ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

# DO-MPPI: Improved Model Predictive Path Integral Path Planning with Dynamic Obstacle Evaluation for Mobile Robots

Peng Deng, Junmin Li*, Zhonghua Xu

School of Mechanical Engineering, Xihua University, Chengdu, Sichuan 610039, China

*Corresponding author: Junmin Li

## Abstract

**To address the problems of delayed obstacle avoidance response and insufficient dynamic safety margin of traditional local path planning algorithms in complex dynamic environments, this paper proposes an improved Model Predictive Path Integral (MPPI) path planning algorithm fused with a dynamic obstacle evaluation mechanism, named DO-MPPI. The method introduces a Kalman filter-based obstacle motion state estimation and temporal position prediction module into the MPPI sampling framework, constructs a spatio-temporal joint collision detection model, and automatically eliminates sampled trajectories with collision risk through a high-weight collision penalty, thereby achieving proactive avoidance of dynamic obstacles while retaining the global optimization ability of random sampling. Simulation experiments are carried out in purely static, low-speed same-direction, and high-speed crossing dynamic obstacle scenarios, and physical experiments are conducted on a self-developed four-wheel differentially driven firefighting inspection robot platform. The results show that, compared with traditional MPC, standard MPPI, and DWA algorithms, the proposed DO-MPPI algorithm significantly reduces the collision rate, improves the arrival success rate, maintains a larger minimum safety distance, and shortens the obstacle avoidance response time, demonstrating superior proactive obstacle avoidance performance in complex dynamic environments.**

## Keywords

**Model Predictive Path Integral (MPPI); path planning; dynamic obstacle avoidance; mobile robot; Kalman filter; firefighting inspection robot.**

## 1. INTRODUCTION

With the intelligent upgrading of smart manufacturing, firefighting emergency response, and public safety, the autonomous navigation capability of mobile robots has become a key indicator of their engineering practicality. In typical scenarios such as indoor inspection, warehouse logistics, and human-robot coexistence, robots must not only cope with static obstacles but also respond in real time to the uncertainty caused by dynamic obstacles such as moving personnel and transported equipment. How to achieve efficient and smooth real-time obstacle avoidance while ensuring a sufficient safety margin has long been a core problem in the field of mobile robot path planning.

To address the above problem, scholars at home and abroad have proposed a large number of local path planning and obstacle avoidance methods. Early methods are represented by the Artificial Potential Field (APF) approach, which guides robot motion by constructing attractive and repulsive fields. It is computationally simple and responds quickly, but it tends to fall into local minima and may fail to reach the goal[1]. The Dynamic Window Approach (DWA) samples and evaluates trajectories in velocity space and can handle kinematic constraints well, but its search is local in complex dynamic environments and global optimality is difficult to guarantee[2]. Sampling-based methods, such as rapidly exploring random trees (RRT, RRT*) and their variants, construct a search tree through random sampling and possess probabilistic completeness and asymptotic optimality; they are widely used for global path planning, but in dynamic environments they usually need to be combined with a local planner[3-4]. Model Predictive Control (MPC) handles constraints by rolling optimization of future control sequences, but its solution depends on the derivative information of the dynamics model and the cost function, resulting in heavy computation and limited real-time performance[5].

In recent years, learning-based methods represented by Deep Reinforcement Learning (DRL) have provided new ideas for dynamic obstacle avoidance. Such methods train policy networks through a large number of interactions, enabling robots to make end-to-end obstacle avoidance decisions in mapless or partially observable conditions[6-7], and they exhibit higher avoidance success rates than traditional reactive methods in pedestrian-dense environments[8]. However, learning-based methods usually depend on the quality of training in simulation, have limited generalization when facing complex random motions outside the training distribution, and it is difficult to provide strict safety guarantees.

Model Predictive Path Integral (MPPI) control is a sampling-based model predictive control method rooted in the information-theoretic framework. By sampling a large number of trajectories in parallel and taking a cost-weighted average, it does not require gradients of the dynamics or the cost function, making it naturally suitable for real-time control under nonlinear and non-convex constraints[9]. To improve its safety and environmental adaptability, researchers have successively proposed sampling enhancement methods integrated with control barrier functions, the MPPI-DBaS algorithm combined with discrete barrier states, and adaptive MPPI path planning methods for complex dynamic environments. In terms of global-local coordination, the RRT*-guided MPPI path planning framework uses the global path as prior guidance for local sampling, effectively balancing global optimality and dynamic responsiveness; MPPI has also been successfully applied to different platforms, such as path tracking of tracked harvesters and reactive motion planning of humanoid robots.

Aiming at the autonomous navigation requirements of firefighting inspection robots in complex dynamic environments, this paper proposes an improved MPPI path planning algorithm fused with a dynamic obstacle evaluation mechanism (DO-MPPI). The method introduces obstacle motion state estimation and temporal position prediction based on a Kalman filter into the traditional MPPI sampling framework, constructs a spatio-temporal joint collision detection model, and automatically eliminates sampled trajectories with collision risk through a high-weight collision penalty, thereby combining the global optimization ability of full random sampling with the proactive prediction ability of dynamic obstacle avoidance. The remainder of this paper is organized as follows: Chapter 2 describes the model design of the DO-MPPI algorithm and the construction of the dynamic obstacle function; Chapter 3 verifies the effectiveness of the algorithm through simulation and physical experiments in terms of collision rate, arrival success rate, minimum safety distance, and obstacle avoidance response time; Chapter 4 summarizes the paper and discusses future work.

## 2. METHODS AND MATERIALS

### 2.1. Model Predictive Path Integral Algorithm

Model Predictive Path Integral (MPPI) is a sampling-based model predictive control method rooted in the information-theoretic framework. Owing to the inherent advantages of stochastic optimization and natural parallel computation, combined with an improved dynamic obstacle evaluation mechanism, it enables a robot to adapt agilely to complex dynamic environments.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

The algorithm follows a core execution logic of "sample-predict-select": within a finite prediction horizon it generates a large number of random motion trajectories in parallel and approximates the optimal control distribution through the statistical distribution properties of the samples. For a four-wheel differentially driven firefighting robot, the trajectory generation strictly obeys the differential kinematics constraints of the chassis, ensuring that every planned trajectory is physically executable.

From a theoretical perspective, the MPPI algorithm is built on the intersection of stochastic optimal control and information theory, relying on the intrinsic connection between free energy and the KL divergence. Unlike traditional MPC, which relies on linearization of the dynamics or convex optimization, MPPI transforms the control problem into finding an optimal control distribution $\mathbb{Q}^*$, which minimizes the KL divergence with respect to a baseline control distribution $\mathrm{P}$.

Let the state of the robot at time $t$ be $x_t$, and the control input be $v_t$. For the motion control characteristics of the four-wheel differentially driven robot, the control input $v_t$ is modeled as a random variable perturbed by Gaussian white noise, and its mathematical expression is as follows:

$$v_t \sim \mathrm{N}\,(u_t, \Sigma)$$

where $u_t$ is the nominal control mean to be optimized, and $\Sigma$ is the control noise covariance matrix. The discrete-time state transition equation of the system is:

$$x_{t+1} = F(x_t, v_t)$$

Based on the information-theoretic derivation, the update law of the optimal control sequence can be obtained by the weighted average of a large number of randomly sampled trajectories, and the weight is exponentially related to the total cost of the trajectory. The specific implementation consists of the following steps:

(1) Parallel Trajectory Sampling

Using the parallel computing capability of multi-core CPUs, the nominal control sequence $U^* = \{u_0, \ldots, u_{T-1}\}$ is superimposed with disturbance noise $\delta u_{t,k} \sim \mathrm{N}\,(0, \Sigma)$ to generate $K$ candidate control sequences. Then, forward integration is performed based on the nonlinear kinematics model $F(\cdot)$ of the four-wheel differentially driven chassis to derive $K$ predicted trajectories . $\tau_k = \{x_{0,k}, \ldots, x_{T,k}\}$ This sampling process does not require additional gradient information of the dynamics, so it naturally has good adaptability to complex environmental constraints such as non-smooth and non-convex conditions.

(2) Trajectory Cost Evaluation

The total cost of the k-th trajectory $\tau_k$ is calculated as $S(\tau_k)$. The design of the cost function is not single-dimensional: it not only contains the core path tracking error penalty and the static obstacle avoidance cost in traditional navigation tasks, but also innovatively incorporates a dynamic obstacle avoidance improvement term specially designed for dynamic environments, so as to achieve a comprehensive evaluation of complex scenes.

(3) Optimal Control Update

According to the free-energy principle, the update formula of the optimal control input obtained in this study is:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

$$u_0^* = u_0 + \frac{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} S(\tau_k)\right) \delta u_{0,k}}{\sum_{k=1}^{K} \exp\left(-\frac{1}{\lambda} S(\tau_k)\right)}$$

In the formula, $\lambda$ is the temperature parameter used to adjust the exploration depth in the random sampling process. A smaller $\lambda$ makes the weights highly concentrated on low-cost high-quality trajectories, whereas a larger $\lambda$ tends to average all sampled trajectories. The updated control sequence is smoothed by a Savitzky-Golay filter, and the first step of the control quantity is finally output as the current velocity command.

## 2.2. Improved Algorithm: DO-MPPI

In MPPI trajectory planning in dynamic environments, balancing proactive risk perception and global trajectory optimality is a core difficulty, as is how to strengthen dynamic obstacle avoidance while maintaining global sampling coverage and using obstacle motion prediction to guide efficient convergence of the algorithm. The original MPPI algorithm excels at global optimization through full random sampling and importance weighting, but its obstacle avoidance relies on passive response of the cost terms and is prone to insufficient safety margin in highly dynamic scenarios because of delayed risk perception. In contrast, velocity-obstacle-type proactive obstacle avoidance methods focus on front-loaded risk avoidance; they are constraint-based planning strategies built on motion prediction that screen feasible control quantities by clipping the collision region in advance, but overly strong constraints tend to shrink the sampling space and cause the algorithm to fall into local suboptimal solutions.[10]

To enable MPPI to perceive and avoid obstacles proactively in dynamic environments, this paper proposes an improved DO-MPPI algorithm. The strategy introduces a dynamic obstacle processing module into the traditional MPPI framework, aiming to improve the environmental adaptability of the robot in complex systems: it performs broad sampling exploration in scenarios with high uncertainty, focuses on path refinement in known low-cost regions, and introduces a strong perturbation mechanism to escape from local stagnation.

## 2.3. Dynamic Obstacle Function

The dynamic obstacle function is the core module of the improved MPPI algorithm. Its role is to transform the proactive risk prediction idea of velocity obstacles into a cost form that can be weighted and integrated within the MPPI framework, thereby realizing a trade-off between "global sampling exploration" and "dynamic risk avoidance". The total cost function $S(\tau_k)$ is defined as the weighted accumulation of the costs of all state points within the prediction horizon $T$:

$$S(\tau_k) = \sum_{t=0}^{T-1} \left( J_{static}\left(x_{t,k}\right) + J_{dynamic}\left(x_{t,k}\right) \right) + \phi(x_{T,k})$$

(1) Basic Navigation Evaluation Indicators $J_{static}$

The core design goal of this part of the evaluation indicators is to ensure that the robot can continuously, smoothly and efficiently track the global reference path in static environments. The indicators are generally composed of multiple sub-items combined by weighted summation:

The path tracking cost is used to measure the deviation of the planned trajectory from the reference path, and specifically includes three key indicators: lateral deviation (PathAlign, $w = 3.0$), longitudinal progress (PathFollow, $w = 5.0$), and heading angle error (PathAngle, $w = 4.0$). In addition, this study introduces a forward motion incentive term (PreferForward,

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

$w=7.0$ ), which forces the robot to prefer forward motion when no obstacle avoidance is required, avoiding hesitation or unnecessary stopping during path tracking.

Goal approach cost: an activation threshold of 1.0 m is set, and this term is activated only when the robot enters the range of 1.0 m around the local goal point. By constraining the Euclidean distance and the relative heading angle between the robot and the goal point, this term guides the robot pose to converge gradually and finally achieve precise docking.

Static environment safety cost: calculated based on the real-time updated local cost map, a repulsive field model is used to quantify the risk of trajectories passing through the static obstacle region. A relatively high weight coefficient $w=20.0$ is assigned to this term; once a generated trajectory touches the inflated layer of a static obstacle, a huge penalty cost is immediately imposed, fundamentally preventing static collisions.

Kinematic constraint cost: used to strictly limit the linear velocity $v_x \in [-0.1, 0.8] m/s$ and the angular velocity $\omega_z \in [-1.0, 1.0] rad/s$ of the robot, ensuring that the control commands output by the navigation layer always satisfy the physical performance boundaries of the actuator, avoiding motion loss of control caused by command overshoot.

(2) Dynamic Obstacle Prediction and Avoidance $J_{dynamic}$

In view of the inherent limitation that the traditional cost map can only represent static environment information and cannot characterize the motion trend of obstacles, this study designs a dedicated DynamicObstaclesCritic evaluation function to handle the dynamic obstacle avoidance task. Based on the set of dynamic obstacle states output by the upstream Kalman filter module,$O = \{(p_{obs}^i, v_{obs}^i, r_{obs}^i)\}$ , this function carries out forward-looking collision detection and risk assessment in the time dimension.[11]

For the $k$ -th trajectory at time $t$, the predicted pose of the robot is $\mathbf{x}_{t,k}$ . The algorithm first uses a linear motion model to calculate the predicted position of the $i$-th obstacle at the same time, denoted as $\hat{p}_{obs}^i(t)$ :

$$\hat{p}_{obs}^i(t) = p_{obs}^i(0) + v_{obs}^i \cdot (t \cdot \Delta t)$$

After completing the position prediction, this study constructs a dynamic collision detection model: each predicted obstacle is equivalent to a circular forbidden zone with radius $R_{safe} = r_{obs}^i + \delta_{padding}$ , and the robot body is simplified as a rectangular bounding box with orientation attributes. Through coordinate transformation, the shortest distance from the obstacle center to the edge of the robot bounding box is accurately calculated; if this distance is less than zero, a spatio-temporal collision at that moment is determined and a collision penalty $C_{collision} = 100000.0$ is immediately applied to the trajectory. The overall calculation form is shown in Eq. (8):

$$J_{dynamic}(x_{t,k}) = w_{dyn} \cdot \left( \mathrm{I}_{collision}(x_{t,k}, \hat{p}_{obs}^i(t)) \cdot C_{collision} \right)$$

In the system constructed in this study, the dynamic obstacle avoidance weight $W_{dyn}$ is set to 10.0. Since the value of $C_{collision}$ is far larger than the conventional cost terms such as path tracking and static safety, the total cost of trajectories with dynamic collision risk tends to infinity and the corresponding sampling weight approaches zero. This mechanism enables the MPPI algorithm to autonomously generate smooth avoidance maneuvers such as deceleration or detour without triggering global path replanning, thus achieving proactive adaptation to dynamic inspection environments.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

## 2.4. Simulation Experiments

To verify the performance of the proposed algorithm, simulation experiments were carried out. The experimental hardware platform was configured with an AMD Ryzen 7 9800X3D processor, 32 GB memory, and an NVIDIA RTX 5080 GPU; the software environment was based on Ubuntu 20.04 with the ROS2 operating system and Python 3.13, and the simulations were completed in the Visual Studio Code integrated development environment. The specific simulation parameters are listed in Table 1.

In terms of cost function settings, all algorithms based on the model predictive path integral architecture, including standard MPPI, o-MPPI, pi-MPPI, R-MPPI, and the proposed DO-MPPI algorithm, adopted identical running cost and terminal cost functions; CPU parallel acceleration was also used, and the number of trajectories, the temperature parameter, and the dynamic obstacle weight remained constant in all comparison experiments to exclude the interference of weight differences on performance evaluation.[12-15]

## 2.5. Experimental Platform

This study independently built a four-wheel differentially driven firefighting inspection robot chassis prototype for complex indoor environments. The hardware system adopts a dual-core layered architecture of "positioning and navigation unit + low-level control unit": the positioning and navigation unit is equipped with the domestic high-performance edge computing platform RK3588 and runs the ROS2 Galactic operating system, undertaking high-computing-load tasks such as SlamToolbox graph optimization mapping, deep vision-LiDAR heterogeneous data fusion, FBOW visual relocalization, and DO-MPPI path planning. The low-level control unit uses an STM32F407 microcontroller, responsible for millisecond-level real-time tasks, including high-frequency data acquisition of wheel-hub motor encoders and the MPU6050 inertial measurement unit, Kalman filter preprocessing, and wheel-hub motor control signal output.

In terms of sensor configuration, a SLAMTEC RPLIDAR A2 single-line LiDAR is horizontally installed on the top of the robot as the main sensor for 2D mapping and obstacle avoidance; an Orbbec Gemini2 depth camera is configured in front of the vehicle body, providing high-resolution depth and RGB images, compensating for the vertical blind zone of the single-line LiDAR while supporting AprilTag and bag-of-words model visual matching. All sensors are fixed through rigid structures, and spatial TF joint calibration and timestamp synchronization are completed at the software layer. The physical layout is shown in Figure 2.

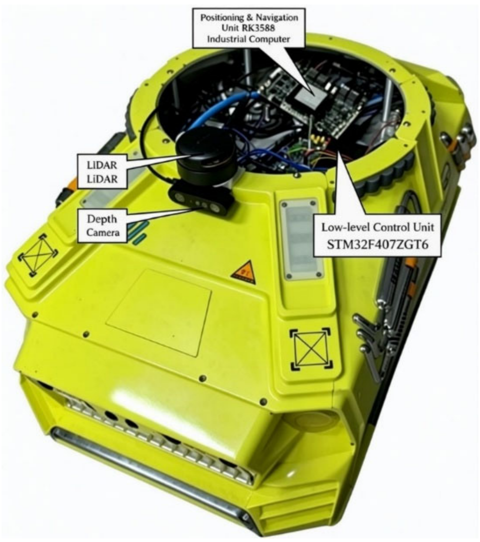



**Figure 2.** Physical prototype and sensor layout of the firefighting inspection robot

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

**Table 1.** Simulation parameters

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Prediction horizon steps | $N$ | 30 | - |
| Control period | $dt$ | 0.05 | s |
| Number of sampled trajectories | $K$ | 2000 | - |
| Temperature parameter | $\lambda$ | 0.1 | - |
| Dynamic obstacle penalty weight | $w_{dyn}$ | 10 | - |

# 3. RESULTS AND DISCUSSION

## 3.1. Performance Comparison of Algorithms in Different Scenarios

To comprehensively evaluate the performance of the improved MPPI algorithm under different environmental complexities, three representative simulation scenarios were designed: a purely static obstacle scenario, a low-speed same-direction dynamic obstacle scenario, and a high-speed crossing dynamic obstacle scenario. In each scenario, each algorithm was run independently 50 times, and the average values were taken as the final results. The evaluation indicators include collision rate, average tracking error (ATE), average time per step, arrival success rate, and minimum safety distance.

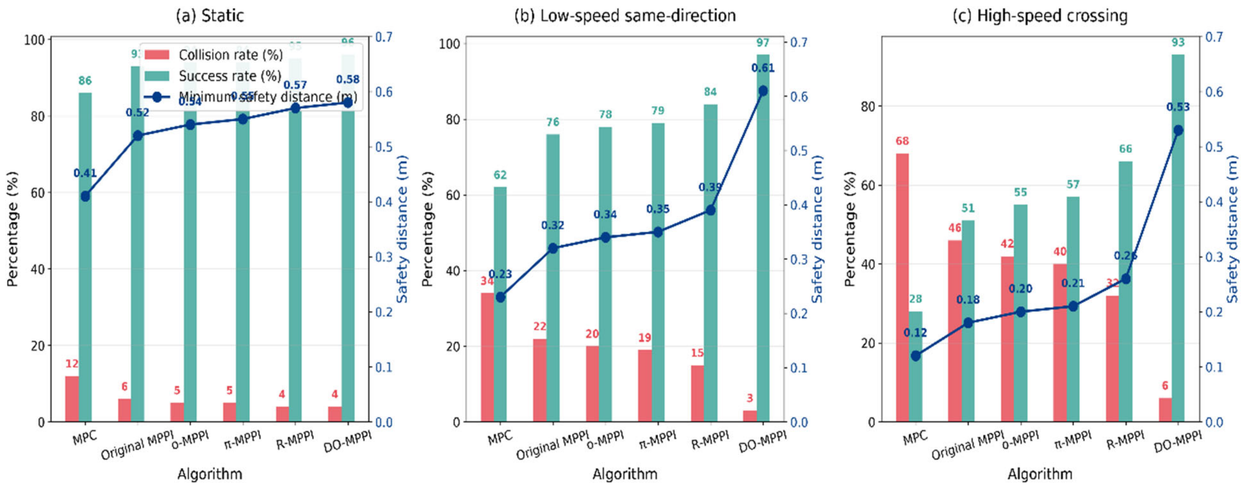



**Figure 3.** Comparison of simulation results

The simulation data show that in the purely static environment all algorithms can complete basic navigation, and the proposed algorithm achieves a 95.67% arrival success rate and a 0.58 m minimum safety distance, showing excellent optimization stability.

As the environmental complexity increases, the ability of traditional algorithms to cope with dynamic risk decreases significantly. In the low-speed same-direction dynamic obstacle scenario, the collision rate of traditional MPC rises to 34% and its success rate drops to 62%, while the collision rate of the proposed algorithm is only 3% and the success rate is as high as 97%.

In the most challenging high-speed crossing dynamic scenario, the probability of MPC falling into local minima increases significantly, with the collision rate soaring to 68% (success rate falling to 28%); the collision rate of basic MPPI also reaches 46%. In contrast, benefiting from the proactive risk prediction mechanism introduced by the dynamic obstacle evaluation function, the proposed algorithm still maintains a high success rate of 93% and a collision rate of only 6% in the extreme scenario.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009

## 3.2. Physical Experiment Verification

This experiment uses a traditional local path planning algorithm as the baseline to verify the spatio-temporal prediction ability of the DO-MPPI algorithm for dynamic obstacle trajectories, that is, whether it can execute deceleration or detour decisions before the obstacle intrudes into the planned path, realizing proactive safety avoidance.

The experiment was carried out in a factory indoor scene. The robot traveled in a straight line from start point A to end point B, with a distance of about 10 m. An experimenter walked across from the lateral front of the robot's path at about 1.0 m/s to simulate a dynamic moving obstacle. The experimental environment is shown in Fi

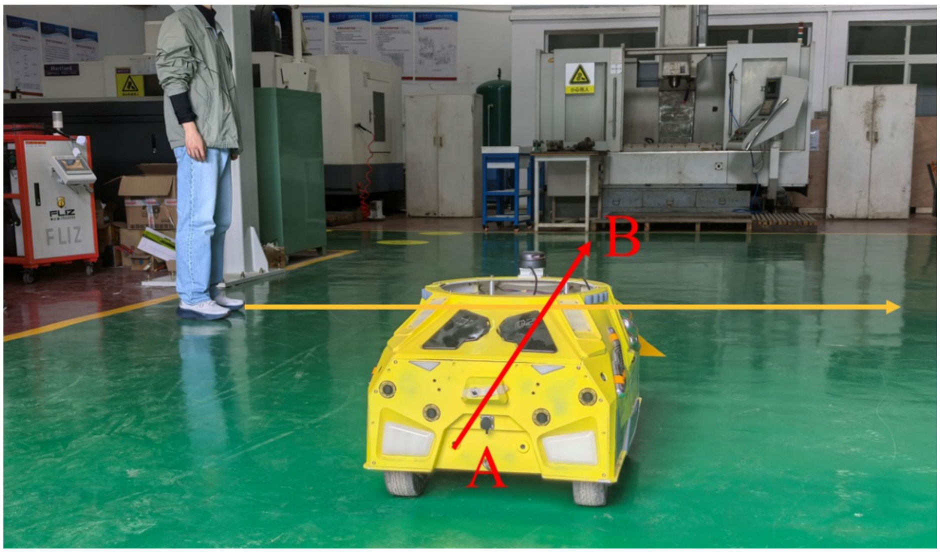
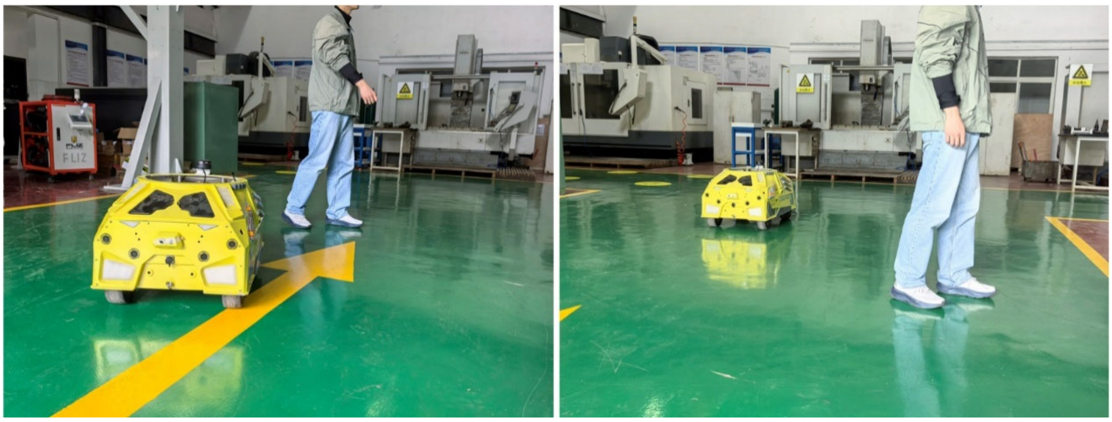



**Figure 4.** Experimental environment for active avoidance of dynamic obstacles

The experiment was performed according to the following steps:

(1) Switch the local planner to the standard DWA algorithm and complete the above crossing task;

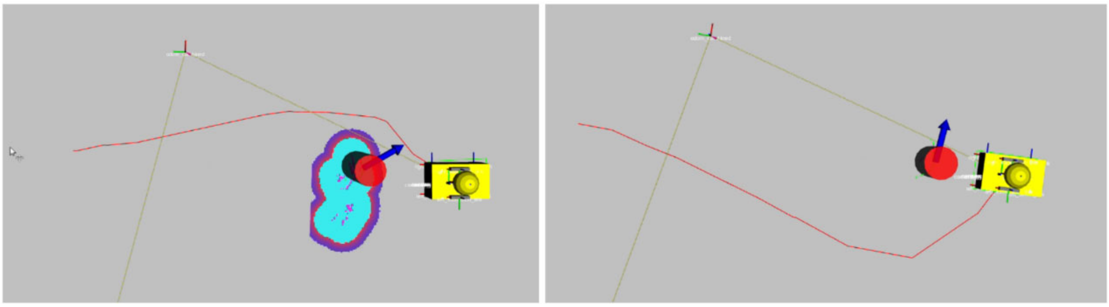
(2) Switch to the proposed DO-MPPI algorithm, enable the DynamicObstacleTracker node and the DynamicObstaclesCritic evaluation function, and repeat the experiment in the same scene.



(a) Obstacle avoidance in progress (b) Obstacle avoidance completed

**Figure 5.** Obstacle avoidance process of the DO-MPPI algorithm

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202609_12(9).0009



(a) DWA path (b) DO-MPPI path

**Figure 6.** Comparison of path planning between DWA and DO-MPPI

The obstacle avoidance process of the DO-MPPI algorithm is shown in Figure 5, and the path planning comparison results of the two algorithms are shown in Figure 6. From the path shapes, the dynamic tracking module of DO-MPPI can predict the pedestrian motion trend and determine in advance that there is a collision risk with the future trajectory of the robot. The controller then adjusts the control quantity accordingly, making the robot decelerate slowly and adjust its heading slightly in the opposite direction of the pedestrian's motion, completing the detour from behind the pedestrian. In contrast, the DWA algorithm only evaluates the cost based on the current obstacle position, lacks prediction of future states, and can only trigger a passive response when danger approaches.

Relying on the velocity vector output by the dynamic tracking module and the spatio-temporal collision detection mechanism of the dynamic obstacle evaluation function, the DO-MPPI algorithm maps the future motion trajectory of obstacles into the cost space. When a collision risk is predicted at a future moment, the corresponding high-cost trajectories are automatically eliminated, and the algorithm naturally converges to the optimal solution of decelerating and yielding or detouring in advance. The experimental results show that the algorithm effectively solves the obstacle avoidance lag problem in dynamic environments and significantly improves the traffic safety in human-robot coexistence scenarios.

To further quantify the response agility of the algorithm to sudden dynamic obstacles, this study collected the running logs of the low-level control cycle, used the native ROS2 DWA algorithm as the control baseline, extracted the core spatio-temporal indicators of the dynamic obstacle avoidance process, and compared them. The quantitative results are shown in the table below.

**Table 2.** Comparison of dynamic obstacle avoidance response parameters of local planners

| Local planning algorithm | Obstacle avoidance response time (s) | Human-robot minimum safety distance (m) | Avoidance strategy mode |
|---|---|---|---|
| DWA (baseline) | 0.85~1.12 | 0.28 | Passive reactive |
| DO-MPPI (proposed) | 0.15~0.22 | 0.85 | Proactive predictive |

As can be seen from Table 2, the obstacle avoidance response time of the traditional DWA algorithm is about 1 s. It can only trigger a passive response after the obstacle intrudes into the cost map, resulting in the human-robot minimum safety distance being compressed to 0.28 m, which is in the danger zone. In contrast, thanks to the time-domain look-ahead prediction of the Kalman tracking module, the decision response time of the proposed DO-MPPI algorithm is only about 0.2 s. The robot can plan a detour path away from the pedestrian in advance, and the human-robot minimum safety distance is stably maintained at 0.85 m. The quantitative data further verify the significant advantage of the DO-MPPI algorithm over the traditional framework in dynamic safe passage ability.

## 4. CONCLUSION

Aiming at the path planning problem of mobile robots in complex dynamic environments, this paper proposes a DO-MPPI path planning algorithm fused with a dynamic obstacle evaluation mechanism, which is intended to solve the limitations of the traditional MPPI algorithm, such as lagging obstacle avoidance response, insufficient dynamic safety margin, and passive obstacle avoidance of local planners. The main contributions of this study are as follows:

a) A dynamic obstacle evaluation function, DynamicObstaclesCritic, is designed. Based on a Kalman filter, the motion state estimation and temporal position prediction of dynamic obstacles are carried out, and a spatio-temporal joint collision detection model is constructed, integrating the proactive risk prediction mechanism into the cost evaluation system of MPPI and realizing the transformation from passive response to proactive avoidance.

b) An improved MPPI control strategy for dynamic environments is proposed. A dynamic obstacle cost term is introduced into the traditional MPPI sampling framework. Through a high-weight collision penalty, risky trajectories are automatically eliminated while the global optimization ability of the original full random sampling algorithm is retained, achieving an effective balance between dynamic obstacle avoidance performance and path optimality.

It can be seen from the simulation and physical experiments that, in multiple scenarios such as purely static, low-speed same-direction, and high-speed crossing conditions, the proposed DO-MPPI algorithm outperforms traditional MPPI and DWA algorithms in terms of collision rate, arrival success rate, minimum safety distance, and obstacle avoidance response time, and shows a significant proactive obstacle avoidance advantage especially in high-dynamic crossing scenarios.

The results of this study provide a new technical path for the autonomous navigation of indoor inspection robots, especially in the fields of firefighting inspection, factory logistics, and human-robot coexistence operations, where its fast-response dynamic obstacle avoidance ability has high engineering application value. The current research is mainly verified in indoor structured environments. In future work, the algorithm will be further extended to complex scenarios with multiple dynamic obstacles, and the real-time computational efficiency and adaptability to unstructured environments will be optimized.[16,17]

## REFERENCES

[1] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, 5(1), 90–98. https://doi.org/10.1177/027836498600500106.

[2] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33. https://doi.org/10.1109/100.580977

[3] LaValle, S. M., & Kuffner, J. J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378–400. https://doi.org/10.1177/02783640122067453

[4] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846–894. https://doi.org/10.1177/0278364911406761

[5] García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: theory and practice-a survey. Automatica, 25(3), 335–348. https://doi.org/10.1016/0005-1098(89)90002-2

[6] Tai, L., Paolo, G., & Liu, M. (2017). Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 31–36). IEEE. https://doi.org/10.1109/IROS.2017.8202134

[7] Pfeiffer, M., Schaeuble, M., Nieto, J., & others. (2017). From perception to decision: a data-driven approach to end-to-end motion planning for autonomous ground robots. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1527–1533). IEEE. https://doi.org/10.1109/ICRA.2017.7989182

[8] Everett, M., Chen, Y. F., & How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. IEEE Access, 9, 10357–10377. https://doi.org/10.1109/ACCESS.2021.3050338

[9] Williams, G., Aldrich, A., & Theodorou, E. A. (2017). Model predictive path integral control: from theory to parallel computation. Journal of Guidance, Control, and Dynamics, 40(2), 344–357. https://doi.org/10.2514/1.G001921

[10] Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. The International Journal of Robotics Research, 17(7), 760–772. https://doi.org/10.1177/027836499801700706

[11] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82(1), 35–45. https://doi.org/10.1115/1.3662552

[12] Tao, C., Kim, H., Yoon, H. J., & others. (2022). Control barrier function augmentation in sampling-based control algorithm for sample efficiency. IEEE Robotics and Automation Letters, 7(2), 3499–3506. https://doi.org/10.1109/LRA.2022.3147429

[13] Wang, F., Cheng, Y., & Tao, C. (2025). MPPI-DBaS: safe trajectory optimization with adaptive exploration. In 2025 44th Chinese Control Conference (CCC) (pp. 1616–1622). IEEE. https://doi.org/10.48550/arXiv.2502.14387

[14] Chen, G., Huang, Z., Zheng, J., & others. (2026). Adaptive MPPI path planning for mobile robots in complex dynamic environments. Computer Integrated Manufacturing Systems. https://doi.org/10.13196/j.cims.2026.0119

[15] Zheng, J., Huang, Z., Liu, J., & others. (2025). Path planning for mobile robots based on RRT*-guided MPPI in complex environments. Application Research of Computers, 42(12), 367–375. https://doi.org/10.19734/j.issn.1001-3695.2025.04.0127

[16] Wei, W., He, M., Qi, Z., & others. (2026). MPPI path tracking control method for tracked harvesters considering slip effects. Transactions of the Chinese Society for Agricultural Machinery, 57(8), 89–99. https://doi.org/10.6041/j.issn.1000-1298.2026.08.009

[17] Chen, Z., Fan, X., Ni, F., & others. (2025). Reactive motion planning framework based on control barrier function and sampling-based MPC for humanoid upper-body robots. Science China Technological Sciences, 68(11), 2120301. https://doi.org/10.1007/s11431-025-3053-1