

Article

Enhancing Obstacle Avoidance in Dynamic Window Approach via Dynamic Obstacle Behavior Prediction

Bongsu Hahn

Department of Mechanical and Design Engineering, Hongik University, Sejong 30016, Republic of Korea; hahn@hongik.ac.kr

Abstract: This paper proposes an enhanced local path planning method based on the Dynamic Window Approach (DWA), enabling a mobile robot to safely avoid obstacles and efficiently reach its destination. To overcome the limitations of the conventional DWA in handling dynamic obstacles and to improve goal reachability, the velocity term—originally evaluated solely by speed—was redefined as the distance difference between the robot's predicted future position and the target destination. This modification allows the robot to more effectively anticipate its short-term position while simultaneously considering potential obstacle locations. In particular, a linear prediction model for dynamic obstacle behavior was introduced, which estimates the future positions of obstacles based on their current position, velocity, and heading direction. Under the assumption that obstacles maintain constant speed and direction over short intervals, this model enables the robot to proactively plan avoidance maneuvers before a collision risk arises. Furthermore, a novel risk assessment strategy was incorporated to enhance collision prevention. This approach categorizes obstacles in front of the robot according to both distance and angle, evaluates obstacle density in various directions, and guides the robot toward safer paths with fewer surrounding obstacles. The effectiveness of the proposed method was validated through extensive simulations, comparing the conventional DWA, a modified DWA with the new velocity term, and the proposed DWA with dynamic obstacle behavior prediction. The results demonstrated that the proposed approach significantly reduced the number of collisions and overall travel time, thereby confirming its superiority in highly dynamic and uncertain environments.



Academic Editor: Guanghong Yang

Received: 24 March 2025

Revised: 22 April 2025

Accepted: 23 April 2025

Published: 24 April 2025

Citation: Hahn, B. Enhancing Obstacle Avoidance in Dynamic Window Approach via Dynamic Obstacle Behavior Prediction. *Actuators* **2025**, *14*, 207. <https://doi.org/10.3390/act14050207>

Copyright: © 2025 by the author. 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Dynamic Window Approach (DWA); obstacle behavior prediction; collision avoidance

1. Introduction

Indoor autonomous robots are being utilized in various applications, such as automated transport in logistics centers, drug delivery in hospitals, and guide robots in office environments, and are leading innovation in modern industries and service fields. However, most of these applications are typically conducted in indoor environments, which are characterized by narrow passageways, the sudden appearance of obstacles, and the presence of moving people. As a result, robots must navigate through complex and unpredictable conditions, encountering both static and dynamic obstacles that can significantly impact their movement. For smooth and efficient operation, accurate and reliable obstacle avoidance technology is essential. In other words, obstacle avoidance technology plays a critical role in autonomous robot systems, enabling them to perceive their surroundings, detect obstacles, and adjust their paths in real time to safely reach their destination. In

particular, real-time obstacle avoidance for dynamic obstacles is a fundamental requirement for ensuring a robot's agile response in unpredictable indoor environments.

In the early days of robotics, traditional techniques for obstacle avoidance in indoor autonomous robots, such as the Vector Field Histogram (VFH) [1], the potential field method [2], and the Bug algorithm [3], were widely used due to their real-time processing capability and simplicity of implementation. However, these traditional methods have limitations in complex environments and dynamic obstacle avoidance, making them rarely used in modern applications. For instance, one major drawback is the tendency to get stuck in a local optimum, causing the robot to stop at a specific state. This makes such methods unsuitable for navigating complex environments and avoiding dynamic obstacles.

To overcome the limitations of these traditional methods, recent studies have explored velocity-based, probability-theory-based, and reinforcement-learning-based avoidance strategies that consider robot dynamics, environmental complexity, and dynamic obstacles [4–6]. Among these approaches, probability-theory-based and reinforcement-learning-based strategies provide effective obstacle avoidance, even in complex dynamic environments. However, they have the drawback of requiring considerable computational resources, making real-time processing challenging when the robot continuously recalculate its movement path based on changing environmental information. As a result, velocity-based avoidance techniques, which can be implemented in a relatively simple manner, are widely utilized for real-time obstacle avoidance in robots.

Representative velocity-based obstacle avoidance techniques include the DWA [7] and Velocity Obstacles (VOs) [8]. The VO technique calculates the relative velocity of the robot and obstacles in the velocity space and predicts the potential collisions. However, it has a major drawback: it does not explicitly account for the robot's dynamic constraints, which can result in discontinuous or abrupt avoidance paths that are difficult for the robot to follow in real-world scenarios.

To address the drawbacks of VOs, modified and extended VO algorithms, such as Generalized Velocity Obstacles (GVOs) [8,9], Acceleration Velocity Obstacles (AVOs) [10], and Curvature Velocity Method (CVM) [11], have been developed. These modified methods aim to overcome the dynamic limitations of traditional VOs. However, they still suffer drawbacks, such as reduced obstacle avoidance speed or increased computational complexity.

On the other hand, the DWA is a local path planning algorithm that generates velocity commands in real time while considering the robot's dynamic constraints. The DWA calculates the velocity window in which the robot can move without collision in the velocity space and selects the optimal velocity among this set. Since the DWA explicitly accounts for the robot's acceleration, deceleration, and rotational speed limits, it ensures that the selected path remains within the robot's physically feasible range. By simultaneously considering both velocity and rotational speed, the DWA generates more realistic and dynamically viable trajectories.

However, despite its advantage, the DWA is not a perfect solution in all situations. Since it was originally designed for avoiding static obstacles, it has limitations in selecting an appropriate velocity space when navigating dynamic obstacles or complex environments. Additionally, its avoidance performance can vary depending on the weight values assigned to the evaluation function, which determines the optimal velocity command. To address these shortcomings, various studies have explored improvements to the DWA using machine learning, reinforcement learning, and adaptive control [12–17]. These approaches are aimed at optimizing the evaluation function weights or enhancing path planning performance through non-linear control models. In particular, Cheng et al. [18] proposed a cross-platform deep-reinforcement-learning model that enables autonomous navigation in

diverse scenes without relying on global information. Such an approach allows effective path planning, even in highly complex and unpredictable environments. However, they did not explicitly account for dynamic obstacles, making it difficult to generalize to new, unseen dynamic obstacles that were not included in the training environment.

In this paper, we propose a local path planning method that enhances the DWA to enable a mobile robot to safely avoid obstacles and efficiently reach its destination. While the conventional DWA evaluates the velocity component solely based on speed, our method redefines this term as the distance between the robot's predicted future position and the destination. This modification improves goal reachability and reduces path deviation during navigation, especially in dynamic environments.

Moreover, the proposed approach incorporates a linear prediction model that estimates the future positions of both the robot and nearby obstacles based on their current velocity and direction. This forward-looking mechanism enables the robot to anticipate potential collisions and proactively adjust its path. To further enhance safety, we introduce a novel multi-dimensional risk assessment strategy that classifies obstacles in front of the robot based on their relative distance, angle, and spatial density. The robot is then guided to navigate toward the direction with fewer obstacles, resulting in safer and more agile movement, even in unpredictable indoor environments.

The novelty of our method lies in the combination of predictive modeling and a risk-aware evaluation function. Specifically, to the best of our knowledge, this is the first approach to redefine the velocity evaluation metric in the DWA using goal-oriented distance rather than raw speed. Additionally, the integrated risk terms—based on distance, angle, and density—provide a more comprehensive assessment of dynamic obstacle interference, significantly improving the overall performance of the DWA in dynamic scenarios.

This contribution is particularly valuable in scenarios requiring fast, reactive planning in dynamic and uncertain environments. This study focuses on enhancing the DWA, a widely adopted real-time local planner for mobile robots, rather than aiming to surpass all existing path planning frameworks. The objective is to improve the DWA's performance in dynamic environments while preserving its real-time applicability and computational efficiency.

While no direct quantitative comparison is made with reinforcement learning, potential field methods, or other contemporary techniques, the discussion serves to clarify the rationale for selecting a DWA-based framework. RL-based approaches often demand extensive training data and suffer from generalization issues in unseen scenarios, whereas potential field methods are prone to local minima. In contrast, the proposed method improves real-time adaptability and reliability by incorporating short-term prediction and risk-aware evaluation.

The remainder of this paper is structured as follows. Section 2 describes the conditions of the robot and obstacles used in the proposed method. Section 3 presents the obstacle avoidance approach based on the DWA integrated with dynamic obstacle behavior prediction. Section 4 provides simulation results to verify the effectiveness of the proposed method. Finally, Section 5 concludes the study and discusses potential directions for future research.

2. System Description

2.1. Robot Kinematic Model

The robot used in this paper is a two-wheeled mobile robot, consisting of two drive wheels and a supporting caster wheel, as shown in Figure 1. The robot exhibits non-holonomic characteristics, meaning its movement is subject to constraints that prevent it from moving freely in all directions. To navigate, the robot generates linear and angular

velocity by controlling the angular velocity of its drive wheels. The corresponding kinematic model is defined as follows,

$$\mathbf{P}_{robot}(t) = [x_{robot}(t), y_{robot}(t)]^T$$

$$\begin{bmatrix} \dot{x}_{robot}(t) \\ \dot{y}_{robot}(t) \\ \dot{\theta}_{robot}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta_{robot}(t) & 0 \\ \sin \theta_{robot}(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (1)$$

where x_{robot} , y_{robot} are the Cartesian coordinates, θ_{robot} is the heading angle, v is the linear velocity, and ω is the angular velocity of the robot.

The linear velocity v and angular velocity ω of the robot are determined by the angular velocities v_L and v_R of the two wheels, the distance b between the wheels, and the radius r of the wheels. These relationships can be expressed as

$$v = \frac{r}{2}(v_L + v_R), \quad \omega = \frac{r}{b}(v_R - v_L) \quad (2)$$

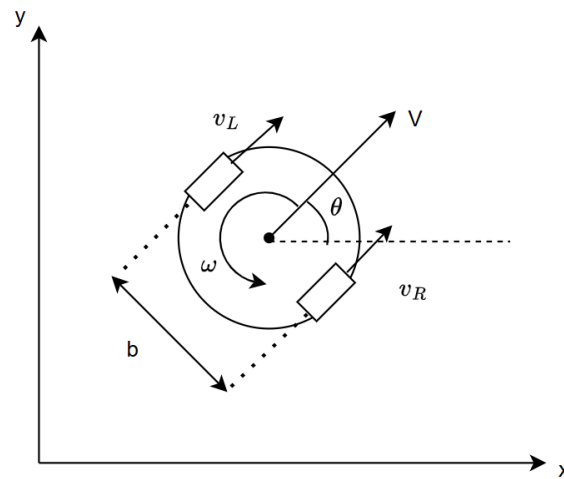


Figure 1. A two-wheeled mobile robot.

2.2. Obstacle Constraints

In indoor environments, humans and animals, which can change direction more easily than vehicles or larger transport systems, are the most common moving entities. Generally, indoor vehicles tend to maintain a consistent direction and velocity of travel, but humans or animals, which are potential obstacles, may exhibit behaviors such as sudden turns, stopping to avoid collisions, or unexpected changes in direction. Therefore, in this paper, obstacles are modeled as holonomic entities to better capture these characteristics. This approach allows for the simulation of unpredictable obstacle behaviors that may occur in indoor environments.

The obstacle's speed is limited to be slower than the robot's speed. This is to prevent situations where collisions become unavoidable, even when using predictive planning, which can occur if the obstacle's speed equals or exceeds that of the robot. Additionally, the obstacle's speed may vary dynamically during interactions at each of the robot's calculation time steps.

The obstacle is assumed to have a circular shape with a defined radius boundary. To standardize the test conditions, all obstacles are set to the same size and speed. Each obstacle has a radius of 0.3 m, which is 0.1 m larger than the robot's radius. This size configuration helps clearly identify potential limitations in the robot's obstacle detection and avoidance process. Additionally, the initial positions and movement directions of the

obstacles are randomly assigned, allowing the robot's obstacle avoidance algorithm to adapt to various scenarios.

3. Enhanced DWA with Dynamic Obstacle Prediction

We propose a method for predicting the behavior of dynamic obstacles and integrating it into the DWA to generate the robot's optimal path, as illustrated in Figure 2. The block diagram in the figure outlines the entire process, from predicting obstacle movement using the location and velocity data of the obstacle to performing path planning and generating the optimal trajectory accordingly.

First, the obstacle's velocity V_{obs} , at the current time t , its movement direction θ_{obs} , and the position (x_{obs}, y_{obs}) are provided as input. This information represents the obstacle's current state and is essential for predicating its subsequent movement. Simultaneously, the robot's target point and current position are also input, defining the target position that the robot must reach.

Based on the obstacle's current state, the future position prediction step is executed to estimate its next point position. This step calculates the coordinates of the obstacle at time $t + 1$ by considering its velocity, movement direction, and time interval Δt . These predictions rely on a linear model that assumes that the obstacle moves at a constant speed and direction. This assumption is reasonable within short prediction intervals, where obstacle motion tends to be locally linear. Although obstacles may exhibit non-linear or erratic behavior over longer durations, their short-term movement can often be approximated as uniform. This trade-off enables efficient real-time computation while maintaining sufficient accuracy for local path planning. Each step in Figure 2 continuously interacts with and responds to changes in the obstacle's movement, ensuring high reliability even under dynamic conditions. The details of each step are explained in the following paragraphs.

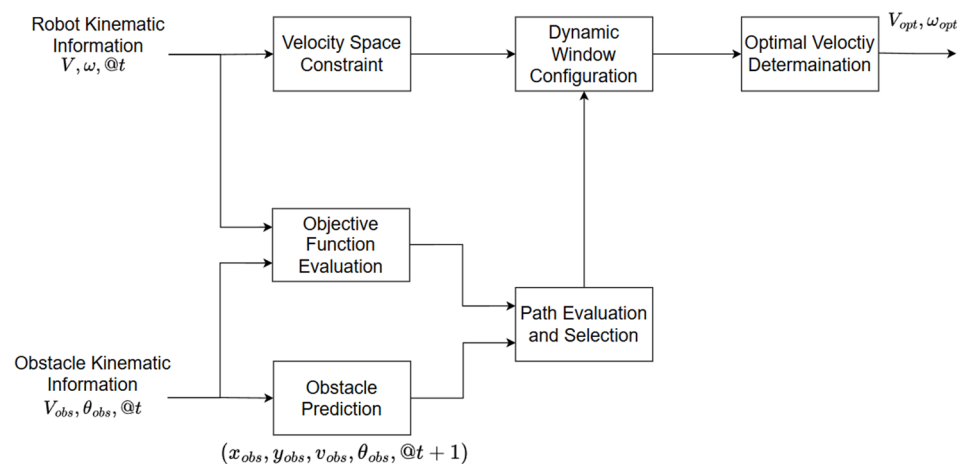


Figure 2. Overall block diagram of the enhanced DWA with dynamic obstacle prediction.

3.1. Dynamic Obstacle Behavior Prediction

We modeled the time-dependent behavior of obstacles based on Figure 3 to predict their future positions. The model estimates the next position of a moving obstacle on a two-dimensional plane using its previous and current positions, velocity, movement direction, and time variation. Predicting an obstacle's next position requires its velocity and movement direction. These future values are predicted based on the change in position over time, using the difference between the previous and current positions as follows,

$$\begin{aligned}
\mathbf{P}_{obs}(t) &= [x_{obs}(t), y_{obs}(t)]^T \\
\begin{cases} x_{obs}(t+1) = x_{obs}(t) + V_{obs}(t) \cos(\theta_{obs}(t))\Delta t \\ y_{obs}(t+1) = y_{obs}(t) + V_{obs}(t) \sin(\theta_{obs}(t))\Delta t \end{cases} \\
\theta_{obs}(t) &= \tan^{-1} \left(\frac{y_{obs}(t) - y_{obs}(t-1)}{x_{obs}(t) - x_{obs}(t-1)} \right) \\
V_{obs}(t) &= \frac{\sqrt{(x_{obs}(t) - x_{obs}(t-1))^2 + (y_{obs}(t) - y_{obs}(t-1))^2}}{\Delta t}
\end{aligned} \tag{3}$$

where $\mathbf{P}_{obs}$, x_{obs} , and y_{obs} represent the obstacle's position vector, x-position, and y-position, respectively, while θ_{obs} and V_{obs} denote the movement direction and velocity of the obstacle.

The formula assumes that the obstacle moves in a constant direction $\theta_{obs}(t)$ with a constant speed $V_{obs}(t)$ from its current location. The linear and angular velocity of the obstacle are determined using position data from its previous and current locations. To predict the obstacle's next position, its movement direction is decomposed into x-axis and y-axis components. The displacement in each direction is calculated and added to estimate the next location. Specifically, the displacement in the x-axis is given by $\Delta x = V_{obs}(t)\cos(\theta_{obs}(t))\Delta t$, which represents the contribution of the x-axis component in the movement direction. Similarly, the displacement in the y-axis direction is given by $\Delta y = V_{obs}(t)\sin(\theta_{obs}(t))\Delta t$. By summing these displacement components with the current position, the obstacle's next location is predicted.

In the proposed method, the movement of dynamic obstacles is accounted for through a linear prediction model based on their current velocity and heading direction. Specifically, the obstacle's current position $\mathbf{P}_{obs}(t)$, velocity $V_{obs}(t)$, and heading $\theta_{obs}(t)$ are used to estimate its future position $\mathbf{P}_{obs}(t + \Delta t)$. This assumes that the obstacle maintains a constant velocity and direction over a short prediction horizon.

In real-world scenarios, such modeling can be achieved by continuously tracking obstacles using onboard sensors such as LiDAR, RGB-D cameras, or radar. Velocity and heading can be inferred using frame-to-frame position changes, for example, through Kalman filters or visual odometry. While the proposed linear model is a simplified approximation, it is suitable for real-time path planning due to its computational efficiency. For complex environments, this framework can be extended to incorporate probabilistic prediction models or learned motion patterns from observed data.

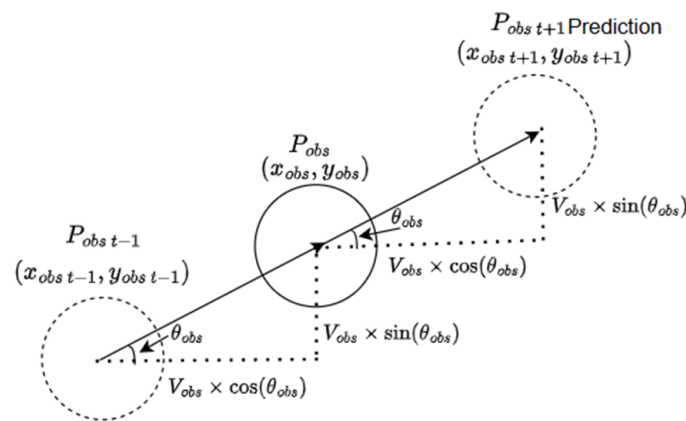


Figure 3. Visualization of future obstacle position prediction.

3.2. Dynamic Window: Velocity Constraints and Settings

Before determining the linear velocity v and angular velocity ω of a mobile robot, it is essential to define its physical and dynamic velocity limits. Without this process, the robot may generate abnormal velocity outputs, making it impossible to accurately determine the distance it can travel within a given time. Defining the velocity space constraints based on

the robot's physical characteristics and feasible movements is a critical step in mobile robot motion planning.

To enhance DWA for dynamic obstacle avoidance, we incorporate future obstacle positions into the dynamic window computation. The conventional DWA assumes that obstacles remain stationary at their current positions $\mathbf{P}_{obs}(t)$, which can lead to suboptimal path selection when navigating in environments with moving obstacles. This limitation arises because the set of feasible velocity pairs (v, ω) is determined without considering future obstacle movement. The conventional dynamic window is defined as follows,

$$V_d = [v_{min}, v_{max}], \Omega_d = [\omega_{min}, \omega_{max}] \quad (4)$$

where v_{max} and ω_{max} denote the maximum linear velocity and maximum angular velocity of the robot, respectively, as determined by its physical constraints.

To address this issue, the proposed enhanced DWA generates a dynamic window based on the predicted obstacle position $\mathbf{P}_{obs}(t+1)$. Based on the future obstacle position (3), the dynamic window is defined as follows,

$$\begin{aligned} V_d^{mod} &= \left\{ v \mid v_{min} \leq v \leq v_{max}, \text{ if } d_{safe}(v, \omega, \mathbf{P}_{obs}(t+1)) > r_{robot} + r_{obs} \right\} \\ \Omega_d^{mod} &= \left\{ \omega \mid \omega_{min} \leq \omega \leq \omega_{max}, \text{ if } d_{safe}(v, \omega, \mathbf{P}_{obs}(t+1)) > r_{robot} + r_{obs} \right\} \end{aligned} \quad (5)$$

where $d_{safe}(v, \omega, \mathbf{P}_{obs}(t+1))$ represents the minimum distance between the robot and the predicted obstacle position given a velocity pair (v, ω) and r_{robot} and r_{obs} are the radii of the robot and the obstacle, respectively.

This modification ensures that the robot selects only those velocity pairs that prevent future collisions. By considering the obstacle's predicted motion, the robot can proactively adjust its trajectory to avoid potential collisions rather than reacting to obstacles only when they are directly in its path. As illustrated in Figure 4, the proposed method prevents the robot from choosing high-risk trajectories (marked in red), which would result in future collisions. Instead, the robot identifies safer trajectories (marked in black) by steering toward regions with a lower predicted obstacle density. This approach significantly enhances the effectiveness of the DWA in dynamic environments, improving both collision avoidance and navigation efficiency.

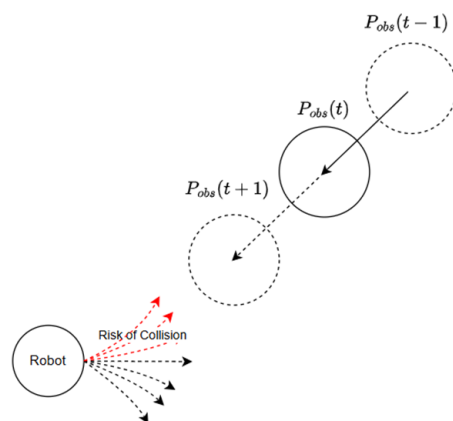


Figure 4. Dynamic window visualization.

3.3. The Objective Function

The DWA algorithm without obstacle behavior prediction generates a local path in real time using a dynamic window and is effective at avoiding static obstacles and some

dynamic obstacles. However, it still struggles with avoiding multiple dynamic obstacles. To address this limitation, this paper proposes an evaluation function, defined as follows:

$$J(v, \omega) = J_{heading}(v, \omega) + J_{distance}(v, \omega) + J_{clearance}(v, \omega) + J_{future(obs)}(v, \omega, v_o, \omega_o) + J_{risk(obs)}(v, \omega) + J_{count(obs)}(v, \omega) \quad (6)$$

By integrating future obstacle position prediction $J_{future(obs)}$, a penalty term $J_{risk(obs)}$ based on the obstacle distance and angle, and another penalty term $J_{count(obs)}$ based on the number of obstacles into the standard DWA objective function, the proposed function enables more precise obstacle avoidance while enhancing stability and overall performance. Each term in Equation (6) is derived as follows.

The term $J_{heading}$ evaluates the robot's heading angle toward the destination and is mathematically defined by

$$J_{heading} = -\alpha_h \times \left| \theta_{Heading}(t) \right|$$

$$\theta_{Heading}(t) = \tan^{-1} \left(\frac{y_{goal} - y_{robot}(t)}{x_{goal} - x_{robot}(t)} \right) - \theta_{robot}(t) \quad (7)$$

where $\theta_{Heading}$ represents the robot's current heading angle toward the destination and x_{goal} , y_{goal} , x_{robot} , and y_{robot} denotes the x and y coordinates of the destination and the robot's current position, respectively. The α_h is a weight factor.

The term $J_{distance}$ is defined as the Euclidean distance between the robot's current position and the destination and is formulated with a weighting factor α_d as follows:

$$J_{distance} = -\alpha_d \sqrt{(x_{goal} - x_{robot}(t))^2 + (y_{goal} - y_{robot}(t))^2} \quad (8)$$

In the conventional DWA, the $J_{distance}$ primarily evaluates the robot's speed, assigning a higher score to faster velocities regardless of the direction. However, unlike the DWA without obstacle behavior prediction, which is designed with static obstacle avoidance in mind, the DWA with dynamic obstacle behavior prediction is specifically developed for environments with multiple dynamic obstacles. To address this, the $J_{distance}$ is redefined as the difference in the distance to the destination, ensuring that the robot gradually reduces this distance. This adjustment accounts for scenarios where the robot may deviate significantly from the shortest path while navigating around obstacles.

$J_{clearance}$ is a scoring term that evaluates the distance to nearby obstacles and is defined with a weighting factor α_c as follows:

$$J_{clearance} = e^{\alpha_c(d_{safe} - clearance_{obs})},$$

$$d_{safe} = \frac{V_{max}}{4},$$

$$clearance_{obs} = \|\mathbf{P}_{obs}(t) - \mathbf{P}_{robot}(t+1)\|. \quad (9)$$

It represents the difference between the obstacle's current position and the robot's predicted future position. By comparing the robot's current distance to the obstacle with its distance when moving at its current direction and speed, the risk of collision along the robot's expected trajectory can be assessed. $J_{clearance}$ is designed to increase exponentially, ensuring that distant obstacles are perceived as low-risk, while nearby obstacles incur a strong penalty, encouraging the robot to move away rapidly. d_{safe} in Equation (9) defines the safety distance for collision avoidance. This parameter is set to enable the robot to perform smooth evasive maneuvers over the course of three dynamic window iterations.

$J_{future(obs)}$ incorporating predictions for both the future positions of the obstacle and the robot is defined as

$$\begin{aligned} J_{future(obs)} &= e^{\alpha_f(dist_{safe}-dist_{future(obs)})}, \\ dist_{future(obs)} &= \|P_{obs}(t+1) - P_{robot}(t+1)\|, \end{aligned} \quad (10)$$

This allows the robot to proactively respond to potential collisions, making it particularly effective in avoiding high-speed approaching obstacles. The influence of $J_{future(obs)}$ is minimal for distant obstacles due to its exponential scaling, similar to $J_{clearance}$. However, when the obstacle distance falls within d_{safe} , the robot enters a high-risk collision zone. At this stage,

$J_{future(obs)}$ becomes more significant, ensuring that the robot adjusts its trajectory by actively predicting and avoiding nearby dynamic obstacles.

$J_{risk(obs)}$ facilitates effective obstacle avoidance by assigning differential penalties based on the distance and angle of obstacles relative to the robot's movement direction. It is mathematically defined for N obstacles as follows,

$$\begin{aligned} J_{risk(obs)} &= \sum_{n=1}^N f(\theta_H, d_n) \\ f(\theta_H, d_a) &= \begin{cases} \beta_1, |\theta_H| \leq \frac{\pi}{12} \text{ and } d_a > dist_{safe} \\ \beta_2, |\theta_H| \leq \frac{\pi}{12} \text{ and } d_a \leq dist_{safe} \\ \beta_3, |\theta_H| \leq \frac{\pi}{6} \text{ and } d_a \leq dist_{safe} \\ \beta_4, |\theta_H| \leq \frac{\pi}{6} \text{ and } d_a \leq 2dist_{safe} \\ \beta_5, |\theta_H| \leq \frac{\pi}{6} \text{ and } d_a \leq 3dist_{safe} \end{cases} \\ \theta_H &= \tan^{-1} \left(\frac{y_{obs} - y_{robot}}{x_{obs} - x_{robot}} \right) - \theta_{robot} \\ d_n &= \sqrt{(x_{obs,n} - x_{robot})^2 + (y_{obs,n} - y_{robot})^2} \end{aligned} \quad (11)$$

where θ_H , and d_n represent the angle and distance between the robot and the obstacle, respectively. When an obstacle is detected within a certain distance in the robot's field of view, a differentiated score is assigned to appropriately adjust the perceived risk. However, if no obstacle is present in the movement direction, even at a short distance, the risk is considered low.

The parameters $\beta_1, \beta_2, \beta_3, \beta_4$, and β_5 in $f(\theta_H, d_n)$ act as weighting factors that determine the penalty assigned based on the obstacle's distance and angle relative to the robot's movement direction. These coefficients are designed to progressively increase as the obstacle gets closer or deviates further from the robot's heading, ensuring an adaptive risk assessment for obstacle avoidance. Specifically, β_1 applies when the obstacle is relatively far and well-aligned with the robot's direction, exerting minimal influence on the path selection. As the obstacle moves within the safe distance threshold, β_2 applies a slightly higher penalty, while β_3 counts for cases where the obstacle is both closer and more angularly misaligned. When the obstacle is within twice the safe distance, β_4 further increases the avoidance penalty, and β_5 applies the highest penalty when the obstacle is in immediate proximity, requiring rapid maneuvering to prevent collision. To ensure a proper penalty scaling, these values follow a monotonically increasing order, such that $\beta_1 < \beta_2 < \beta_3 < \beta_4 < \beta_5$.

The selection of these coefficients was determined through empirical tuning, where the penalty values were manually adjusted based on heuristic rules to ensure effective obstacle avoidance. In this approach, the coefficients are assigned in a progressively increasing manner to reflect the increasing risk posed by nearby obstacles. Typically, an exponential

scaling pattern is used to ensure that distant obstacles exert minimal influence while closer obstacles induce a significantly stronger avoidance response. By carefully setting these values, the robot can effectively navigate dynamic environments, maintaining a balance between obstacle avoidance and efficient path planning.

Finally, $J_{\text{count}(\text{obs})}$ is introduced to proactively mitigate obstacle interference by assessing the number of obstacles on both sides of the robot's movement direction and adjusting the trajectory toward the side with fewer obstacles, defined as

$$J_{\text{count}(\text{obs})} = \begin{cases} -(count_{\text{obsR}} - count_{\text{obsL}}), & count_{\text{obsR}} > count_{\text{obsL}} \\ -(count_{\text{obsL}} - count_{\text{obsR}}), & count_{\text{obsL}} > count_{\text{obsR}} \\ 0, & count_{\text{obsR}} = count_{\text{obsL}} \end{cases} \quad (12)$$

$$count_{\text{obsL}} = \sum_{n=1}^N \delta(-\frac{\pi}{9} < \theta_{H,n} \leq 0)$$

$$count_{\text{obsR}} = \sum_{n=1}^N \delta(0 \leq \theta_{H,n} < \frac{\pi}{9})$$

Here, $count_{\text{obsR}}$ and $count_{\text{obsL}}$ represent the number of obstacles on the right and left sides of the robot's movement direction, respectively. The number of obstacles within a 20-degree angle in the forward field of view is evaluated to ensure that no penalty is imposed when moving toward an area with no obstacles, thereby facilitating proactive obstacle avoidance.

The rationale for determining the number of obstacles based on the direction of movement is that obstacles positioned far behind or to the sides do not directly influence the robot's intended trajectory and thus do not need to be considered in real-time path planning. Unlike conventional approaches that focus on minimizing the shortest distance, $J_{\text{count}(\text{obs})}$ is designed to reduce speed fluctuations caused by obstacles and promote a safer and smoother navigation strategy.

3.4. Optimal Path Selection Based on Route Evaluation

The objective function evaluation determines the optimal linear velocity v and angular velocity ω , which guide the robot toward its target while ensuring safe obstacle avoidance. This is achieved by maximizing the objective function (5). The optimal control inputs (v^* , ω^*) are then determined by solving

$$(v^*, \omega^*) = \arg \max_{(v, \omega) \in \mathbf{D}} J(v, \omega) \quad (13)$$

where $\mathbf{D}$ represents the set of feasible velocities, constrained by the dynamic window limits (5). Once the optimal v^* , ω^* values are selected, the corresponding path within the dynamic window is evaluated based on its predicted trajectory. This path is expected to provide theoretically optimal obstacle avoidance and target-directed movement, but it must be translated into actual motor commands for execution.

During the optimal path selection process, the selected velocities v^* , ω^* are converted into motor control inputs, allowing the robot to execute the planned motion. The robot's new position and orientation are then updated using the kinematic model:

$$\begin{aligned} x_{\text{robot}}(t+1) &= x_{\text{robot}}(t) + v^* \cos(\theta_{\text{robot}}(t)) \Delta t \\ y_{\text{robot}}(t+1) &= y_{\text{robot}}(t) + v^* \sin(\theta_{\text{robot}}(t)) \Delta t \\ \theta_{\text{robot}}(t) &= \theta_{\text{obs}}(t) + \omega^* \Delta t \end{aligned} \quad (14)$$

This iterative update ensures that the robot continuously avoids obstacles and follows a dynamically feasible trajectory toward its target while maintaining smooth and safe navigation.

4. Verification

In this chapter, we evaluate the performance of the proposed dynamic obstacle behavior prediction and avoidance algorithm based on the DWA through simulation studies. Specifically, we verify the validity of the process by which the algorithm accurately predicts an obstacle's future position using its dynamic characteristics, including position, velocity, and movement direction, and subsequently generates an optimal avoidance trajectory for the robot in real time. Through this evaluation, we aim to determine whether or not the proposed method enables the robot to navigate stably and efficiently in complex and highly dynamic environments. The key parameters of the robot and obstacles used for performance evaluation are presented in Table 1. These parameters were carefully chosen by considering the robot's dynamic constraints and its interaction with the environment.

The primary objectives of the experiments conducted in this study are as follows. First, we analyze whether or not the proposed enhanced DWA algorithm can effectively predict the movement of dynamic obstacles. To assess this, we design various scenarios with different obstacle speeds and movement directions and evaluate whether or not the algorithm maintains high prediction accuracy under these varying conditions. Second, we verify whether the generated avoidance path—based on the predicted obstacle positions—satisfies the robot's dynamic constraints while ensuring collision-free movement. To achieve this, we quantitatively evaluate the algorithm's performance in multiple experimental environments.

The experimental setup consists of two main scenarios. The first scenario evaluates the robot's obstacle avoidance performance when it moves along a straight or diagonal path at a constant velocity in a multi-obstacle environment. This scenario assesses the algorithm's ability to predict obstacle movement in a structured environment and generate an appropriate avoidance trajectory. The second scenario evaluates the robot's avoidance performance in an unstructured environment where multiple obstacles move in random directions. This scenario is designed to test the algorithm's effectiveness in highly dynamic and unpredictable conditions.

To visualize the robot's behavior over time, the simulation results include trajectory plots marked with sequential time-step indicators. These numbered markers enable an understanding of how the robot dynamically adapts its path in response to moving obstacles. In particular, Case II demonstrates the trajectory modifications the robot makes when navigating high-density and unpredictable obstacle configurations.

Additionally, the assumptions and constraints used in this experiment are detailed in Section 4.1. These constraints are introduced to simplify the experimental setup in the initial validation stage and to ensure the reliability and reproducibility of the experiments. The main parameters of the robot and obstacles used for performance evaluation are summarized in Table 1, where they have been set to reflect the robot's dynamic characteristics and its interaction with the environment.

In the structured scenarios of Case I (I-1 and I-2), four dynamic obstacles were used, while in the randomized scenarios of Case II twelve dynamic obstacles moving in random directions were deployed. All obstacles were constrained to a maximum speed of 1.05 m/s, as summarized in Table 1.

While the proposed method has been validated through extensive simulations, real-world deployment with physical mobile robots remains as future work. In real environments, factors such as sensor noise, actuator delays, and localization errors may affect the prediction and avoidance performance. Nevertheless, the algorithm's modular structure enables integration with robust localization and filtering techniques (e.g., Kalman filtering or visual-inertial odometry), allowing it to adapt to physical noise and uncertainty.

In terms of computational complexity, the proposed enhancements maintain the core structure of the DWA and involve lightweight extensions such as short-horizon linear prediction and a risk-based cost function. The overall computational load remains suitable for real-time deployment on embedded systems. However, in large-scale environments with numerous dynamic obstacles, optimization and pruning techniques may be incorporated to ensure scalability and responsiveness. Evaluating the method on a physical robot platform in such settings will be a key focus of future research.

Table 1. Robot and obstacle parameters used in the experiment.

Items	Units	Values
Robot radius	m	0.1
Robot maximum speed	m/s	1.5
Robot maximum angular velocity	rad/s	1
Obstacle radius	m	0.15
Obstacle maximum speed	m/s	1.05

4.1. Experimental Assumptions and Constraints

The experiments conducted in this study were performed as simulations in the MATLAB environment, with several assumptions and constraints applied to streamline the experimental process. These assumptions and constraints were introduced to simplify the experimental setup, focusing on evaluating model performance and validating the proposed algorithm.

In the experimental setup, physical phenomena such as wheel slip were not considered. The robot's motion was modeled under ideal conditions, and complex physical effects, such as wheel slippage, which may occur during movement, were omitted. Additionally, sensor data errors were not taken into account, as the simulation directly generated ideal data rather than using actual sensor readings. As a result, sensor noise and measurement errors that may arise in real-world applications were not incorporated into the evaluation process.

To ensure a controlled evaluation, several constraints were imposed on the experiment. The maximum speed of obstacles was limited to 70% of the robot's maximum speed to reflect realistic conditions. If obstacles moved too quickly, the robot's reaction time would be insufficient, increasing the likelihood of collisions. By capping the obstacle speed at 70% of the robot's maximum speed, the experimental environment ensured that the algorithm's performance could be assessed under stable and repeatable conditions. Furthermore, all obstacles were assumed to have a uniform size and shape and move in a linear manner. Standardizing obstacle size and movement patterns removed unnecessary variability from the performance evaluation, allowing the focus to remain solely on path planning and collision avoidance mechanisms.

In addition to the physical assumptions mentioned above, this study also conducts a simulation-based sensitivity analysis to investigate how each weight parameter in the objective function—namely, α_h , α_c , α_f , α_r , and α_d —affects the overall navigation performance. The sensitivity of the overall evaluation function to a specific weight α_i can be theoretically expressed as

$$S_{\alpha_i} = \frac{\partial J}{\partial \alpha_i} \quad (15)$$

However, the objective function J used in the DWA includes non-linear factors and discontinuities such as sudden collisions, obstacle configurations, and environmental constraints, which makes it challenging to obtain a closed-form derivative analytically. In fact, due to these real-world characteristics, the evaluation function does not behave as a

smooth or differentiable function. Therefore, we adopt an empirical approach based on simulation to assess the sensitivity and influence of each weight.

To perform this analysis, we independently varied each weight value from 0.1 to 0.9 in increments of 0.1, while keeping all other weights fixed. For each weight configuration, 20 randomized dynamic obstacle scenarios were simulated, and two key performance metrics—average arrival time and collision rate—were recorded. This allowed us to observe how performance trends change with different weight settings.

Figure 5 visualizes the results of this sensitivity analysis. In most cases, the optimal performance (i.e., shortest arrival time and lowest collision rate) was observed when the weight value ranged approximately between 0.4 and 0.6. However, the shape of the performance curves is non-linear, and in some cases exhibits steep gradients outside this optimal region. This non-linearity arises from the structure of the DWA evaluation function, which integrates multiple competing objectives and is influenced by the robot's real-time kinematic constraints.

It is important to note that the optimal weight values derived in this analysis are based only on two performance metrics: arrival time and collision rate. In real-world applications, other factors—such as path smoothness, obstacle clearance margin, energy efficiency, or actuator limits—may also need to be considered. Therefore, the results presented here represent a partial cross-section of the complete optimization space.

In the case studies that follow, weight parameters were fine-tuned through trial-and-error methods, taking into account multiple performance objectives beyond just time and collision rate. These final weight combinations were then applied to validate the system in structured and unstructured dynamic environments.

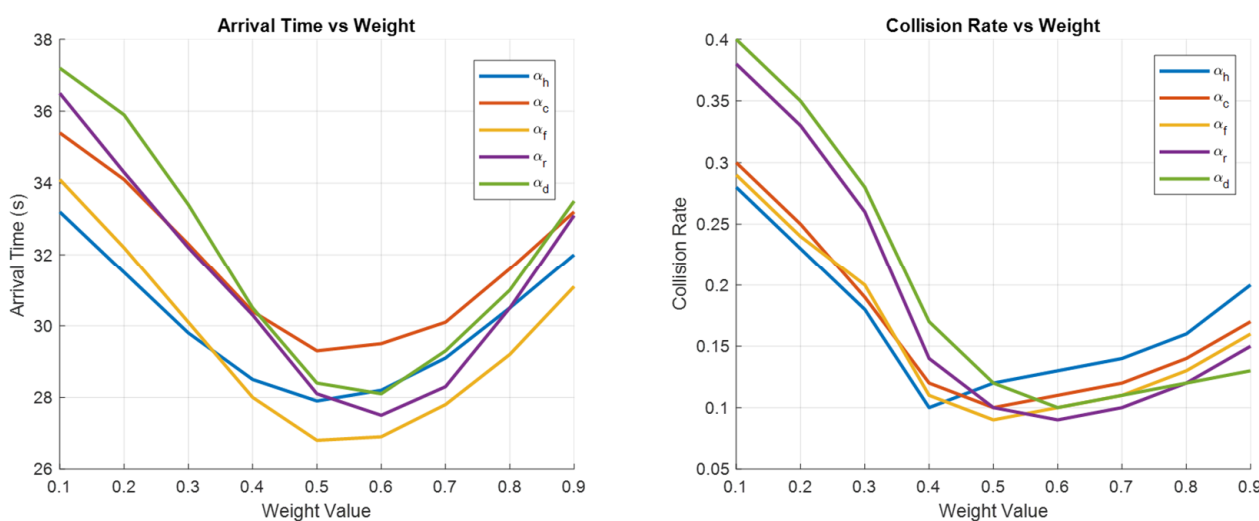


Figure 5. Sensitivity analysis of each weight parameter in the evaluation function: trends in arrival time and collision rate as a function of individual weight values.

4.2. Case I: Structured Path Obstacle Avoidance Experiments

4.2.1. Experimental Setup for Case I

As shown in Figure 6, these experiments consist of two distinct scenarios designed to evaluate the robot's ability to avoid obstacles moving in a predefined direction. In these experiments, four obstacles appear sequentially, requiring the robot to navigate and execute avoidance maneuvers accordingly. The experiment is categorized into Case I-1 and Case I-2, and the performance of the proposed improved DWA is verified through comparative experiments with the conventional DWA to assess its effectiveness in dynamic obstacle avoidance.

Case I-1 focuses on evaluating whether or not the robot can effectively avoid obstacles approaching from the side rather than directly from the front. In this experiment, the robot must avoid four obstacles descending from above while moving diagonally from the lower left to the upper right toward a designated target point, as shown in Figure 6a. All obstacles move at the same speed. Similarly, Case I-2 examines the robot's ability to avoid four obstacles moving horizontally from left to right while the robot moves vertically from bottom to top toward the target point, as shown in Figure 6b. In this case, all four obstacles also move at a uniform speed.

Thus, both Case I-1 and Case I-2 assess how effectively the robot adapts its avoidance strategy under structured obstacle movement conditions. Additionally, the comparative evaluation between the conventional and improved DWA provides insight into the advantages of incorporating dynamic obstacle behavior prediction in real-time navigation.

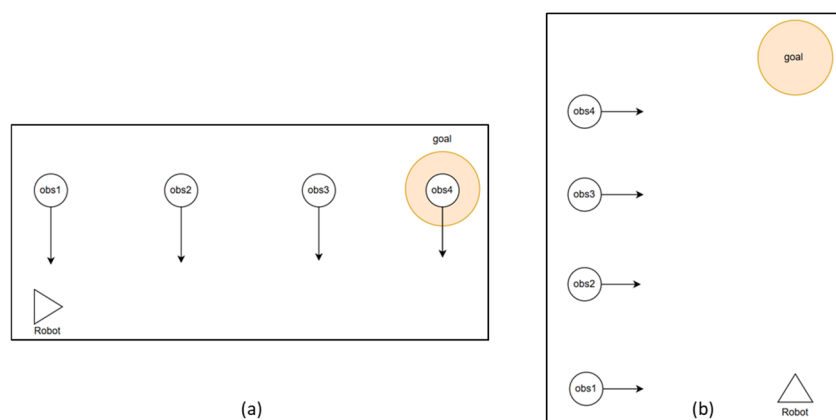


Figure 6. Experimental setup for structured path obstacle avoidance: (a) Case I-1 and (b) Case I-2.

4.2.2. Results of Case I-1

As shown in Figure 7, the experimental results of Case I-1 highlight the differences between the conventional DWA and the improved DWA in terms of obstacle avoidance, collision risk, and successful goal achievement. In this figure, the gray traces represent the obstacle trajectories, while the black trace represents the robot's trajectory. The numbers on both traces correspond to the same time step, meaning that, at a given moment, the robot and obstacles are located at their respective numbered positions. Additionally, colored regions indicate time steps where the robot and obstacles are in close proximity, marking high-risk zones where collision likelihood increases.

In Figure 7a, which represents the conventional DWA, the robot fails to reach the goal, indicating that it either cannot find a valid path or becomes trapped in an unfavorable region. Although no direct collisions occur, the robot encounters high-risk areas, such as the time step labeled "3", where it closely approaches an obstacle (highlighted in red). This suggests that the conventional DWA does not effectively anticipate moving obstacles, leading to inefficient and reactive behavior. Instead of proactively adjusting its trajectory, the robot selects a path that does not fully account for the dynamic environment, ultimately preventing it from reaching the goal.

In contrast, Figure 7b, which represents the improved DWA, demonstrates significantly better performance. The robot successfully reaches the goal, showcasing its ability to dynamically adjust its path in response to moving obstacles. Unlike the conventional DWA, the improved approach predicts obstacle movement and proactively modifies the trajectory to avoid high-risk zones. Although some close encounters with obstacles remain, as indicated by the colored regions, the robot consistently adapts its movement to prevent

collisions. The improved DWA takes a longer but safer path, prioritizing obstacle avoidance while still progressing toward the goal.

A direct comparison between the two results highlights the limitations of the conventional DWA, which fails to predict obstacle motion and struggles with effective navigation in dynamic environments. The improved DWA, by incorporating dynamic obstacle behavior prediction, allows the robot to intelligently assess risks, adjust its trajectory accordingly, and successfully reach the goal. While both methods face close encounters with obstacles, the improved DWA consistently chooses paths that minimize collision risk and ensures a more reliable and adaptive navigation strategy.

Overall, these findings emphasize the importance of predictive modeling in obstacle avoidance algorithms. The conventional DWA's assumption of static obstacles results in suboptimal path selection, whereas the improved DWA's ability to anticipate and react to dynamic obstacles enables more effective and safer navigation. This experiment confirms that integrating obstacle behavior prediction into real-time motion planning enhances both success rate and path efficiency, making it a crucial improvement for autonomous navigation in dynamic environments.

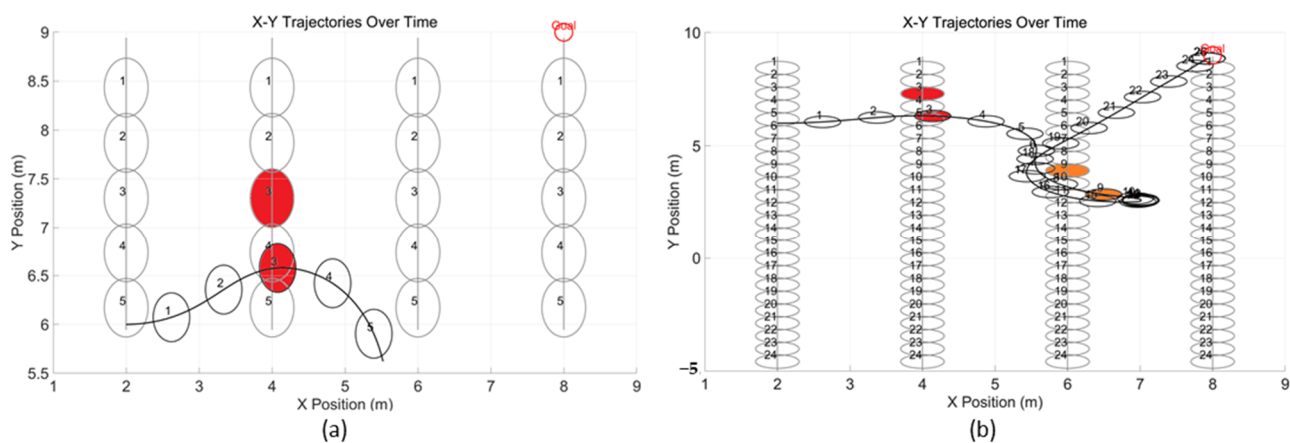


Figure 7. Comparison of robot trajectories in Case I-1 using (a) the conventional DWA and (b) the proposed method: Circled numbers represent the temporal sequence of both the robot and obstacles, allowing visualization of their motion over time. (Black: temporal trajectory of a robot, Gray: temporal trajectories of obstacles, Red and Orange: the instantaneous positions of the robot and a specific obstacle at the same time step).

4.2.3. Results of Case I-2

Figure 8 illustrates the obstacle avoidance results of Case I-2 for both the conventional and the improved DWA. Figure 8a presents the results of the conventional DWA. While the robot successfully reached the target destination, multiple instances of close encounters with obstacles occurred throughout the navigation process. The red, orange, and yellow-highlighted regions indicate moments where the robot came dangerously close to moving obstacles, increasing the risk of potential collisions. Furthermore, since the conventional DWA treats obstacles as static entities, it fails to account for their actual movement and formulates avoidance strategies based on this incorrect assumption.

As a result, the robot occasionally selected a path that moved in the same direction as the obstacles, leading to prolonged navigation times. Due to the conventional DWA's static obstacle assumption, it did not consider the obstacles' future positions, resulting in a trajectory where the robot followed the obstacles rather than avoiding them efficiently. This ineffective path selection significantly increased the total travel distance, ultimately leading to an extended goal arrival time of 42 s.

Figure 8b presents the results of the improved DWA. Similar to the conventional DWA, the robot successfully reached the target destination. However, the improved DWA significantly reduced the travel time to 20 s—less than half of that of the conventional DWA. This improvement can be attributed to the enhanced predictive capability of the improved DWA, which anticipates the movement of obstacles and proactively adjusts the trajectory. Unlike the conventional DWA, which treats obstacles as being static and reacts to their movement belatedly, the improved DWA incorporates obstacle velocity and direction into its planning, enabling it to optimize its path early and avoid unnecessary detours. Additionally, while the total number of high-risk proximity instances remains comparable between both methods, the improved DWA maintains a safer distance from obstacles during these encounters. This suggests that its predictive obstacle avoidance strategy enables the robot to maintain a more efficient and collision-free trajectory while optimizing travel time.

The findings of this experiment demonstrate that the improved DWA significantly enhances both obstacle avoidance efficiency and travel time optimization compared to the conventional DWA. The conventional DWA, due to its static obstacle assumption, sometimes results in the robot traveling alongside obstacles rather than effectively avoiding them, leading to an increase in overall travel time. In contrast, the improved DWA anticipates obstacle movements and adjusts the trajectory preemptively, optimizing efficiency and reducing unnecessary deviations. The reduction in goal arrival time from 42 s to 20 s highlights the practical advantages of integrating dynamic obstacle behavior prediction into DWA-based navigation algorithms, confirming its effectiveness in improving trajectory planning and overall navigation performance in dynamic environments.

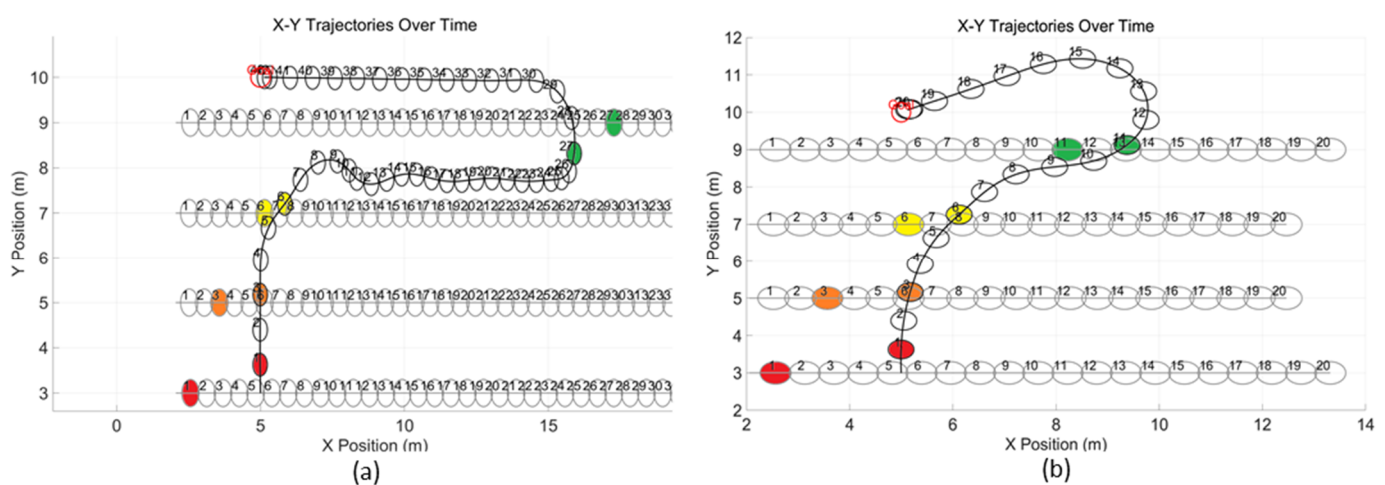


Figure 8. Comparison of robot trajectories in Case I-2 using (a) the conventional DWA and (b) the proposed method. Circled numbers represent the temporal sequence of both the robot and obstacles, allowing visualization of their motion over time. (Black: temporal trajectory of a robot, Gray: temporal trajectories of obstacles, Red and Orange: the instantaneous positions of the robot and a specific obstacle at the same time step).

4.3. Case II: Random Path Obstacle Avoidance Experiments

4.3.1. Experimental Setup for Case II

The experiment involves navigating past 12 dynamic obstacles following random paths to reach the target point, as illustrated in Figure 9. Each obstacle moves at a constant speed, but its direction is randomly determined and updated every 0.5 s. Additionally, to prevent collisions, obstacles that would otherwise overlap are programmed to move in opposite directions.

A total of 40 trials were conducted, with 20 trials using the method proposed in this paper and 20 trials using the conventional DWA. To ensure identical experimental conditions, the obstacle trajectories from the first trial were recorded, and the same recorded trajectories were used from the second to the twentieth trial to maintain consistency.

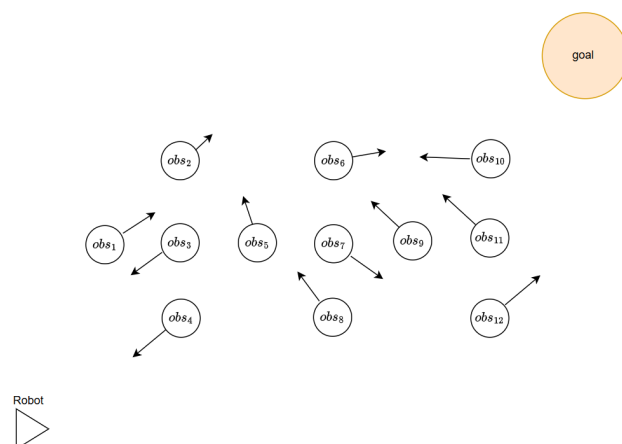


Figure 9. Experimental environment for Experiment 3.

4.3.2. Results of Case II

Figure 10 presents the results of the random obstacle avoidance experiment, comparing the performance of the conventional DWA in (a) and the improved DWA in (b). The blue markers represent the robot, the green marker represents the goal, and the red markers denote obstacles. The circled numbers indicate the temporal sequence, providing insight into the robot's movement and obstacle avoidance performance over time.

In Figure 10a, the conventional DWA exhibits multiple instances where the robot comes into close proximity with obstacles before reaching the goal. Notably, at steps 3 and 5, the robot approaches obstacles at a significantly reduced distance, increasing the likelihood of collisions. A similar pattern is observed at step 6, where the proximity between the robot and obstacles remains high, suggesting that the conventional DWA struggles to effectively avoid dynamic obstacles. This limitation arises because the conventional DWA treats obstacles as static objects, failing to account for their future movement. Consequently, the algorithm reacts to obstacles only when they are already within close range, leading to a delayed response and a higher risk of collision.

Conversely, in Figure 10b, the improved DWA demonstrates a notable reduction in collision risk under the same obstacle conditions. While steps 3 and 4 show similar avoidance behavior to the conventional DWA, a clear distinction emerges at steps 5 and 6, where the improved DWA maintains a greater minimum distance from obstacles. This indicates that the improved DWA successfully anticipates obstacle trajectories and adjusts its path preemptively, resulting in a more effective and safer avoidance strategy. The ability to predict the future positions of obstacles allows the improved DWA to implement an advanced avoidance mechanism, ensuring that the robot navigates past obstacles with minimal risk of collision.

Overall, the results demonstrate that the improved DWA significantly outperforms the conventional DWA in dynamic obstacle avoidance. While the conventional DWA exhibits limited avoidance capabilities due to its static obstacle assumption, the improved DWA effectively integrates future obstacle motion prediction, enabling more precise and proactive avoidance maneuvers. This experiment validates that the enhanced predictive capability of the improved DWA leads to higher reliability and safety in environments with randomly moving obstacles, ensuring superior obstacle avoidance performance.

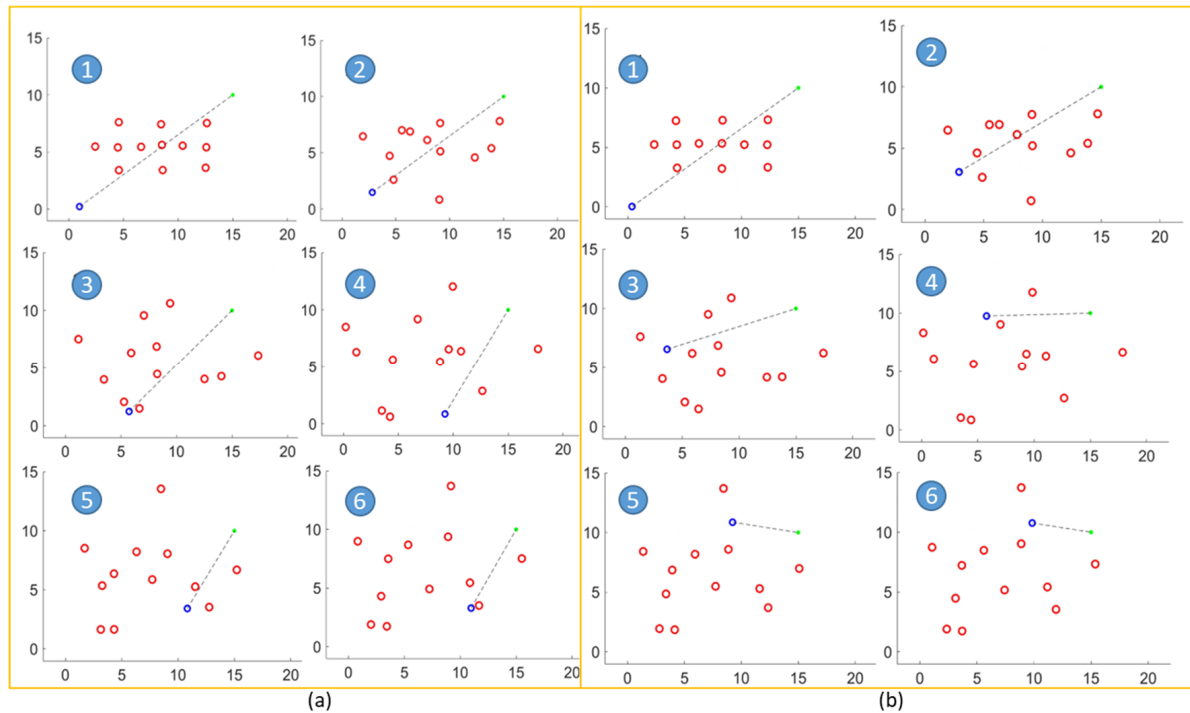


Figure 10. Trajectory comparison under random obstacle conditions in Case II using (a) the conventional DWA and (b) the proposed method. Numbers represent the temporal order of robot and obstacles, illustrating adaptive path evolution. (Blue: a robot, Red: obstacles, Green: destination).

4.3.3. Results of 20 Repeated Random Trials for Case II

The table presents the results of 20 randomized obstacle avoidance experiments comparing the conventional DWA and the improved DWA. A detailed analysis of the results reveals several key differences.

In the conventional DWA, a total of five collisions occurred during the experiments, specifically in trials 3-3, 3-5, 3-14, 3-16, and 3-20. In contrast, the improved DWA did not experience any collisions throughout the 20 trials. This demonstrates that the obstacle motion prediction capability of the improved DWA significantly enhances avoidance performance compared to the conventional model.

The collisions observed in the conventional DWA algorithm, as shown in Table 2, primarily occur due to its assumption that obstacles remain stationary during local path planning. This assumption causes the robot to generate trajectories based only on the obstacles' current positions $P_{obs}(t)$, without accounting for their future movements. As a result, when a dynamic obstacle moves into the robot's trajectory after the velocity pair (v, ω) has been selected, the robot fails to adapt in time and collides.

In contrast, the proposed method integrates a future position prediction mechanism that estimates where obstacles will move in the next time step $P_{obs}(t + 1)$, based on their current velocity and direction. This allows the robot to anticipate potential collisions and proactively adjust its path. Additionally, the use of risk-aware penalties, $J_{risk}(obs)$, $J_{count}(obs)$, further biases the evaluation toward safer trajectories by penalizing velocity candidates that lead toward dense or high-risk obstacle zones. Consequently, the robot not only avoids direct collisions but also maintains a safer buffer distance throughout navigation.

Additionally, a comparison of the average arrival times indicates that the conventional DWA recorded an average arrival time of 46.7 s, whereas the improved DWA achieved a reduced average of 36 s, reflecting a 23% decrease in travel time. This reduction suggests that the obstacle prediction feature not only improves collision avoidance but also enables more efficient path planning and motion control. Notably, in the conventional DWA, trials 3-

10 and 3-15 exhibited significantly increased arrival times of 7.52 s and 10.82 s, respectively. This suggests that the conventional DWA often required detours and frequent directional adjustments to avoid obstacles, leading to prolonged travel times. In contrast, the improved DWA maintained a consistent and shorter travel time, with the longest recorded time being 53.4 s (trial 3-12).

These findings confirm that the improved DWA with obstacle behavior prediction effectively reduces collision risk and optimizes the travel path, leading to faster arrival times. More importantly, in highly dynamic environments with randomly moving obstacles, the improved DWA demonstrates superior stability and efficiency compared to the conventional model, reinforcing its practical applicability in real-world autonomous navigation scenarios.

As discussed in the results of Cases I and II, the robot exhibited context-aware behavior by selecting safer paths when direct routes were obstructed. To ensure that the robot maintains a balance between goal reachability and obstacle avoidance in complex scenarios, the proposed risk assessment strategy integrates both directional penalty and spatial density analysis. The term J_{risk} assigns a piecewise penalty based on the relative angle and distance of obstacles within the robot's field of view. This penalizes forward directions with high obstacle concentration, encouraging safer paths. Additionally, J_{count} quantifies the number of obstacles on the left and right sides of the robot, guiding the robot to steer toward the side with lower obstacle density.

These risk-based terms are integrated with the goal-oriented heading, clearance, and future prediction terms in the overall evaluation function. As a result, the robot does not simply pursue the shortest or most direct path, but instead dynamically adjusts its trajectory to prioritize safety when necessary.

The simulation results presented in Case I and Case II illustrate that the proposed method is capable of adaptive trade-offs. For example, when dense or fast-moving obstacles block the most direct path, the robot selects a longer but safer trajectory that still ensures timely arrival. This demonstrates that the proposed approach effectively balances proactive avoidance and target-directed navigation under varying levels of environmental complexity.

Table 2. Collision occurrence and arrival times for the destination.

Trial Number	Conventional DWA (s)	Improved DWA (s)
3-1	49.9	28.8
3-2	33	33.6
3-3	Collision	48.3
3-4	42.2	32.6
3-5	Collision	34
3-6	33.2	33
3-7	52.4	42.1
3-8	39.5	37.9
3-9	33.3	32.2
3-10	75.2	23.6
3-11	38.5	27.7
3-12	42.3	53.4
3-13	34.3	31.6
3-14	Collision	39.9
3-15	108.2	47.1
3-16	Collision	44.5
3-17	35.6	31.7
3-18	41.6	51.7
3-19	41.2	32
3-20	Collision	34.2

5. Conclusions

In contrast to conventional and recently extended DWA-based methods, our approach combines predictive obstacle modeling with a novel risk-aware evaluation strategy that emphasizes future interaction space. By incorporating forward-looking metrics such as $J_{future(obs)}$, $J_{risk(obs)}$, and $J_{count(obs)}$, the proposed method demonstrates enhanced adaptability to high-density dynamic environments. This contribution offers a new perspective on path optimization by explicitly predicting short-term obstacle behavior and redefining the core structure of the DWA evaluation function.

This paper proposed an enhanced Dynamic Window Approach (DWA) incorporating dynamic obstacle behavior prediction to improve real-time path planning and obstacle avoidance in environments with moving obstacles. The conventional DWA assumes static obstacles, making it challenging to accurately predict the movements of dynamic obstacles. In contrast, the proposed method integrates a prediction mechanism to estimate future obstacle positions, enabling proactive obstacle avoidance. By incorporating trajectory prediction and adaptive risk assessment, the proposed approach enhances both safety and navigation efficiency.

To validate the effectiveness of the proposed method, extensive simulation experiments were conducted in both structured obstacle environments and randomized dynamic obstacle scenarios. The results demonstrated that the enhanced DWA effectively avoided obstacles while significantly reducing arrival time compared to the conventional DWA. In randomized obstacle environments, the proposed approach consistently exhibited a lower collision rate and shorter arrival times. Specifically, across 20 repeated trials, the conventional DWA resulted in five collisions, whereas the enhanced DWA successfully navigated all trials without collisions. Additionally, the average arrival time was reduced by 23%, further proving its efficiency in dynamic environments.

These findings confirm that integrating dynamic obstacle behavior prediction into DWA significantly improves real-time obstacle avoidance performance. By anticipating obstacle movements, the proposed method reduces unnecessary path deviations and generates smoother, more adaptive trajectories. This capability is particularly valuable for applications such as autonomous delivery robots, warehouse automation, and service robots operating in crowded environments.

In conclusion, the proposed enhanced Dynamic Window Approach (DWA) incorporating dynamic obstacle behavior prediction effectively addresses key limitations of the conventional DWA, particularly in environments with moving obstacles. By integrating future obstacle prediction and a risk-aware evaluation strategy, the method demonstrates significant improvements in collision avoidance performance and navigation efficiency, as confirmed through structured and randomized simulation experiments.

The proposed method assumes that dynamic obstacles maintain a constant velocity and heading over short prediction intervals. While this linear approximation enables efficient real-time computation, it may not fully capture abrupt changes in obstacle behavior, such as acceleration or direction reversal. These limitations may reduce prediction accuracy and, in turn, affect avoidance performance in highly dynamic settings. Nevertheless, this assumption is generally reasonable within short prediction intervals, where obstacle motion tends to be locally linear. Although obstacles may exhibit non-linear or erratic behavior over longer durations, their short-term trajectories can often be approximated as uniform. This trade-off provides a balance between computational simplicity and prediction reliability, making the approach suitable for real-time applications.

To further improve robustness and adaptability, future work will explore the integration of probabilistic or learning-based motion prediction models capable of representing more diverse and non-deterministic behaviors. Additional extensions to the path planning

strategy will target complex environmental constraints such as narrow corridors or uneven terrain. Finally, real-world validation using physical robot platforms will be conducted to evaluate the practical applicability of the proposed approach under dynamic and uncertain conditions.

We acknowledge that learning-based approaches have shown promising results in specific contexts. However, in this study, we prioritize immediate responsiveness, interpretability, and deterministic control—attributes that are critical in real-world autonomous systems. Future research may extend this framework to hybrid models that incorporate learning-based prediction or integrate with global planners for broader applicability.

Funding: This work was supported by the Hongik University new faculty research support fund.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The author would like to express his gratitude to Seung Heon Lee at Kyungil University, Korea, for his valuable support in setting up the experimental framework.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Borenstein, J.; Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [\[CrossRef\]](#)
2. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1995**, *5*, 90–98. [\[CrossRef\]](#)
3. Lumelsky, V.; Skewis, T. Incorporating range sensing in the robot navigation function. *IEEE Trans. Syst. Man Cybern.* **1990**, *20*, 1058–1068. [\[CrossRef\]](#)
4. Chen, P.; Pei, J.; Lu, W.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [\[CrossRef\]](#)
5. Yousfi, N.; Rekik, C.; Jallouli, M.; Derbel, N. Optimized fuzzy controller for mobile robot navigation in a cluttered environment. In Proceedings of the 2010 7th International Multi-Conference on Systems, Signals and Devices, Amman, Jordan, 27–30 June 2010.
6. Jones, M.; Djahel, S.; Welsh, K. Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–39. [\[CrossRef\]](#)
7. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [\[CrossRef\]](#)
8. Fiorini, P.; Shiller, Z. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [\[CrossRef\]](#)
9. Wilkie, D.; Berg, J.; Manocha, D. Generalized velocity obstacles. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009.
10. Berg, J.; Snape, J.; Guy, S.J.; Manocha, D. Reciprocal collision avoidance with acceleration-velocity obstacles. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
11. Ko, N.Y.; Simmons, R.G. The lane-curvature method for local obstacle avoidance. In Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications, Victoria, BC, Canada, 17 October 1998.
12. Arras, K.O.; Persson, J.; Tomatis, N.; Siegwart, R. Real-Time Obstacle Avoidance for Polygonal Robots with a Reduced Dynamic Window. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002.
13. Wu, C.; Yu, W.; Li, G.; Liao, W. Deep reinforcement learning with dynamic window approach based collision avoidance path planning for maritime autonomous surface ships. *Ocean Eng.* **2023**, *284*, 115208. [\[CrossRef\]](#)
14. Hong, Z.; Chun-Long, S.; Zi-Jun, Z.; Wei, A.; De-Qiang, Z.; Jing-Jing, W. A modified dynamic window approach to obstacle avoidance combined with fuzzy logic. In Proceedings of the 2015 14th International Symposium Distributed Computing and Applications for Business Engineering and Science, Guiyang, China, 18–24 August 2015.
15. Kim, J.; Yang, G.-H. Improvement of Dynamic Window Approach using Reinforcement Learning in Dynamic Environments. *Int. J. Control Autom. Syst.* **2022**, *20*, 2983–2992. [\[CrossRef\]](#)
16. Missura, M.; Bennewitz, M. Predictive collision avoidance for the dynamic window approach. In Proceedings of the 2019 International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019.

17. Guan, M.; Wen, C.; Wei, Z.; Cheng, L.; Zou, Y. A Dynamic Window Approach with Collision Suppression Cone for Avoidance of Moving Obstacles. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018.
18. Cheng, C.; Zhang, H.; Sun, Y.; Tao, H.; Chen, Y. A Cross-platform deep reinforcement learning model for autonomous navigation without global information in different scenes. *Control Eng. Pract.* **2024**, *150*, 105991. [[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.