapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**, *294*, 405–426. [CrossRef]
2. Tang, Y.; Zakaria, M.A.; Younas, M. Path planning trends for autonomous mobile robot navigation: A review. *Sensors* **2025**, *25*, 1206. [CrossRef] [PubMed]
3. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
4. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
5. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report TR 98-11; Computer Science Department, Iowa State University: Ames, IA, USA, 1998.
6. Luo, S.; Zhang, M.; Zhuang, Y.; Ma, C.; Li, Q. A survey of path planning of industrial robots based on rapidly exploring random trees. *Front. Neurorobot.* **2023**, *17*, 1268447. [CrossRef] [PubMed]
7. Tang, Y.; German, Y.O. Enhanced Q-Learning for Grid Navigation: Direction-Sensitive Priors, Annealed Exploration, and Potential-Based Reward Optimization. *Inform. Autom.* **2026**, *25*, 1235–1259. [CrossRef]
8. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [CrossRef]
9. Sun, Y.; Wang, W.; Xu, M.; Huang, L.; Shi, K.; Zou, C.; Chen, B. Local path planning for mobile robots based on fuzzy dynamic window algorithm. *Sensors* **2023**, *23*, 8260. [CrossRef] [PubMed]
10. Li, J.; Sun, J.; Liu, L.; Xu, J. Model predictive control for the tracking of autonomous mobile robot combined with a local path planning. *Meas. Control* **2021**, *54*, 1319–1325. [CrossRef]
11. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
12. Missura, M.; Bennewitz, M. Predictive collision avoidance for the dynamic window approach. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8620–8626. [CrossRef]
13. Kobayashi, M.; Motoi, N. Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles. *IEEE Access* **2022**, *10*, 17018–17029. [CrossRef]
14. Hahn, B. Enhancing obstacle avoidance in dynamic window approach via dynamic obstacle behavior prediction. *Actuators* **2025**, *14*, 207. [CrossRef]
15. Tao, Y.; Song, Y.; Wan, J.; Zhang, W.; Liu, H.; Wang, X.; Wei, H. TAP-time-aligned prediction: An improved dynamic obstacle avoidance method for mobile robots. *Int. J. Dyn. Control* **2025**, *13*, 206. [CrossRef]
16. Chen, S.; Liu, H.; Liu, M.; Chen, S. Prob-DWA: A probabilistic collision prediction-based dynamic window approach for obstacle avoidance. *Robot. Auton. Syst.* **2026**, *202*, 105496. [CrossRef]
17. Motoi, N.; Ueda, Y. Velocity obstacle considering collision prediction time and non-holonomic constraint for mobile robot. In Proceedings of the 2024 IEEE 18th International Conference on Advanced Motion Control (AMC), Kyoto, Japan, 28 February–1 March 2024; pp. 1–6. [CrossRef]
18. Cao, Y.; Nor, N.M. An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation. *Decis. Anal. J.* **2024**, *11*, 100471. [CrossRef]
19. Yang, L.; Li, C.; Wang, Q.; Shi, S.; Yang, M. Adaptive dynamic windowing approach based on risk degree function. *Trans. Inst. Meas. Control* **2024**, *46*, 1542–1550. [CrossRef]
20. Zhai, X.; Tian, J.; Li, J. A real-time path planning algorithm for mobile robots based on safety distance matrix and adaptive weight adjustment strategy. *Int. J. Control Autom. Syst.* **2024**, *22*, 1385–1399. [CrossRef]
21. Wang, J.; Yang, L.; Cen, H.; He, Y.; Liu, Y. Dynamic obstacle avoidance control based on a novel dynamic window approach for agricultural robots. *Comput. Ind.* **2025**, *167*, 104272. [CrossRef]
22. Yao, M.; Deng, H.; Feng, X.; Li, P.; Li, Y.; Liu, H. Improved dynamic windows approach based on energy consumption management and fuzzy logic control for local path planning of mobile robots. *Comput. Ind. Eng.* **2024**, *187*, 109767. [CrossRef]
23. Meng, Y.; Cao, M. Path planning for mobile robots in dynamic environments based on a multistrategy-enhanced dynamic window approach. *Trans. Inst. Meas. Control* **2026**, *48*, 4596–4608. [CrossRef]
24. Hou, J.; Jiang, W.; Luo, Z.; Yang, L.; Hu, X.; Guo, B. Dynamic path planning for mobile robots by integrating improved sparrow search algorithm and dynamic window approach. *Actuators* **2024**, *13*, 24. [CrossRef]
25. Zhou, R.; Zhou, K.; Wang, L.; Wang, B. An improved dynamic window path planning algorithm using multi-algorithm fusion. *Int. J. Control Autom. Syst.* **2024**, *22*, 1005–1020. [CrossRef]

26. Lu, Y.; Da, C. Global and local path planning of robots combining ACO and dynamic window algorithm. *Sci. Rep.* **2025**, *15*, 9452. [CrossRef] [PubMed]
27. Lv, C.; Wang, H.; Wang, S.; Cai, M.; Chen, S. Path planning of mobile robots based on safety in real-time dynamic environments. *Robot. Auton. Syst.* **2026**, *204*, 105568. [CrossRef]
28. Qin, L.; Gao, Y.; Lu, X. SSA*-PDWA: A hierarchical path planning framework with enhanced A* algorithm and dynamic window approach for mobile robots. *Comput. Mater. Contin.* **2026**, *87*, 86. [CrossRef]
29. Jiang, X.; Kuroiwa, T.; Cao, Y.; Sun, L.; Zhang, H.; Kawaguchi, T.; Hashimoto, S. Enhanced pure pursuit path tracking algorithm for mobile robots optimized by NSGA-II with high-precision GNSS navigation. *Sensors* **2025**, *25*, 745. [CrossRef] [PubMed]
30. Tonola, C.; Faroni, M.; Beschi, M.; Pedrocchi, N. Anytime informed multi-path replanning strategy for complex environments. *IEEE Access* **2023**, *11*, 4105–4116. [CrossRef]
31. Liu, Y.; Zhang, L.; Liu, J.; She, Y.; Zhang, B.; Min, F. Dynamic Window Approach using GOA search algorithm for collision avoidance based on obstacle trajectory prediction. *IEEE Access* **2025**, *13*, 44169–44183. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.