Dynamic Obstacle Avoidance and Path Planning for Mobile Robots Integrating Improved Rapidly-Exploring Random Tree-Star and Improved Dynamic Window Approach

Dynamic Obstacle Avoidance and Path Planning

Xianyong Wei[1*], Hongying Si[2]
Shangqiu Polytechnic, Shangqiu 476000, China[1]
School of Mathematics and Statistics, Shangqiu Normal University, Shangqiu 476000, China[2]

***Abstract*—With the application and popularization of artificial intelligence and intelligent robots in daily life, the autonomous navigation and flexible operation capabilities of mobile robots have become particularly critical. Mobile robots perform well in regular environments, but face problems such as low accuracy in dynamic obstacle avoidance and weak adaptability to complex terrains. This study proposes to enhance the adaptability of the Rapidly-exploring Random Tree Star algorithm and integrate it with the A-Star algorithm, the Dynamic Window Approach, and visual sensor to construct an obstacle avoidance model. The objective is to enable the improved model to recognize various terrain features and enhance the accuracy of the path planning algorithm. The proposed model performed well in obstacle avoidance, with a success rate of 95.78% after ten training epochs and no more than four collisions within 4 minutes. In the experiment, as the obstacle increased every minute, the response speed of the proposed model remained below 25 seconds. The above results indicate that the quality of the planned path is higher than that of the other three models. The path optimization improvement combined with the A* algorithm is effective and has high real-time and accuracy, which can make mobile robots widely used in industries such as services, navigation, and logistics.**

***Keywords*—Rapidly-exploring random tree-star; dynamic window approach; A-star algorithm; dynamic obstacle avoidance; path planning; mobile robot**

I. Introduction

Driven by intelligent robot technology, Mobile Robot (MR) is widely used in industries such as autonomous driving, intelligent warehousing, and services. MR can replace humans in heavy and tedious labor and high-risk operations, among which dynamic obstacle avoidance and path planning are key technologies for MR to work safely and efficiently [1-2]. However, obstacle avoidance and path planning in complex dynamic environments still face numerous challenges, particularly in handling unstructured terrains such as forests, urban streets, and high-density dynamic obstacles. Existing methods exhibit limitations in real-time performance and adaptability. The latest research methods for obstacle avoidance in MR usually combine global path planning algorithms with local obstacle avoidance algorithms, and integrate multi-sensor data with dynamic environment prediction techniques to improve real-time performance, robustness, and obstacle avoidance accuracy in complex environments [3]. However, challenges such as low computational efficiency and suboptimal path optimization still exist, particularly in highly dynamic environments where robots may struggle to timely avoid fast-moving obstacles. In addition, they rely on simplified motion models and local obstacle information, so that they are not suitable for complex terrains such as urban streets or forests [4]. The current mainstream obstacle avoidance methods include Dynamic Window Approach (DWA), A-Star (A*), and Rapidly-Exploring Random Tree (RRT). RRT can generate progressively optimal global paths through random sampling. The heuristic search of A* can reduce redundant paths in random sampling by guiding the path to converge quickly towards the target point. Slight improvements to DWA can enhance the adaptability of obstacle avoidance models to dynamic environments with complex terrain [5]. Therefore, to deal with the low accuracy in path planning for MRs and poor adaptability to complex terrains, this study proposes a dynamic obstacle avoidance and path planning model for MR integrated RRT and A* with improved Dynamic Window Approach (IRA*-DWA). The proposed approach consists of two main components. First, by integrating the improved RRT and A * algorithm, the global path planning was optimized. RRT provides efficient exploration capabilities in unknown environments, while the heuristic search of A* further refines the initial path generated by RRT, ensuring both optimality and smoothness. Second, this global path planning is deeply integrated with DWA, forming a "global-local" dual-layer planning structure, where the global path generated by RRT-A* provides directional guidance for DWA. The improved DWA incorporates a dynamic obstacle trajectory prediction model and multi-source visual sensor data fusion to update and evaluate obstacle states in real -time. This integration allows the robot to maintain the optimality of the global path while dynamically adjusting local obstacle avoidance strategies, effectively coordinating responses to both static and dynamic obstacles. As a result, the system significantly enhances obstacle avoidance stability and efficiency in complex environments. The proposed model aims to achieve efficient and real-time autonomous

*Corresponding Author.

navigation in highly dynamic environments, providing valuable insights for future research on global path optimization and real-time obstacle avoidance strategies in complex and unstructured terrains.

The study is divided into five sections. Section II summarizes and discusses the research on dynamic obstacle avoidance and path planning. Section III constructs the obstacle avoidance model by integrating RRT and DWA, while incorporating the A* algorithm and visual sensors to enhance the model's ability to recognize terrain features. Section IV validates the improved algorithm and evaluates the overall performance of the obstacle avoidance model. Section V discusses the experimental results, explains how different algorithms are integrated, and how they improve the performance of the model in various environments. Section VI presents the conclusion, summarizing the findings of the study.

II. Related Work

Dynamic obstacle avoidance and path planning are crucial research directions in the field of MRs [6]. The main path planning includes global planning, local planning, and hybrid path planning [7-8]. There are abundant research results on improving path planning. For example, Huber et al. built a real-time perception-based fast obstacle avoidance strategy for MRs in dynamic and complex environments. The controller processed over 30,000 data points per second, with an evaluation time of 1ms, successfully avoiding collisions in complex indoor and outdoor environments [9]. Guo et al. built a dynamic obstacle avoidance risk zone strategy using Kalman filter and nonlinear model for robot obstacle avoidance. The robot could smoothly avoid moving obstacles with a high success rate. This method could effectively control the motion of robots [10]. Chen et al. proposed a risk aware sampling style local trajectory planning design based on a dual structure particle dynamic occupancy graph for the safe flight of quadcopter drones in dynamic environments. In field testing, the drone achieved 6m/s under the motion capture system and 2.5m/s when running on a low-cost single board computer [11]. Qi et al. built a distributed collaborative control algorithm on the basis of Hooke's law and damping repulsion function for collision and obstacle avoidance in multi-rotor formation tracking. In addition, a separation merging strategy was designed based on pigeon obstacle avoidance behavior to calculate the optimal speed for keeping the multi-rotor away from obstacles [12]. Li Z et al. proposed a collision avoidance framework that integrated B-splines and nonlinear model predictive control for the dynamic constraint problem of autonomous multi-axis distributed vehicles in path planning. The proposed framework was validated in different driving scenarios on the environmental testing platform, demonstrating the ability to effectively improve the accuracy of path planning and path tracking [13].

In terms of dynamic obstacle avoidance, mainstream research in academia has transitioned from discussing geometric model-based obstacle avoidance methods such as Artificial Potential Field (APF) and Vector Field Histogram (VFH) to developing and improving DWA algorithms. For example, Muñoz-Bañón et al. proposed a new Naive Valley Path method based on LiDAR to address the insufficient information accuracy. In practical applications, the system underwent autonomous driving for over 20 kilometers on BLUE, a research platform at the University of Alicante Science Park, with an average road center deviation of 0.24 meters and an average sampling time of 19.8ms [14]. Wang et al. built an anti-interference APF method JA-APF to address GPS signals being easily interfered with in unmanned surface ship path planning. The JA-APF could effectively solve the impact of GPS interference on path planning results and restore normal path planning as soon as possible [15]. Li Y et al. proposed an optimized A* algorithm that integrated cubic Bezier curves and DWA to address excessive path turns and long running time in practical applications. Compared with traditional algorithms, the algorithm reduced the turns on the path by 50% and the path length by 3.62% [16]. Kobayashi and Motoi combined DWA and virtual manipulator technology for local path planning of MRs. The simulation results verified the effectiveness of this method, especially in dynamic and narrow spaces, which could effectively avoid collisions and generate smooth paths [17].

From the above research, current research on MR dynamic obstacle avoidance and path planning obstacle avoidance mainly faces problems such as low obstacle avoidance accuracy, high computational resource consumption, inability to quickly obtain optimal solutions in complex environments, and susceptibility to getting stuck in local optima. Therefore, this study proposes the IRA*-DWA model, which introduces several key innovations. It integrates the improved RRT and A* algorithms to enhance the accuracy and computational efficiency of global path planning, while also incorporating an optimized DWA obstacle avoidance strategy, enabling the robot to make faster avoidance decisions in dynamic environments. Furthermore, this model combines dynamic obstacle trajectory prediction and multi-source sensor data fusion to achieve more accurate environmental perception, improve adaptability to complex and unstructured terrain, and effectively alleviate local optimization problems. The IRA*-DWA model aims to reduce the computational burden of path planning while ensuring rapid adaptation to complex environments in dynamic obstacle avoidance scenarios.

III. Improved Path Planning Algorithm and Improved IRA*- DWA Model Construction

A. Algorithm Strategy Combining RRT* and A*

Path planning can ensure safe and efficient navigation of MR in complex environments. RRT can quickly explore high-dimensional spaces through random sampling and incremental tree construction, making it extremely suitable for global path planning [18-19]. RRT* adds a path optimization mechanism depending on RRT, gradually approaching the optimal path by continuously reconnecting nodes, which improves the quality of path planning [20]. The schematic diagram of path exploration for RRT and RRT* is shown in Fig. 1.

In Fig. 1, RRT quickly generates a feasible path tree from the starting point to the target point through random sampling, but the path is often long and not smooth. RRT* adds a path optimization step on this basis, gradually improving the path by reconnecting nodes, resulting in a shorter final generated path. Although RRT* can generate asymptotic optimal paths in path planning, it is prone to insufficiently smooth paths. The heuristic search of A* can effectively optimize the smoothness and

feasibility of paths, which compensates for the shortcomings of RRT*. The operation of the A* algorithm is shown in Fig. 2.

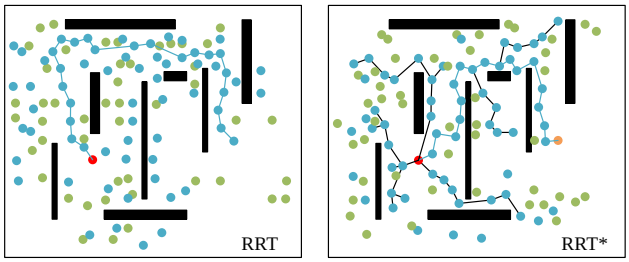


Fig. 1. Operation principal diagram of RRT*.

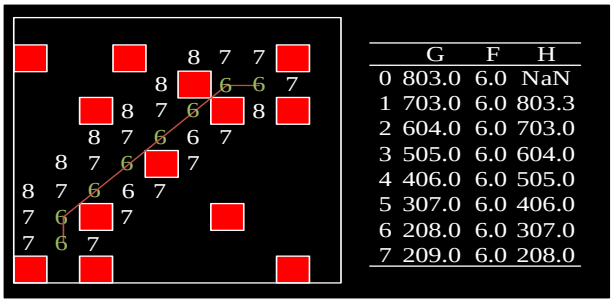


Fig. 2. Operation principal diagram of A*.

In Fig. 2, the A* algorithm calculates the total generation value of each node from the starting point, where F=G+H. G is the current path cost and H is a heuristic estimate. The node with the smallest F-value is used as the extension node of the current path, and updates adjacent nodes while avoiding red obstacles. The algorithm continuously repeats this process, ultimately obtaining the optimal path. The heuristic search of the A* is used to optimize the direction of the RRT* extension tree, making the search process more goal oriented. Based on this, an improved RRT* is obtained. Firstly, a tree T with the starting point q_{start} as the root node is initialized. A cost value priority queue Q for storing each expansion node is initialized and the starting point is added to the queue. During each expansion process, a node q_{near} is selected from the current tree and guided the random sampling point q_{rand} through the A* algorithm. The heuristic function of A* algorithm is shown in Eq. (1).

$$h(q_{rand}) = \left\| q_{rand} - q_{goal} \right\| \quad (1)$$

In Eq. (1), q_{goal} is the target point. $h(q_{rand})$ is the heuristic distance from the current random point q_{rand} to the target point. The heuristic value is combined with the cost value of the current node. The node with the lowest cost is selected for expansion, as shown in Eq. (2).

$$f(q_{rand}) = g(q_{near}) + C(q_{near}, q_{rand}) + h(q_{rand}) \quad (2)$$

In Eq. (2), $f(q_{rand})$ signifies the total cost of the node. $g(q_{near})$ signifies the cost of the path from the starting point to q_{near}. $C(q_{near}, q_{rand})$ is the actual cost from q_{near} to q_{rand}, representing distance, time, etc. The extension method of RRT* is used to add the newly sampled node q_{rand} to the current tree and expand the tree by connecting q_{near} and q_{rand}. Path optimization is carried out, checking the connection between the new node q_{rand} and the existing node and optimizing the path to reduce the total cost of the path. Eq. (3) displays the cost function.

$$C(q_1, q_2) = \left\| q_1 - q_2 \right\| \quad (3)$$

In Eq. (3), q_1 and q_2 are two points in the path applied to obtain the distance or cost between them. $\left\| q_1 - q_2 \right\|$ signifies the Euclidean distance between q_1 and q_2. In the process of path backtracking, the cost function f of the path is used to guide optimization. The expression is shown in Eq. (4).

$$f_{optimized} = \min(f_{current}, f_{optimized}) \quad (4)$$

In Eq. (4), f can check if there is a shorter path. By continuously optimizing and updating the paths in the tree, the total cost can be reduced. The optimal path is selected for connection, as expressed in Eq. (5).

$$\hat{q}_{new} = \arg_{q \in Near(q_{new})} (C(q, q_{new}) + C(q, q_{master})) \quad (5)$$

In Eq. (5), $Near$ represents the set of nodes in the tree that are closer to q_{new}. The optimal path is chosen to connect q_{new} and its main node q_{master}. Thus, the method integrated A* algorithm and RTF* algorithms (IRA*) is obtained, as shown in Fig. 3.

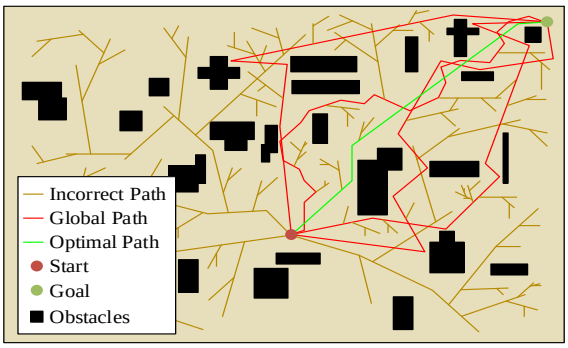


Fig. 3. Optimized algorithm strategy.

As shown in Fig. 3, RRT* first generates a feasible path tree covering complex environments through random sampling. The A* algorithm further optimizes the path based on heuristic functions, selecting the path with the lowest cost and higher smoothness as the final planning result. Through the IRA* algorithm, A* algorithm provides guidance for global optimization, making tree expansion more targeted and directional, and reducing unnecessary path exploration. RRT* ensures fast sampling and path optimization capabilities, ensuring asymptotic optimality of the final path.

B. *Obstacle Avoidance Model Based on Improved RRT* and Improved DWA*

After generating the optimal path through global path planning, MR needs to further construct a dynamic obstacle avoidance model to adapt to changes in dynamic obstacles and ensure the safety and flexibility of the robot during actual operation. This requires MR to be able to perceive dynamic obstacles around it within a limited time in complex environments before making path planning [21]. To achieve this goal, robots need to have the ability to recognize and avoid collisions, as well as perform path planning to find the optimal or feasible route. Table I displays the specific differences between dynamic obstacle avoidance and path planning obstacle avoidance.

TABLE I. COMPARISON BETWEEN DYNAMIC OBSTACLE AVOIDANCE AND PATH PLANNING OBSTACLE AVOIDANCE

Aspect	Dynamic Obstacle Avoidance	Path Planning Obstacle Avoidance
Environmental Type	Primarily dynamic environments, with obstacles changing overtime	Mostly static or slowly changing environments, with relatively stationary obstacle
Algorithm Goal	Real-time avoidance of moving obstacles to prevent collisions	Finding an optimal path from the start point to the destination
Real-Time Requirement	High real-time responsiveness required	Lower real-time requirements; path are generated and then executed
Path Adjustment	Dynamic path adjustment for real-time obstacle avoidance	Preplaned global paths, with potential updates or adjustments
Algorithm Complexity	Higher	Relatively lower
Use Cases	Dynamic traffic, pedestrian avoidance, robot navigation in complex environments	Indoor robots, automated warehouses, drones, etc.

According to Table I, the biggest difference between dynamic obstacle avoidance and path planning obstacle avoidance lies in their dynamism and real-time performance. In a dynamic environment, robots not only need to plan paths, but also need dynamic prediction and real-time obstacle avoidance. Therefore, higher computing power and more accurate perception are crucial. The IRA* can improve the global path planning ability of obstacle avoidance models. To further enhance the dynamic obstacle avoidance ability, the DWA, which is more suitable for dynamic obstacle avoidance, is integrated based on the IRA*. The operation process of DWA is shown in Fig. 4.

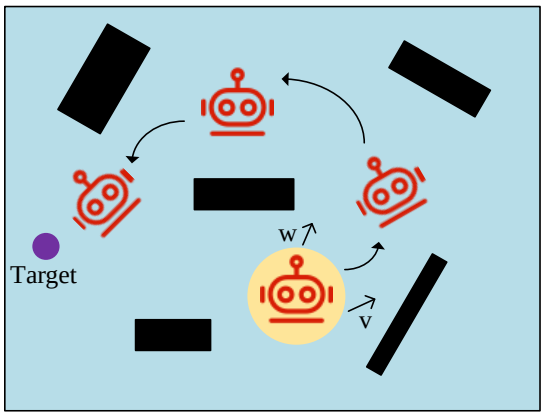


Fig. 4. Operational principal diagram of DWA.

Fig. 4 shows the running process of the DWA algorithm, which obtains information on the current position, speed, and obstacles of the robot through sensors. A set of feasible motion trajectories is generated based on the current speed, acceleration, etc. in the velocity space. Each trajectory is scored based on indicators, and the trajectory with the highest score is selected as the next motion path for the robot. The corresponding speed and direction commands are sent to the robot for actual motion, allowing DWA to achieve real-time obstacle avoidance and path following. In complex and irregular terrain, the shape and position of obstacles may change rapidly. Therefore, the prediction accuracy of traditional distance sensors relied on by the DWA algorithm will decrease. By introducing visual sensors, the perception ability of irregular terrain can be improved and richer environmental information can be provided. Given the current state $x=[x,y,\theta]^T$ of the robot, x and y are position coordinates and θ is orientation. The maximum speed and acceleration of the robot are defined. A time window Δt is defined for predicting future trajectories. Multiple candidate trajectories are generated within Δt based on the speed limit and dynamic model of the robot, and each trajectory is evaluated to calculate its cost function. The cost function of DWA usually consists of three parts: obstacle avoidance cost, velocity cost, and acceleration cost. The total cost function is shown in Eq. (6).

$$\begin{cases} J=\omega_{obs}\cdot J_{obs}+\omega_{vel}\cdot J_{vel}+\omega_{acc}\cdot J_{acc} \\ J_{obs}=\min\limits_{i}\left(\dfrac{1}{\left\|x_{traj(i)}-x_{obs}\right\|^2}\right) \\ J_{vel}=\left\|v-v_{target}\right\| \\ J_{acc}=\left\|a\right\| \end{cases} \tag{6}$$

In Eq. (6), J_{obs} is the obstacle avoidance cost, reflecting the distance between the trajectory and the obstacle. Close distance indicates higher costs. J_{vel} is the speed cost, which measures the difference between the current speed and the

expected speed. J_{acc} is the acceleration cost, which measures the smoothness of the control input. The trajectory with the minimum cost function value is used as the final control input, and the velocity v and angular velocity ω are controlled to calculate the collision risk between the predicted path and obstacles. For the obstacle position $q_{obs}(t)$ at a certain moment, by predicting its motion velocity $v_{obs}(t)$ and acceleration $a_{obs}(t)$, the future position can be estimated, as displayed in Eq. (7).

$$q_{obs}(t+\Delta t) = q_{obs}(t) + v_{obs}(t)\Delta t + \frac{1}{2}a_{obs}(t)\Delta t^2 \quad (7)$$

In Eq. (7), Δt signifies the predicted time step. $q_{obs}(t)$ signifies the current position of the obstacle. Based on the Kalman filter, DWA can achieve multi-step prediction to improve its adaptability to dynamic obstacles, estimate the possible positions in future time periods, and dynamically update the future trajectory of obstacles. Assuming $\hat{q}_{future}(t)$ is the predicted trajectory of the future position of the robot and $\hat{q}_{obs}(t)$ is the predicted trajectory of obstacles, the cost function expression for avoiding collisions is shown in Eq. (8).

$$\cos t_{collision}(v,\omega) = \sum_{i=1}^{n} safe(q_{robot}(t+i), q_{obs}(t+i)) \quad (8)$$

In Eq. (8), $safe(q_{robot}, q_{obs})$ is a function that represents the safe distance between the current robot position and the predicted obstacle position. If the safe distance is below the set threshold, the value is considered high cost. The final cost function is rewritten, as shown in Eq. (9).

$$\begin{cases} \mathrm{A} = \alpha \cdot \cos t_{collision} \\ \mathrm{B} = \beta \cdot \cos t_{speed}(v) \\ \mathrm{H} = \gamma \cdot \cos t_{heading}(\omega) \\ \cos t(v,\omega) = \mathrm{A} + \mathrm{B} + \mathrm{H} \end{cases} \quad (9)$$

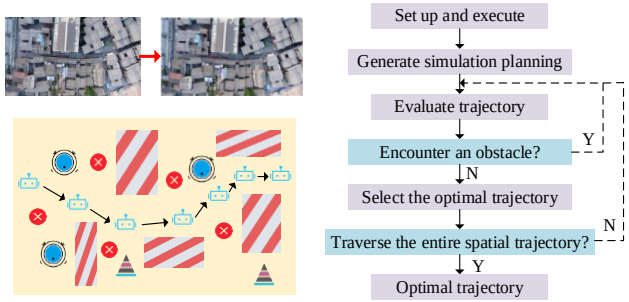
In Eq. (9), α, β, and γ are weight coefficients, representing the weights for controlling obstacle avoidance, speed, and heading, respectively. After combining visual information, for each time step, the position and shape of obstacles can be updated based on data from visual sensors and distance sensors. The cost function expression after introducing visual sensors is shown in Eq. (10).

$$\cos t_{collision}(v,\omega) = \sum_{i=1}^{n} safe(q_{robot}(t+i), q_{obs}(t+i), I_{depth}(t+i), I_{RBG}(t+i)) \quad (10)$$

In Eq. (10), I_{depth} and I_{RBG} represent image data from the depth sensor and the red, green, and blue cameras, respectively. The improved function can consider the visual sensor to provide more accurate obstacle shapes and positions. Finally, the DWA cost function after combining visual and dynamic obstacle prediction is shown in Eq. (11).

$$\begin{cases} \mathrm{A} = \alpha \cdot \cos t_{collision} \\ \mathrm{B} = \beta \cdot \cos t_{speed}(v) \\ \mathrm{H} = \gamma \cdot \cos t_{heading}(\omega) \\ \Gamma = \varphi \cdot \cos t_{visual}(I_{depth}, I_{RGB}) \\ \cos t(v,\omega) = \mathrm{A} + \mathrm{B} + \mathrm{H} + \Gamma \end{cases} \quad (11)$$

In Eq. (11), φ is the weight coefficient related to the visual sensor, used to control the impact of visual information on the total cost function. By predicting the trajectory of dynamic obstacles and combining visual sensors, the adaptability of DWA algorithm in dynamic and complex environments is effectively enhanced. These two improvements enable DWA to more accurately respond to dynamic obstacles and complex terrain, while still ensuring the quality of path planning even in high real-time requirements. The final obstacle avoidance model IRA*-DWA is shown in Fig. 5.



(a)Expected Obstacle Avoidance Effectiveness (b)Model Execution Process

Fig. 5. Operational diagram of IRA*-DWA model.

Fig. 5 (a) shows the expected obstacle avoidance effect of IRA*-DWA in a real scene. A real scene is selected and formatted to design a terrain map filled with static and dynamic obstacles. MR should avoid five erroneous intersections through visual sensors, generate the optimal path based on algorithms, and execute it. Fig. 5 (b) displays the operational process of IRA*-DWA. The obstacle avoidance model first generates an initial path plan based on IRA* and evaluates the path to select the optimal trajectory. During the execution process, the visual sensor is used to perceive the environment and predict the trajectory of dynamic obstacles. The model cyclically traverses the entire spatial path until the final global optimal path is generated, achieving efficient obstacle avoidance and path optimization for dynamic environments and complex terrains.

IV. Experimental Performance Evaluation of IRA* and IRA*-DWA

A. Performance Verification of IRA* Algorithm

To verify the performance, an experimental platform is set up consisting of two parts: software and hardware. The software part uses MATLAB and ROS as simulation environments,

combined with Gazebo for dynamic environment modeling and algorithm verification. Meanwhile, Python is used to write algorithm implementations, including improved RRT and DWA algorithm modules. The hardware part uses a MR platform equipped with RPLIDAR lidar, RGB vision sensors, and NVIDIA Jetson embedded controller. Static and dynamic obstacles are arranged on the experimental site to simulate real complex environments. The performance is verified through hardware operation. The IRA* algorithm is compared with Dijkstra and PRM. Furthermore, to further evaluate the adaptability of the proposed algorithm, experiments are conducted in both indoor structured environments and outdoor unstructured terrains. The indoor experiments include scenarios with narrow passages and randomly distributed obstacles, while the outdoor experiments cover complex terrains such as slopes and gravel paths. These experiments aim to evaluate the robustness and obstacle avoidance ability of the model in different environmental conditions. The indoor and outdoor training results are shown in Fig. 6.

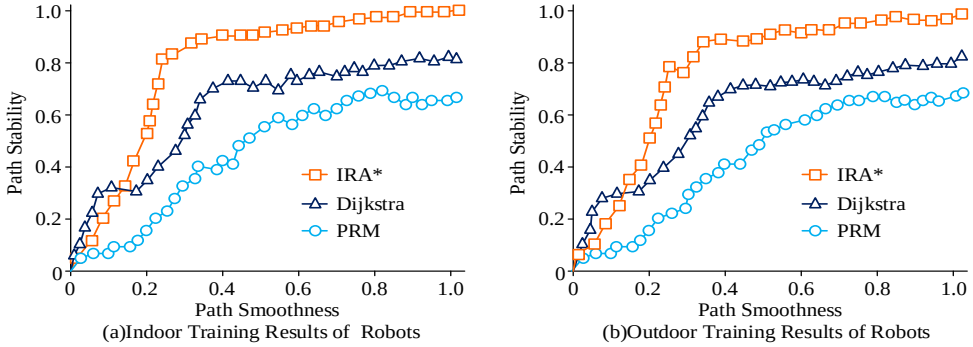


Fig. 6. Comparison of path smoothness and stability.

From Fig. 6, both indoors and outdoors, IRA* consistently outperformed the other two algorithms on path stability and smoothness. In Fig. 6 (a), when the path smoothness reached 0.6, IRA* already reached a path stability of 0.9, while Dijkstra and PRM had path stability of around 0.76 and 0.6, respectively. The performance curve of IRA* rose faster, proving its ability to optimize paths earlier in complex indoor environments. In Fig. 6 (b), the path stability of IRA* approached 1 when the path smoothness was 0.8, while Dijkstra and PRM reached relatively high path stability when the path smoothness was close to 1. Even in dynamic outdoor environments, IRA* still maintained its excellent performance. In summary, IRA* not only significantly improves path smoothness, but also achieves optimal path stability in various environments, demonstrating strong comprehensive performance. Subsequently, the accuracy and obstacle avoidance success rate of the algorithm are validated, as shown in Fig. 7.

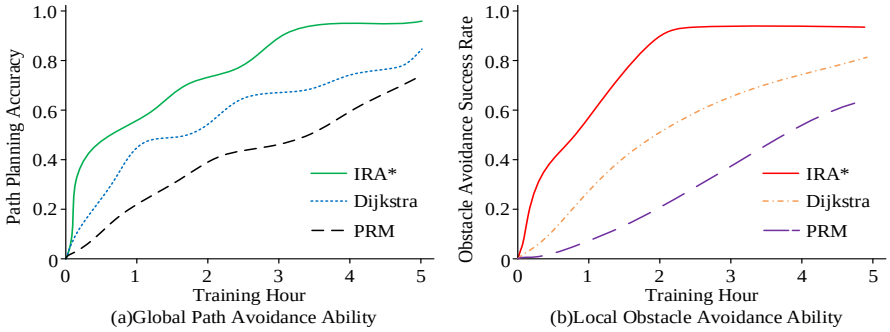
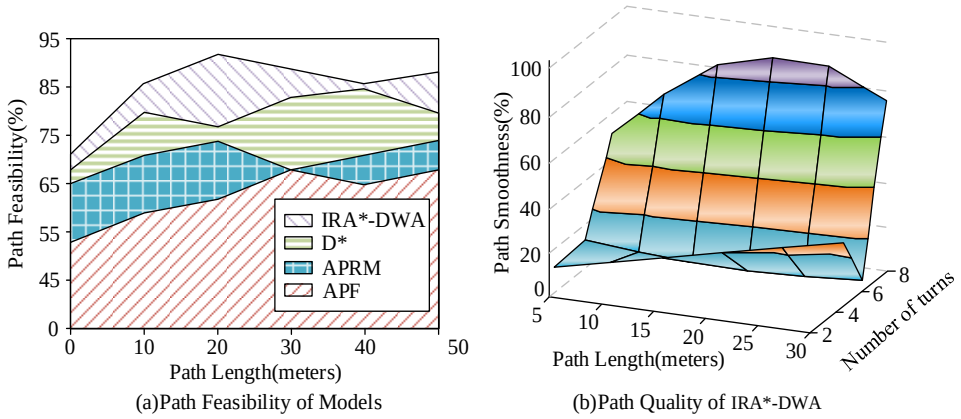


Fig. 7. Comparison of global and local path planning capabilities.

In Fig. 7 (a), IRA* showed a rapid increase in accuracy at the beginning of training, reaching approximately 70% planning accuracy within 2 hours, and ultimately stabilizing at 95.45% in the third hour. Dijkstra grew slowly, reaching only 45.12% within 1 hour and stabilizing at 80.68% after five hours. PRM grew the slowest, only increasing from an initial 20% to a final 65.31%. In Fig. 7 (b), the obstacle avoidance success rate of IRA* rapidly increased in the first two hours, reaching 87%, and approached 90% afterwards, demonstrating extremely high local obstacle avoidance ability. Dijkstra showed a slight lag in improving obstacle avoidance ability, with a relatively steady growth rate, ultimately reaching 81.28% within five hours. PRM had the worst performance, with a slow increase in obstacle avoidance success rate throughout the entire training process, only at 60.36%.

B. *Performance Analysis of IRA*-DWA Model*

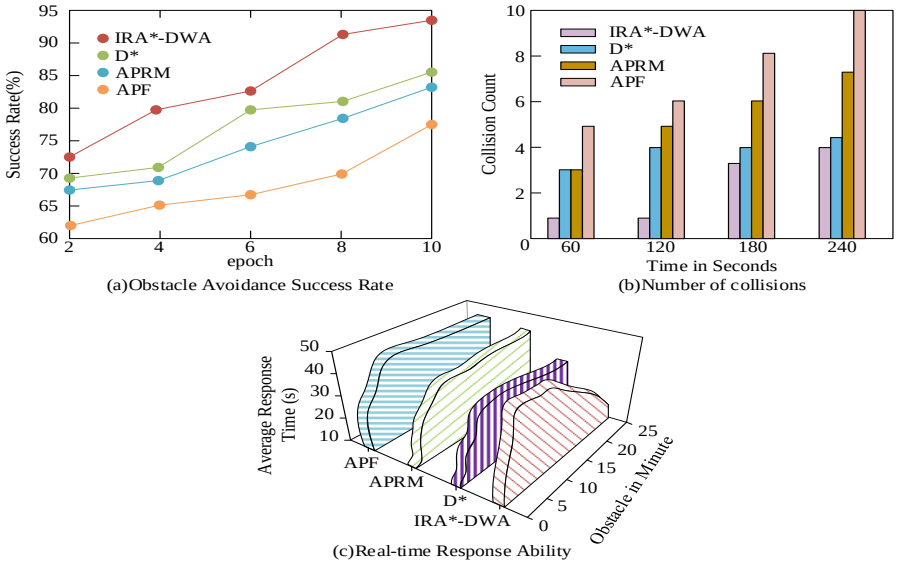
After verifying the performance of the IRA* algorithm, to further validate its practicality and scalability in dynamic obstacle avoidance scenarios, the study also analyzes the application effect of the IRA*-DWA obstacle avoidance model. The experimental setup for the IRA*-DWA model is the same as above. Three datasets, KITTI, OpenLORIS-Scene, and ApolloScope, are selected and compared with D*, Probabilistic Roadmap combined with A* (APRM), and APF model. The path quality performance is shown in Fig. 8.



(a)Path Feasibility of Models (b)Path Quality of IRA*-DWA

Fig. 8. Path quality of models.

In Fig. 8 (a), the path feasibility of IRA*-DWA remained at the highest level, reaching its peak at a path length of 20 and maintaining 86.48% even at a path length of 30. In contrast, the feasibility of D* and APRM was slightly lower, around at 77.64% and 71.4% respectively when the path length exceeded 40. The feasibility of APF was the lowest. To avoid distortion of individual test data, the IRA*-DWA in path quality is presented separately. Fig. 8 (b) shows the three-dimensional visualization effect of path length, number of turns, and path smoothness. IRA*-DWA maintained high smoothness in the path length from 0 to 30. Especially when the path length was 20 and the number of turns was small, the path smoothness was 89.76%. Overall, path length is positively correlated with the number of turns. Longer paths are usually smoother. However, when there are many turns, especially sharp turns, the smoothness is low. Subsequently, the obstacle avoidance success rate, number of collisions, response time, and other specific obstacle avoidance performance are verified, as shown in Fig. 9.



(a)Obstacle Avoidance Success Rate (b)Number of collisions

(c)Real-time Response Ability

Fig. 9. Obstacle avoidance performance.

In Fig. 9 (a), the obstacle avoidance success rate of IRA*-DWA was 73.56% after 2 epochs. As the training epochs increased, its success rate rapidly increased, reaching approximately 95.78% in the 10th round. The obstacle avoidance success rate of D* increased from 68.97% to 83.28%, but it was lower than that of IRA*-DWA. In Fig. 9 (b), within 240 seconds, the number of collisions of IRA*-DWA remained the lowest, basically below 4 times, while the number of collisions of D* was 4 times, APRM was 7 times, and APF was the worst, up to 10 times. As time increased, the number of collisions of IRA*-DWA increased the slowest, showing stability advantages. In Fig. 9 (c), as the number of obstacles increased, the average response time of IRA*-DWA was always controlled within 25 seconds. Even with 25 obstacles, the response time was only 18 seconds. The response time of other models significantly increased with the increase of obstacles. D* had a response time of approximately 30 seconds when encountering 25 obstacles. APRM exceeded 40 seconds. APF had the worst response time, approaching 45 seconds. The results indicated that even as the number of obstacles increased, the IRA*-DWA model consistently maintained a high obstacle avoidance success rate. Furthermore, this study validated the response time of the model in six different environments, and the response time results are shown in Fig. 10.

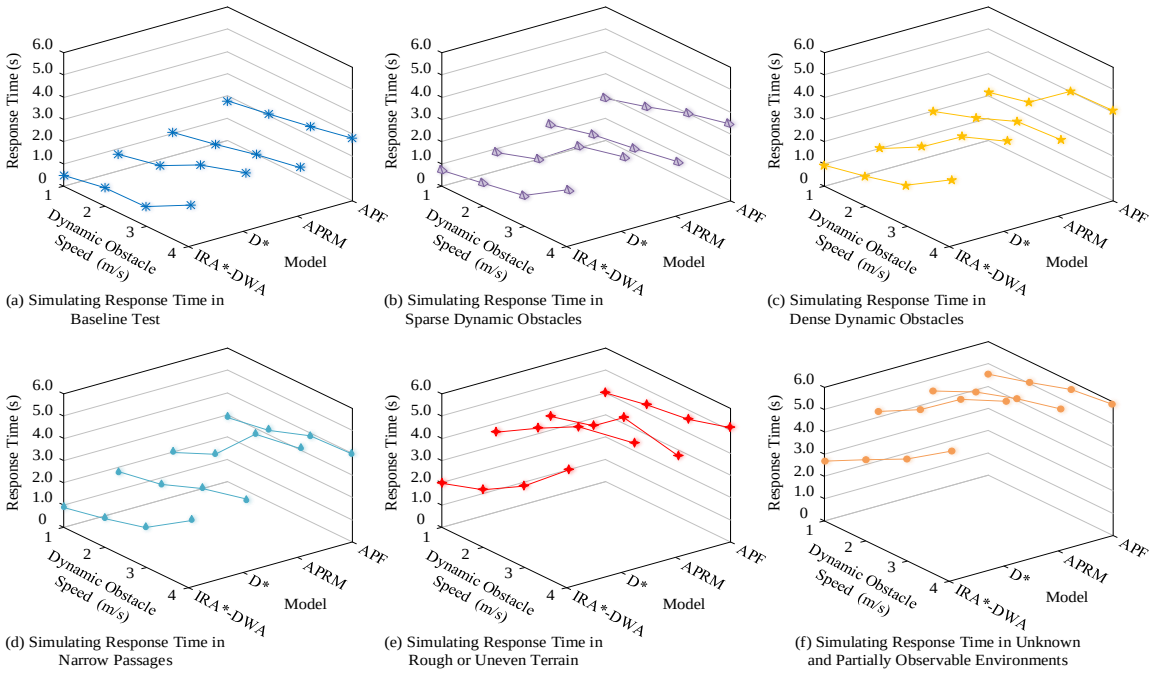


Fig. 10. Simulating response time in different environments.

In Fig. 10 (a), when there was no dynamic obstacle interference, the response time of IRA*-DWA remained below 1.5s across all speed conditions. The response time of D*, APRM, and APF increased significantly with the increase of obstacle speed, reaching 1.4s, 1.6s, and 2.8s, respectively, at 4.0m/s. In Fig. 10 (b), in a low-density dynamic environment, the response time of IRA*- DWA was only 0.9s at 1.0m/s. Although the response time increased with the obstacle speed, it consistently remains below 2.7s, demonstrating significantly higher obstacle avoidance efficiency than the other models. As depicted in Fig. 10 (c), in a high-density dynamic environment, when the obstacle speed was 4.0m/s, the response time of IRA*-DWA increased to 1.7s. However, compared with other models, IRA*-DWA still maintained the lowest response time, effectively avoiding the significant delays observed by traditional algorithms under high computational loads. In Fig. 10(d), in spatially constrained environments, such as narrow passages, the response time of IRA*-DWA increased to 2.2s at an obstacle speed of 4.0m/s, while D* and APRM increased to 2.4s and 3.7s, respectively, with APF reaching a peak of 4.0s. Fig. 10 (e) simulates rough and unstructured terrain, including slopes and gravel surfaces. The response time of IRA*-DWA ultimately increased to 3.6s. Because the IRA*-DWA model integrates visual sensors and trajectory prediction, it can better adapt to complex terrain. Fig. 10(f) evaluates the robot's response capability in an unknown environment. The response time of IRA*-DWA increased to 3.8s at an obstacle speed of 4.0m/s, while D*, APRM, and APF reached 5.2s, 4.8s, and 5.9s, respectively. These results indicate that even with an increase in the number of obstacles, IRA*-DWA can maintain the fastest response time and remain relatively stable. Subsequently, the study evaluates the environmental adaptability of the four models across three datasets, as presented in Fig. 11.

In Fig. 11 (a), when the terrain adaptability was around 1, the sensor adaptability of IRA*-DWA rapidly increased to 92.34% and stabilized at over 93% in the subsequent stage. D* came second, with a final sensor adaptability of around 89.75%. After the terrain adaptability of APRM exceeded 1, the adaptability growth slowed down and stabilized at 85.67%. APF performed the worst, with a final sensor adaptability of 82.3%. In Fig. 11 (b), the terrain adaptability of IRA*-DWA was stable at 94.38%, which was higher than that of other models. The final sensor adaptability of APRM was 83.25%. The sensor adaptability of APF was less than 82%, and its performance was poor. In Fig. 10 (c), the sensor adaptability of IRA*-DWA rapidly increased to 93.5% and eventually stabilized at 94.63%. Overall, the environmental adaptability of the IRA*-DWA model is consistently higher than that of the D*, APRM, and APF. After comparing the environmental adaptability of four models on three datasets, the user experience score is verified in actual scenarios, as shown in Fig. 12.

In Fig. 12 (a), the user experience score of IRA*-DWA was significantly higher than that of other models, distributed in the range of 7.5-9.5. It could maintain a high score even at high feature complexity. The score of D* was slightly lower than that of IRA*-DWA, mainly distributed in 6.5-8.0, and decreased slightly at high feature complexity. The scores of APRM and APF were significantly lower than those of IRA*-DWA, and showed a clear downward trend with increasing feature complexity. In Fig. 12 (b), the adaptability of IRA*-DWA rapidly increased from 52.37% to 57.1% in the first two training rounds, and reached 90% in the seventh round, ultimately stabilizing at 93.64%. D* followed closely, with adaptability increasing from 32.45% to 85.46%, but consistently lower than that of IRA*-DWA. APRM and APF had poor performance and slow growth rates. APF was the worst, only reaching 64.74%. IRA*-DWA shows a clear leading advantage in real-time adaptability, being able to quickly adapt to complex scenarios in a short period of time.

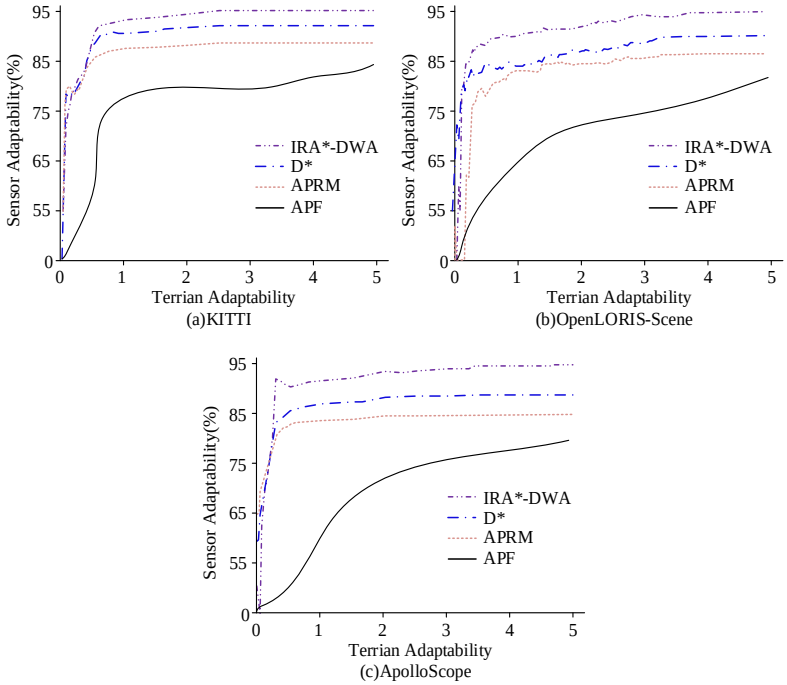


Fig. 11. Environmental adaptability.

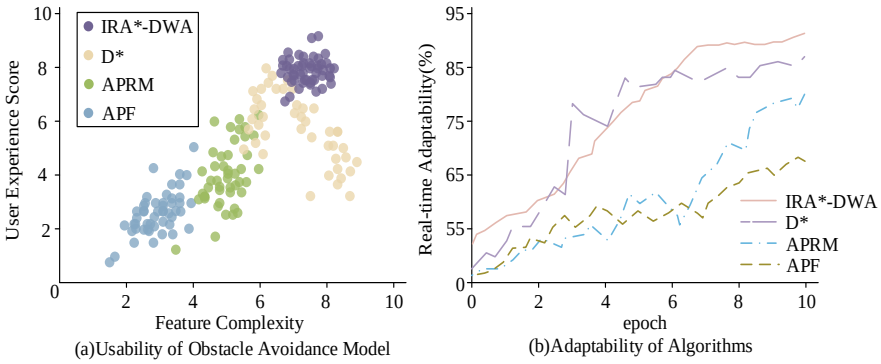


Fig. 12. Comparison between usability and implementation complexity.

V. Discussion

The IRA*-DWA obstacle avoidance model integrates the path adaptability of RRT and A*, an improved DWA-based dynamic obstacle avoidance mechanism, multi-source sensor data fusion technology, and real-time computational optimization strategy to form a comprehensive adaptive navigation system. When navigating in complex spatial structures such as unknown terrains and narrow passages, the system can re-plan the optimal path in real-time. In high-density pedestrian environments or multi-robot scenarios, it can predict obstacle trajectories and rapidly select the best avoidance strategy. By integrating data from LiDAR, RGB cameras, and depth cameras, the system extends its adaptability to extreme conditions such as low-light environments, adverse weather, and irregular terrains. Furthermore, this model utilizes intelligent computing resource management and parallel computing technology to maintain low latency response even under high computing loads. This multidimensional adaptability allows the model to overcome the limitations of laboratory testing environments and maintain stable and efficient navigation and obstacle avoidance performance in real-world dynamic scenarios, providing a reliable solution for autonomous mobility in complex environments.

The experimental results demonstrate that IRA*-DWA outperforms D*, APRM, and APF obstacle avoidance models in terms of path planning accuracy, obstacle avoidance success rate, response time, and environmental adaptability. It has particularly superior real-time obstacle avoidance capability in high dynamic environments. This model can quickly adapt to various complex scenarios, and its robustness and adaptability exceed those of existing mainstream obstacle avoidance models, which has been confirmed by multiple dataset evaluations. Moreover, IRA*-DWA consistently has lower response time than D*, APRM, and APF across all complex dynamic environments, with outstanding computational efficiency and real-time performance, especially in high-density dynamic obstacles, narrow passages, complex terrains, and unknown

environments. The experimental results validate the advantages of integrating path planning with dynamic obstacle avoidance, demonstrating that IRA*-DWA provides an optimized solution for autonomous navigation in high-dynamic environments. Consequently, IRA*-DWA exhibits significant application potential, providing a reliable solution for MR path planning and obstacle avoidance in complex dynamic environments.

VI. Conclusion

In the modern society that pursues high efficiency, the application of MR requires excellent dynamic obstacle avoidance and path planning algorithms. Aiming at the problem that current methods cannot make optimal responses in complex scenes and perform poorly in complex and irregular environments, a MR obstacle avoidance model IRA*-DWA was proposed by integrating improved RRT* and improved DWA. Combining RRT* and A* with the improved DWA, the goal of improving obstacle avoidance accuracy and getting rid of simplified motion models is achieved. The optimized IRA*-DWA model was validated. The IRA*-DWA showed higher path quality and obstacle avoidance ability than other models, with an obstacle avoidance success rate of 95.78%. The adaptability of sensors in the three datasets was 93.45%, 94.38%, and 94.63%, respectively. More importantly, IRA*-DWA performed well on user experience rating, with a score of 7.5-9.5. The IRA*-DWA model had strong real-time adjustment ability, reaching 93.64% after training. The proposed IRA*-DWA performs better than mainstream D*, PRM, and APF models. The above results indicate that the IRA*-DWA model has strong practicality and can be applied in practical scenarios. The proposed IRA*-DWA model is most effective in structured and semi-structured environments with sufficient sensor coverage but may face limitations in highly unpredictable or extremely unstructured terrains where real-time perception and computational constraints significantly impact performance. The improved model may result in a higher computational burden when dealing with path planning in environments with many obstacles. In the future, more flexible and efficient path planning and obstacle avoidance can be achieved by parallelizing the algorithm and strengthening the multimodal planning and decision-making framework.

Funding

This paper was supported by 1) Key scientific research project of universities in Henan Province, Project name: Research on mobile robot path optimization technology for intelligent navigation, Project No.: 23A520059; 2) Key scientific research project of higher universities in Henan Province, project name: Petrov-Galerkin application research of finite element method in nonlinear equation, project number: 25A110013.

References

[1] Yu Z, Si Z, Li X, Wang D, Song H. A novel hybrid particle swarm optimization algorithm for path planning of UAVs. IEEE Internet of Things Journal, 2022, 9(22): 22547-22558.

[2] Zhang T, Xu J, Wu B. Hybrid path planning model for multiple robots considering obstacle avoidance. IEEE Access, 2022, 10(3): 71914-71935.

[3] Kabir H, Tham M L, Chang Y C. Internet of robotic things for mobile robots: concepts, technologies, challenges, applications, and future directions. Digital Communications and Networks, 2023, 9(6): 1265-1290.

[4] Hewawasam H S, Ibrahim M Y, Appuhamillage G K. Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. IEEE Open Journal of the Industrial Electronics Society, 2022, 3(5): 353-365.

[5] Yao Q, Li H, Gao P, Guo H, Zhong C. Mapping Irregular Local Climate Zones from Sentinel-2 Images Using Deep Learning with Sequential Virtual Scenes. Remote Sensing, 2022, 14(21): 5564.

[6] Tang Y, Qi S, Zhu L, Zhuo X, Zhang Y, Meng F. Obstacle avoidance motion in mobile robotics. Journal of System Simulation, 2024, 36(1): 1-26.

[7] Wang N, Zhang B, Chi H, Wang H, Mcloones S, Liu H. DUEL: Depth visUal Ego-motion Learning for autonomous robot obstacle avoidance. The International Journal of Robotics Research, 2024, 43(3): 305-329.

[8] Gu X, Zhang M, Lyu J, Ge Q. Generating Urban Road Networks with Conditional Diffusion Models. ISPRS International Journal of Geo-Information, 2024, 13(6): 203.

[9] Huber L, Slotine J J, Billard A. Fast obstacle avoidance based on real-time sensing. IEEE Robotics and Automation Letters, 2022, 8(3): 1375-1382.

[10] Guo B, Guo N, Cen Z. Obstacle avoidance with dynamic avoidance risk region for mobile robots in dynamic environments. IEEE Robotics and Automation Letters, 2022, 7(3): 5850-5857.

[11] Chen G, Peng P, Zhang P, Dong W. Risk-aware trajectory sampling for quadrotor obstacle avoidance in dynamic environments. IEEE Transactions on Industrial Electronics, 2023, 70(12): 12606-12615.

[12] Qi J, Guo J, Wang M, Wu C, Ma Z. Formation tracking and obstacle avoidance for multiple quadrotors with static and dynamic obstacles. IEEE Robotics and Automation Letters, 2022, 7(2): 1713-1720.

[13] Li Z, Li J, Wang W. Path planning and obstacle avoidance control for autonomous multi-axis distributed vehicle based on dynamic constraints. IEEE Transactions on Vehicular Technology, 2022, 72(4): 4342-4356.

[14] Muñoz-Bañón M Á, Velasco-Sanchez E, Candelas F A, Torres F. Openstreetmap-based autonomous navigation with lidar naive-valley-path obstacle avoidance. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12): 24428-24438.

[15] Wang J, Xiao Y, Li T, Chen C P. A jamming aware artificial potential field method to counter GPS jamming for unmanned surface ship path planning. IEEE Systems Journal, 2023, 17(3): 4555-4566.

[16] Li Y, Jin R, Xu X, Qian Y, Wang H, Xu S, Wang Z. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. IEEE Access, 2022, 10(6): 57736-57747.

[17] Kobayashi M, Motoi N. Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles. IEEE Access, 2022, 10(2): 17018-17029.

[18] Tang Y, Qi S, Zhu L, Zhuo X, Zhang Y, Meng F. Obstacle avoidance motion in mobile robotics. Journal of System Simulation, 2024, 36(1): 1-26.

[19] Lu C, Gao R, Yin L, Zhang B. Human–robot collaborative scheduling in energy-efficient welding shop. IEEE Transactions on Industrial Informatics, 2023, 20(1): 963-971.

[20] Meng B H, Godage I S, Kanj I. RRT*-based path planning for continuum arms. IEEE Robotics and Automation Letters, 2022, 7(3): 6830-6837.

[21] Wanasinghe T R, Gosine R G, Petersen B K, Warrian P J. Digitalization and the future of employment: A case study on the Canadian offshore oil and gas drilling occupations. IEEE Transactions on Automation Science and Engineering, 2023, 21(2): 1661-1681.