PATH PLANNING AND TRAJECTORY TRACKING CONTROL OF WHEELED MOBILE ROBOTS BASED ON RRT-MPC MODEL

Lu Wang
School of Intelligent Manufacturing, Wuxi Vocational College of Science and Technology, Wuxi, 214028, China

Abstract - A comprehensive autonomous navigation framework combining improved rapidly-exploring random tree (RRT) and model predictive control (MPC) is proposed to address the limitations of incomplete kinematic constraints on wheeled mobile robots in complex environments and their susceptibility to external disturbances leading to insufficient navigation accuracy. In the global path planning stage, a multi-objective stochastic adaptive RRT* (MRA-RRT*) algorithm is proposed, which introduces an adaptive step size sampling strategy based on dynamic potential field and a parent node re-selection mechanism based on Nash equilibrium, effectively balancing environmental exploration efficiency and path smoothness. In the local trajectory tracking stage, an improved MPC controller based on a nonlinear disturbance observer (NDO) and a dual closed-loop architecture is designed. By cascading the kinematic outer loop and the dynamic inner loop, real-time feedforward compensation for model mismatch and external environmental disturbances is achieved. The simulation results show that in sparse obstacle environments, the average path length of MRA-RRT* algorithm is 63.55m, the planning time is only 1.86s, and the smoothing cost is as low as 4.21rad. The overall performance is significantly better than traditional planning algorithms. In the trajectory tracking experiment, faced with 5N sudden lateral wind interference, the peak heading angle error of the improved MPC algorithm was only 0.04 rad, while the traditional sliding mode control reached 0.23 rad. The comprehensive navigation model effectively balances planning efficiency, path quality, and control robustness, providing reliable technical support for autonomous navigation of mobile robots under complex working conditions.

Keywords: Wheeled mobile robot, RRT*, Path planning, MPC, Trajectory tracking.

1. Introduction

Driven by industrial automation and artificial intelligence technology, wheeled mobile robots exert a crucial function in warehousing logistics, security inspection and service fields. In complex and dynamic operating environments, autonomous navigation from the starting point to the target point is an important prerequisite for the application of mobile robots [1-3]. The autonomous navigation system mainly has two parts, including global path planning and local trajectory tracking control. The former is to find a collision-free geometric path, while the latter controls the robot to move accurately along the reference path [4-5]. Limited by non-holonomic constraints, environmental uncertainty, and nonlinear dynamic characteristics, designing efficient planning algorithms and robust tracking controllers still faces huge challenges. For path planning, relevant scholars have proposed many advanced algorithms. However, the convergence speed of many algorithms is still slow in narrow channels, and the requirements for path smoothness of non-holonomic constraints of the robot are ignored, making it difficult for the generated path to be accurately executed by the underlying motion controller [6-7]. For trajectory tracking control, existing tracking control research mainly separates kinematics and dynamics. A single model predictive control (MPC) is difficult to deal with dynamic disturbances such as ground slippage, and pure dynamics controllers are slightly insufficient when dealing with complex geometric constraints [8-9]. Therefore, more advanced and effective path planning and trajectory tracking control technologies must be adopted.

Scholars have proposed many advanced algorithms to solve the path planning. Hassan et al. used the hybrid A* for path planning, and combined with the optimized hybrid boundary planning optimization algorithm to adjust the control parameters of mobile robots in dynamic

dx.doi.org/10.17683/ijomam/issue24.9

environments. The hybrid A* generated shorter and smoother paths, while the planning algorithm converged faster [10]. To solve the blind search in RRT, Yuan L et al. proposed an improved Informed RRT* algorithm that integrated adaptive growth and variable weight sampling, which significantly improved the optimization efficiency and trajectory smoothness by combining the minimum capture method [11]. In view of the excessive cost caused by the redundant path branches of traditional Rapidly-exploring random tree (RRT), Liangwu et al. proposed a geometric probability model based on improved artificial potential field and RRT. This fusion strategy effectively solved unreachable targets by enhancing the search directionality [12]. However, these representative baseline algorithms are still limited by the underlying extension mechanism and fall into local deadlocks when dealing with high-density obstacles and narrow channels. Meanwhile, traditional algorithms only use geometric path length as the optimization objective and do not deeply integrate the non-complete kinematic constraints of wheeled mobile robots during the node growth stage. Li X et al. combined kinematic constraints with third-order Bezier curves to smooth and optimize the redundant paths found [13]. However, this step-by-step processing mode is prone to collision risks in narrow corridors with limited environments.

Trajectory tracking of mobile robots is one of the current research focuses. Wheeled mobile robots face trajectory tracking problems due to external disturbances and parameter uncertainty. Therefore, Kordi et al. built a hybrid double-loop control strategy based on nonlinear predictive control and finite-time integral terminal sliding mode control (SMC). This method, combined with the finite-time disturbance observer, successfully eliminated the chattering phenomenon and achieved rapid convergence of the tracking error [14]. To deal with the trajectory tracking of mobile robots under unknown disturbances, Moudoud et al. designed a terminal integral SMC scheme based on an adaptive switching gain strategy. The results showed that the double closed-loop system effectively eliminated the chattering phenomenon and achieved fast finite-time convergence of the state [15]. Meng et al. constructed a variable-step MPC strategy that integrated different keeper mechanisms to address the multiple requirements for accuracy and real-time performance of intelligent vehicle path tracking. This method reduced the lateral error by 56.6% compared with traditional MPC [16]. Existing trajectory tracking research often ignores the limitations of robot configuration manifold constraints. Tang et al. constructed a geometric MPC framework based on the mapping relationship between Lie groups and Lie algebras. The results showed that by convexly solving the error dynamics, this method achieved smoother trajectory tracking compared to nonlinear MPC [17]. However, traditional MPC overly relies on precise system mathematical models. Faced with unmodeled aggregate disturbances such as crosswinds and ground slippage in actual operating conditions, its internal linear prediction model will suffer serious mismatches, leading to steady-state tracking drift.

In summary, existing research has made progress in mobile robot path planning and trajectory tracking, but still faces many limitations. In terms of path planning, it is difficult for traditional sampling algorithms or single meta-heuristic algorithms to take into account search efficiency, path smoothness, and non-holonomic kinematic constraints of the robot. For trajectory tracking, existing control strategies often separate kinematics and dynamics. In response to this type of problem, a path planning and trajectory tracking control model based on improved RRT-MPC is constructed to accurately automatically navigate wheeled mobile robots.

The innovation of the research lies in proposing an improved multi-objective random adaptive RRT* (MRA-RRT*) algorithm to solve the smooth obstacle avoidance under incomplete constraints. Compared with baseline methods such as Informed RRT, this algorithm introduces an adaptive step size mechanism based on dynamic potential field at the bottom layer, which automatically shrinks the step size when approaching obstacles to improve spatial penetration rate. Meanwhile, the Nash balanced re-selection mechanism balances both path length and curvature penalties. In 50 independent Monte Carlo tests, the algorithm achieved a planning success rate of 98% in dense environments, and the average path smoothing cost was reduced by approximately 46.7% compared to Informed RRT*. In addition, a dual loop MPC robust tracking framework combining nonlinear disturbance observer (NDO) is developed. Different from the model dependency of standard MPC, this architecture utilizes NDO to exponentially estimate the total disturbance and feeds the compensation amount forward to the dynamic inner loop, dynamically correcting the prediction error of MPC. Under the composite interference condition of 5N lateral wind and variable friction coefficient, the root mean square error (RMSE) of the heading angle of the improved MPC is only 0.035 rad, which is 63.1% lower than that of the standard MPC.

2. Methodology

The study elaborates on the comprehensive navigation design method of wheeled mobile robots, which mainly includes global path planning and local trajectory tracking control. Firstly, the MRA-RRT* algorithm that integrates the adaptive sampling strategy and the Nash equilibrium re-selection mechanism is proposed. Secondly, an improved MPC

algorithm based on NDO and double closed-loop control is constructed.

2.1 Global Path Planning Algorithm Based on MRA-RRT*

2.1.1 Improvement Strategy and Adaptive Sampling

The wheeled mobile robot is a typical non-holonomic constraint system. Its path planning requires finding a collision-free geometric path in a two-dimensional plane while satisfying the kinematic constraints [18]. The RRT is largely applied in path planning of multi-degree-of-freedom robots because it does not explicitly model the state space and is probabilistically complete.

Nevertheless, the path generated by the standard RRT has defects such as large tortuosity and non-optimality, and it is difficult to satisfy the non-holonomic kinematic constraints of wheeled robots.

Therefore, RRT* is adopted, which introduces a parent node re-selection mechanism based on RRT. Although RRT* improves the path quality, it does not consider the non-holonomic constraints and dynamic smoothness. The study further proposes the MRA-RRT*, whose schematic diagram is shown in Figure 1.

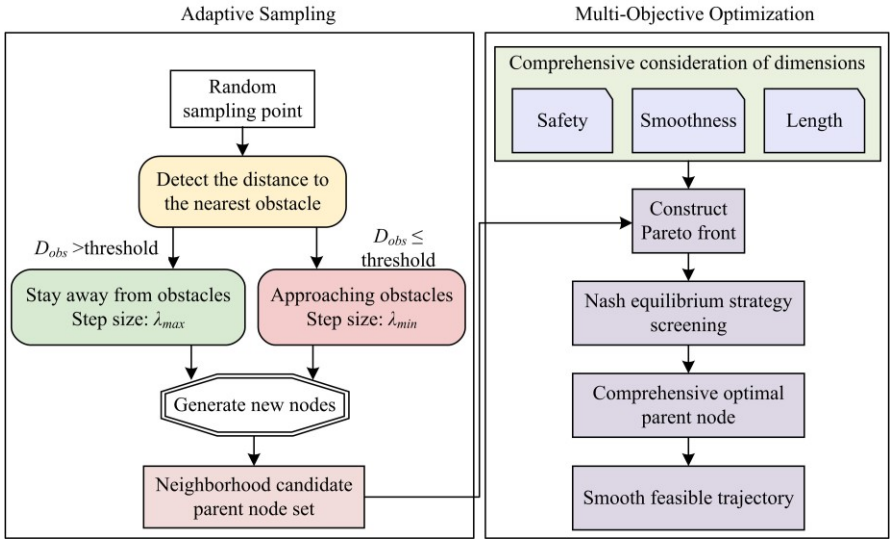


*Figure 1: Schematic diagram of MRA-RRT**

In Figure 1, the MRA-RRT* adopts an adaptive sampling strategy based on dynamic step size and a multi-objective optimization re-selection mechanism. In the adaptive sampling strategy, in areas far away from obstacles, the algorithm uses larger growth steps to improve search efficiency. In areas close to obstacles, the step length automatically shrinks for obstacle avoidance search. In the multi-objective optimization re-selection mechanism, during the node expansion and re-connection stages, the algorithm selects the parent node with the best comprehensive performance by establishing a Pareto front and using the Nash equilibrium strategy. The adaptive step sampling strategy is presented in Figure 2.

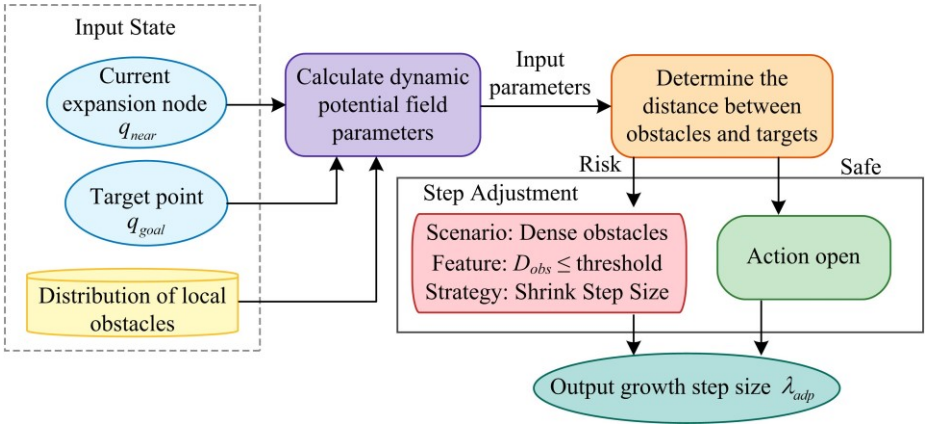


Figure 2: Schematic diagram of an adaptive step size sampling strategy

In Figure 2, the adaptive sampling strategy based on dynamic step size first obtains the current extended node q_{near}, target point q_{goal}, and local obstacle distribution information as input, and calculates the state parameters through the dynamic potential field module. Subsequently, the sampling strategy is divided into two branches by judging whether the distance between the current node and the obstacle exceeds the safety threshold. When the robot is in a safe zone away from obstacles, the system automatically selects a larger step size $\lambda_{\max}$. When the robot approaches the risk zone, the step length quickly shrinks to $\lambda_{\min}$. Finally, the two branches converge to output the dynamically adjusted growth step size λ_{adp}, whose expression is shown in equation (1).

$$\lambda_{adp}=\begin{cases}\lambda_{\min}+(\lambda_{\max}-\lambda_{\min})\cdot S(D_{goal},D_{obs}), & if\ \mu<P_{bias}\\ \lambda_{fix}, & else\end{cases} \tag{1}$$

In equation (1), $\lambda_{\min}$ and $\lambda_{\max}$ represent the bounds of the step size, respectively. λ_{fix} represents a fixed step size. μ represents random probability. P_{bias} is the target bias threshold. $S(\cdot)$ represents the Sigmoid adjustment function with respect to the target distance D_{goal} and the nearest obstacle distance D_{obs}, which is used to smooth the transition step change. In addition, since the wheeled robot is subject to non-holonomic constraints, the connections between nodes cannot simply use straight lines. The new node q_{new} requires integration based on the kinematic model, as presented in equation (2).

$$q_{new}=q_{near}+\int_0^T g(q(t),u_{inp})dt \tag{2}$$

In equation (2), $g(\cdot)$ signifies the state transition equation. u_{inp} signifies the control input of sampling. T signifies the total duration.

2.2 Multi-Objective Optimization Cost Function and Nash Equilibrium Re-selection

Simply pursuing the shortest geometric path can easily lead to the trajectory being close to obstacles, which increases the difficulty of subsequent control. To provide a high-quality reference trajectory for the back-end MPC controller, the study constructs a multi-dimensional cost function vector $C(q)$ containing path length, smoothness and safety, as shown in equation (3).

$$C(q)=[J_{len},J_{smooth},J_{safe}]^T \tag{3}$$

In equation (3), J_{len}, J_{smooth} and J_{safe} represent the path length cost, smoothness cost and security cost, respectively. The expression of J_{len} is shown in equation (4).

$$J_{len}(q_{new})=J_{len}(q_{parent})+\| q_{new}-q_{parent}\|_2 \tag{4}$$

In equation (4), q_{new} and J_{safe} represent the newly generated node and parent node, respectively. The expression of J_{smooth} is shown in equation (5).

$$J_{smooth}(q_{new},q_{parent})=\alpha_1|\theta_{new}-\theta_{parent}|+\alpha_2|\kappa_{new}| \tag{5}$$

In equation (5), θ_{new} and θ_{parent} are the heading angles at the nodes, respectively. κ_{new} is the path curvature at the new node. α_1 and α_2 are the weight coefficients of heading angle change and curvature, respectively.

This indicator effectively restricts the sharp turn of the path. The expression of J_{safe} is shown in equation (6).

$$J_{safe}(q_{new})=\sum_{k=1}^{M}\frac{\delta}{(dist(q_{new},O_k)-R_{safe})^2} \tag{6}$$

In equation (6), O_k signifies the position vector of the k-th obstacle. M signifies the number of obstacles within the detection range. δ signifies the repulsion gain coefficient. $dist(\cdot)$ represents the shortest Euclidean distance between the calculation node and the obstacle surface. R_{safe} signifies the preset safe expansion radius. In the RRT* re-selection stage, the study adopts the Nash equilibrium strategy.

In the neighborhood node set N_{near}, a Pareto front is established. The threat point D is defined, as shown in equation (7).

$$D=[J_{len}^{\max},J_{smooth}^{\max},J_{safe}^{\max}] \tag{7}$$

The node $q*$ that maximizes the profit product of all objective functions relative to the threat point is selected as the optimal parent node, and the expression is shown in equation (8).

$$q^*=\arg\max_{q\in N_{near}}\prod_{i\in\{len,smooth,safe\}}(D_i-J_i(q)) \tag{8}$$

The Nash equilibrium optimal node screening strategy based on Pareto frontier is shown in Figure 3.

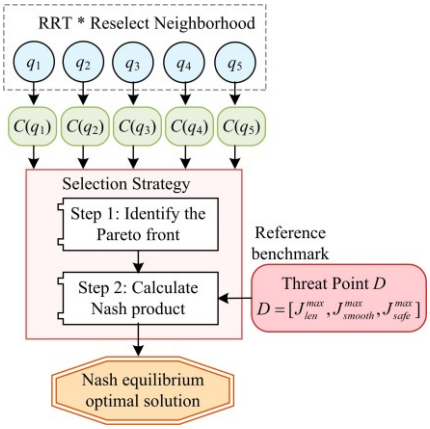


Figure 3: Nash equilibrium optimal node selection strategy based on Pareto frontier

From Figure 3, first, starting from multiple candidate parent nodes, they are mapped to a three-dimensional cost space, corresponding to different cost indicators. Next, obvious inferior solutions are eliminated by identifying the Pareto front. The threat point D is introduced as a reference benchmark to calculate the Nash product composed of the non-dominated solution set and the threat point. Finally, the node that maximizes the Nash product is selected as the Nash equilibrium optimal solution to find the best balance point among conflicting optimization goals.

2.3 Trajectory Tracking Control Algorithm Based on Improved MPC

2.3.1 Coupling Mechanism and Dynamic Modeling of RRT-MPC Model

The geometric path and reference speed planned by the end-MRA-RRT* are not only spatial guidance, but also a temporal status benchmark. Using this benchmark as a reference, the MPC controller predicts the future state of the robot within the rolling time domain and counts the optimal control law to eliminate tracking errors. To achieve high-precision trajectory tracking, a dynamic model for a wheeled mobile robot is established. The posture in the inertial coordinate system is shown in equation (9).

$$\eta = [x, y, \theta]^T \tag{9}$$

The kinematic equation is shown in equation (10).

$$\dot{\eta} = J(\theta)\nu = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{10}$$

In equation (10), $\nu = [v, \omega]^T$ represents the linear velocity and angular velocity vector. It is difficult to overcome slippage in high-speed operation or complex ground conditions by relying solely on kinematic models [19]. Therefore, the study introduces dynamic constraints. Considering ground friction, system inertia and unmodeled dynamics, the dynamic equation is established in equation (11).

$$M(\eta)\dot{\nu} + C(\eta, \nu)\nu + D(\nu) = B_\tau \tau + \tau_{ext} \tag{11}$$

In equation (11), M signifies the inertial mass matrix of the system. C signifies the Coriolis force and centripetal force matrix. D signifies the nonlinear friction damping term. B_τ signifies the moment input transformation matrix. τ signifies the motor torque of the left and right drive wheels. τ_{ext} signifies the lumped external disturbance including wind resistance, slip and model uncertainty. To apply linear MPC, the nonlinear system needs to be Taylor expanded and discretized at the reference trajectory point η_r at each moment. The error state quantity $\tilde{\eta} = \eta - \eta_r$ and the control increment $\tilde{u} = \nu - \nu_r$ are defined to derive the linear discrete error state space equation for predictive control, as shown in equation (12).

$$\tilde{\eta}(k+1) = \Phi_k \tilde{\eta}(k) + \Gamma_k \tilde{u}(k) + \Psi_k d(k) \tag{12}$$

In equation (12), Φ_k and Γ_k signify the Jacobian matrices about the system state and control input. $d(k)$ is the discretized disturbance term. Figure 4 presents the construction process of the linear error prediction model.

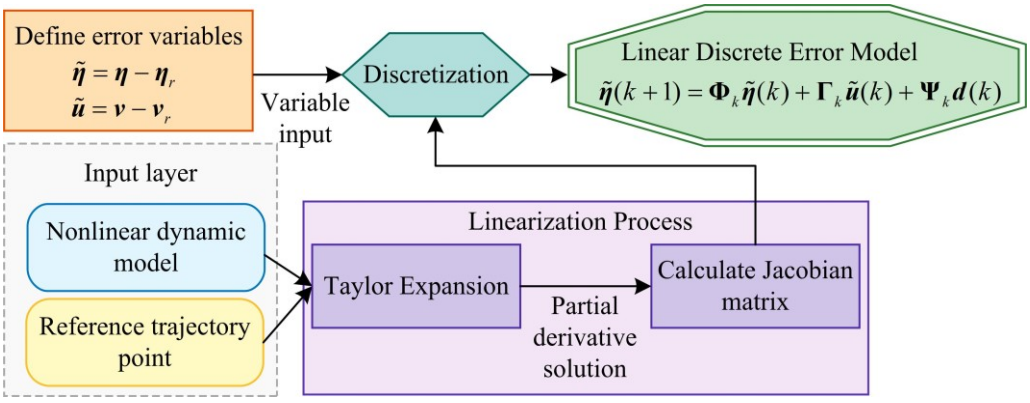


Figure 4: Construction process of linear error prediction model for wheeled mobile robot

In Figure 4, the process starts with the input layer, which contains the nonlinear dynamics equations and the reference trajectory as the linearized operating point. Subsequently, in the processing layer, the is linearized by performing a first-order Taylor series expansion around the reference point and calculating the Jacobian matrix of the system state and control inputs. Next, the continuous linear model is discretized by introducing sampling time to convert the differential equation into a difference equation. Finally, the output layer obtains a linear discrete error prediction model, which provides the prediction equation for the MPC solver.

2.4 NDO Design and Double Closed-Loop Control Architecture

In the actual operation of RRT-MPC, if there is a large deviation between the model prediction and the real environment, the kinematic instructions solved by MPC will not be accurately executed by the underlying driver [20]. Therefore, the study uses NDO to estimate lumped disturbances in real-time. The observer has exponential convergence characteristics and can quickly separate the disturbance components from changes in the system state. The structural block diagram of NDO is shown in Figure 5.

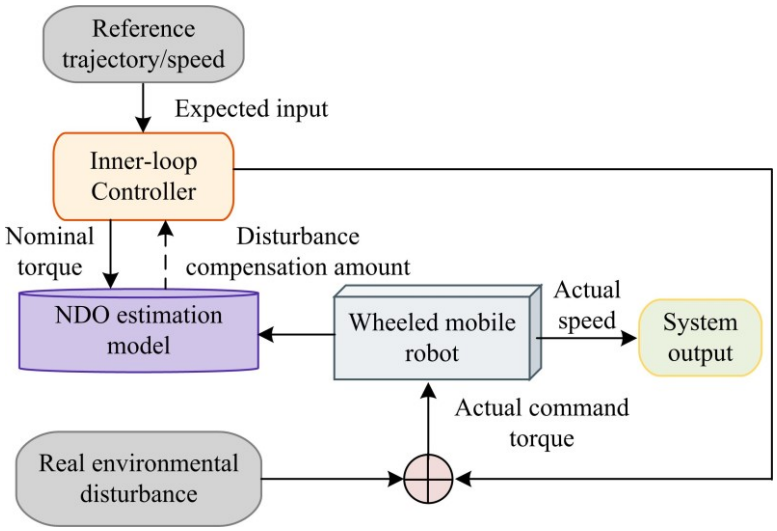


Figure 5: Structure diagram of NDO

In Figure 5, the NDO module is connected in parallel in the control loop, receives the speed feedback and torque input of the robot in real-time, and feeds the calculated disturbance compensation amount directly to the inner loop dynamics controller, thereby actively canceling environmental interference. To solve the excessive computational load and poor real-time performance when a single MPC directly solves complex dynamic constraints, a double closed-loop control architecture is built. The outer loop is kinematic-level MPC, which is responsible for eliminating trajectory tracking errors. The inner loop is a dynamics-level compensation controller, responsible for torque generation and interference rejection. For the outer loop MPC optimization problem, the reference trajectory generated by MRA-RRT* is mainly used as the benchmark. A quadratic objective function is constructed in the prediction time domain N_p to minimize the pose tracking error and control energy consumption. To distinguish it from the cost function in path planning, the quadratic objective function is shown in equation (13).

$$J(\tilde{\eta},\tilde{u}) = \sum_{i=1}^{N_p} \| \tilde{\eta}(k+i \mid k) \|_{W_Q}^2 + \sum_{j=0}^{N_c-1} \| \tilde{u}(k+j \mid k) \|_{W_R}^2 \quad (13)$$

In equation (13), W_Q and W_R signify the state error weight matrix and the control increment weight matrix. To ensure that the solved control instructions are within the physical execution capability of the robot, the optimization problem of equation (13) is transformed into a standard constrained quadratic programming (QP) problem. The system considers control quantity constraints and control increment constraints in the prediction time domain, and their mathematical expressions are shown in equation (14).

$$\begin{cases} U_{min} \le U(k) \le U_{max} \\ \Delta U_{min} \le \Delta U(k) \le \Delta U_{max} \end{cases} \quad (14)$$

In equation (14), $U(k)$ and $\Delta U(k)$ represent the control sequence and the control increment sequence, respectively. The upper and lower bounds are determined by the maximum speed and maximum acceleration of the motor.

In the specific implementation, the quadprog solver in MATLAB is used to solve the constrained QP problem online, and the maximum iteration number of the solver is set to 200, with a constraint tolerance of 1×10^{-3}, to ensure the real-time performance and convergence accuracy of the algorithm. In the inner loop dynamics compensation and torque generation link, the controller receives the desired speed command output from the outer loop and combines it with the disturbance estimate output by the NDO in real-time. Based on the inverse dynamics model, the controller calculates the torque command that drives the motor to achieve the response to the desired motion. Figure 6 presents the overall architecture diagram of the RRT-MPC.

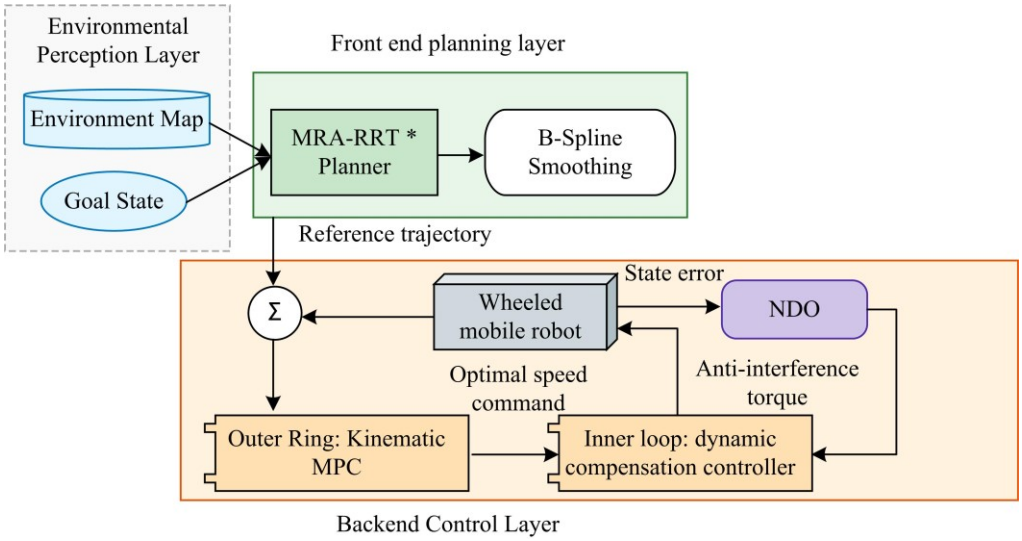


Figure 6: Overall architecture of RRT-MPC path planning and trajectory tracking control system

In Figure 6, the front-end MRA-RRT* algorithm is responsible for processing environment sensing data and generating reference trajectories. The outer loop MPC calculates the optimal speed control command based on the trajectory deviation. The inner loop controller combines NDO to generate anti-disturbance torque in real-time. This hierarchical control architecture ensures the high-precision geometric tracking capability of the planned path and enhances the robustness in response to changes in ground friction and external impacts.

In the closed-loop coupling integration of RRT-MPC, the system adopts a multi-rate operation mechanism. Firstly, the front-end MRA-RRT* planner runs offline when the global static map is known, generating a series of discrete spatial path points. Subsequently, the system uses cubic B-spline curves to smoothly interpolate these discrete points, and combines the maximum speed limit of the robot for time parameterization to generate a reference trajectory with timestamps. In the closed-loop control phase, the outer loop MPC controller samples the reference trajectory at a lower frequency (0.05s) and solves for the optimal control increment. The dynamic compensator containing NDO in the inner loop operates at a higher frequency (0.01s).

3. Results and Analysis

The study verified the effectiveness of the proposed comprehensive navigation framework for mobile robots through simulation experiments. First, the path planning performance of the MRA-RRT* in sparse and trap environments was analyzed.

Subsequently, the advantages of trajectory tracking control of the improved MPC were evaluated by setting extreme working conditions such as sudden lateral wind interference and variable friction coefficient roads.

3.1 Path Planning Analysis based on MRA-RRT*

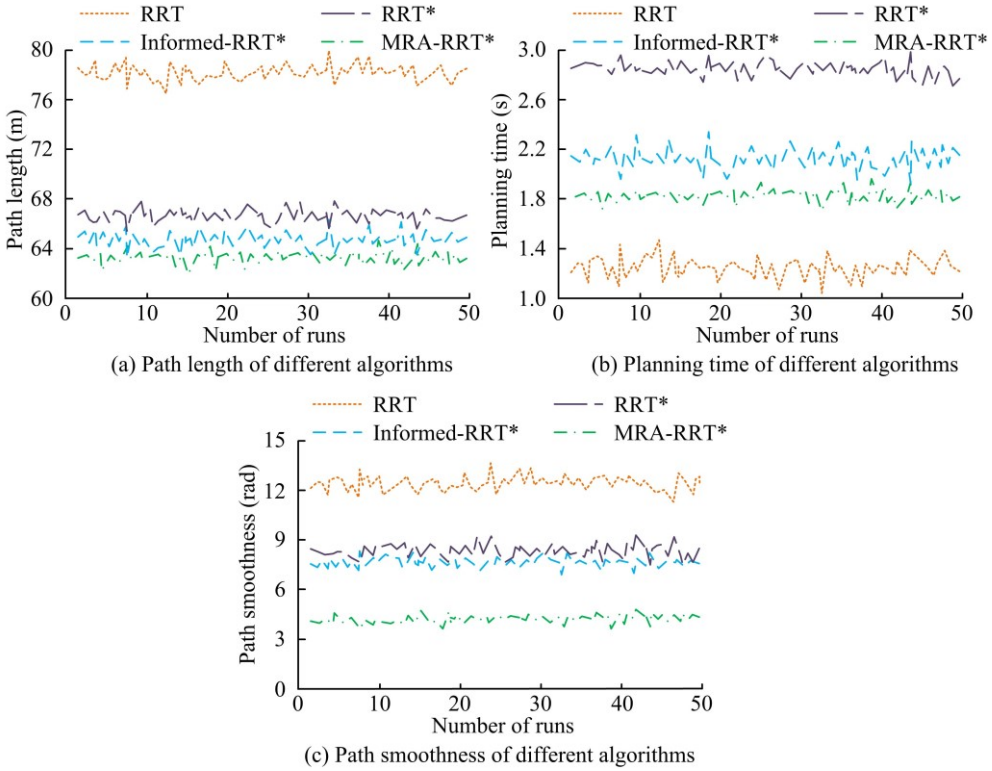
To verify the MRA-RRT* in wheeled mobile robot path planning, the study established a mobile robot motion planning test environment on the MATLAB R2023b simulation platform. The experiment uses a computer with Intel Core i7-12700H processor and 16GB memory for calculation. The experiment selected three obstacle environments of different levels of complexity to compare and test the MRA-RRT* with the traditional RRT, RRT* and Informed-RRT*. Each set of experiments was run independently 50 times in the same environment.

The path length, planning time, and path smoothness were collected. To ensure the effectiveness and reproducibility of simulation results, the core parameters and system constraint ranges of the algorithm are shown in Table 1.

Table 1: Key parameter settings of MRA-RRT and MPC controller*

Module	Parameter name	Value/Range
MRA-RRT*	Sampling step boundary	0.5 m-2.5 m
	Target bias probability	0.1
	Safe expansion radius	0.6 m
	Cost function weight	0.6, 0.4
MPC outer ring	Predictive Time Domain/Control Time Domain	15, 5
	Sampling time	0.05 s
	State error weight matrix	diag(50, 50, 20)
	Control incremental weight matrix	diag(0.1, 0.1)
Physical constraints	Linear velocity/angular velocity limit	1.5 m/s, 1.0 rad/s
	Acceleration limit	2.0 m/s^2

The experiment first simulated an open indoor warehouse environment with a map size of 50m×50m. There are six circular obstacles with different radii randomly distributed in the environment. The gaps between the obstacles are large and there are no complex trap areas. In a sparse environment, the performance indicators are presented in Figure 7. From Figure 7(a), the average path length planned by MRA-RRT* was the shortest, only 63.55m, while the path lengths of RRT, RRT* and Informed-RRT were 78.45m, 66.82m and 65.10m, respectively, which was obviously inferior to the MRA-RRT* algorithm. From Figure 7(b), the average planning time of MRA-RRT* was only 1.86s, while that of RRT* was 2.89s. From Figure 7(c), MRA-RRT* had the best path smoothness, with a smoothing cost of only 4.21rad, while that of Informed-RRT was 8.12rad. The MRA-RRT* algorithm comprehensively optimizes path quality, calculation efficiency and trajectory smoothness.



(a) Path length of different algorithms

(b) Planning time of different algorithms

(c) Path smoothness of different algorithms

Figure 7: Performance metrics of various algorithms in sparse environments

The path planning trajectories are presented in Figure 8. From Figure 8(a), the trajectory of the RRT showed obvious randomness and tortuousness, with many unnecessary turns, resulting in the longest path length and the worst smoothness. From Figure 8(b), compared with RRT, the RRT* optimized the path by introducing a re-selection mechanism, shortened the path length, and the trajectory was relatively smooth. However, since it still employs uniform random sampling, the search efficiency in open areas is not high, and the path still has certain twists and turns. In Figure 8(c), the Informed-RRT* limited the sampling area to an ellipse with the starting point and end point as the focus, which improved the search efficiency and further optimized the path. From Figure 8(d), the trajectory generated by the MRA-RRT* was the smoothest and straightest, almost close to the theoretical optimal path.

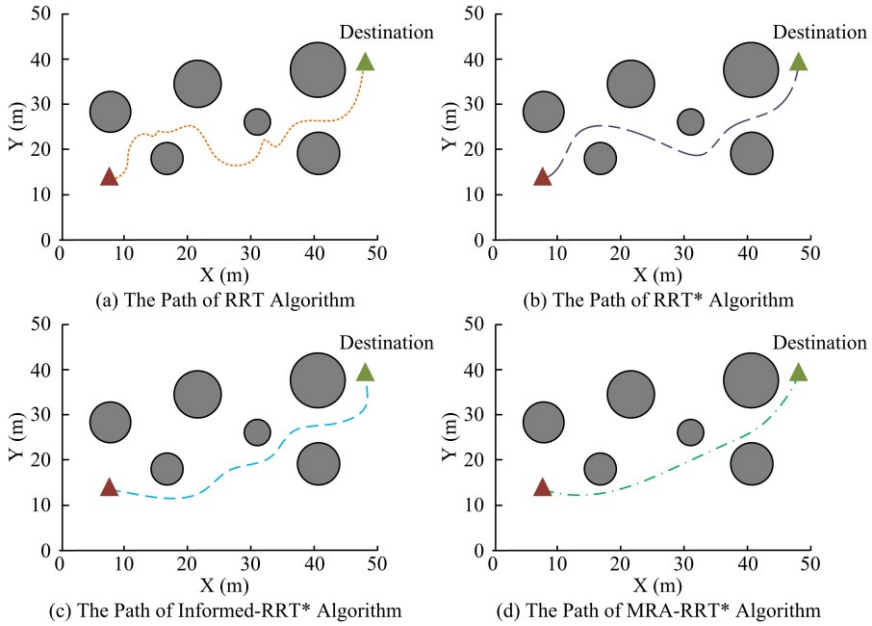


Figure 8: Path planning trajectories of various algorithms

The study verified the ability to escape local minima. The map contained a huge U-shaped depression obstacle, with the starting point located inside the U-shaped opening and the end point located behind the obstacle. Ordinary algorithms tend to fall into greedy search toward the goal and cannot escape the trap. In the U-shaped trap environment, the average performance indicators are presented in Figure 9. From Figure 9(a), in the U-shaped trap environment, the path length planned by MRA-RRT* was the shortest, only 46.21m, and the planning time was only 3.12s.

The path length obtained from the RRT was 62.15m. The RRT* algorithm took the longest, reaching 8.67s. From Figure 9(b), the path smoothing of MRA-RRT* algorithm was only 5.33rad, which was much lower than that of RRT (18.92rad) and Informed-RRT* (10.44rad).

The planning success rate of RRT was only 82.31%, while that of MRA-RRT* was 100%, verifying its ability to quickly escape traps and generate high-quality paths in highly constrained environments.

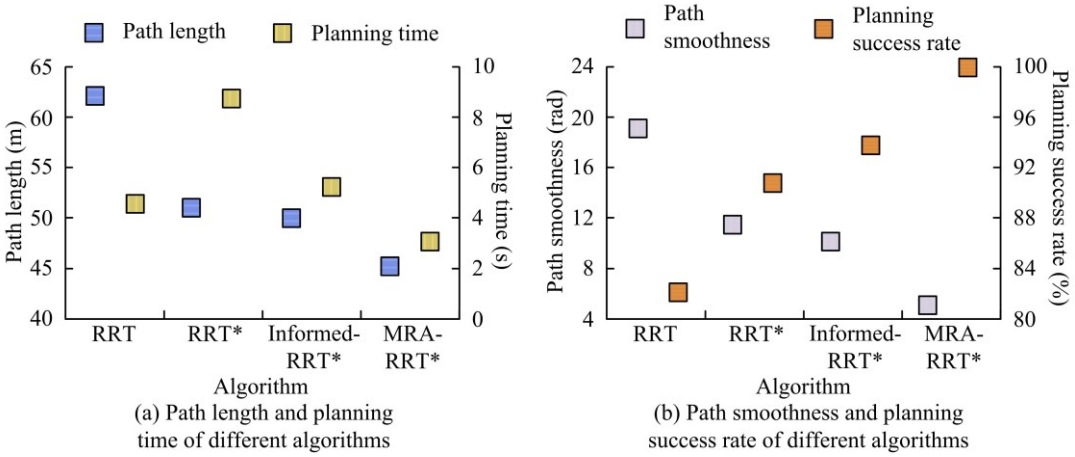


Figure 9: Performance of various algorithms in U-shaped trap environment

To further verify the adaptability of the algorithm in extremely confined spaces, high-density random obstacle environment test cases were introduced in the study. The map size is set to 50m×50m and 45 static obstacles of different sizes are randomly generated, achieving a local obstacle coverage rate of approximately 35%. To eliminate the accidental errors caused by random sampling, all four algorithms were independently run 50 times in this environment, and their average planning time, average path length, smoothing cost, and planning success rate were calculated. The test results are shown in Table 2. In dense environments, the planning success rates of traditional RRT and RRT decreased to 64.0% and 72.0%, respectively. Although Informed RRT optimizes the search space,

it still appears cramped in narrow channels. In contrast, MRA-RRT* maintained a high success rate of 98.0%. Its average planning time was only 4.15 seconds, far lower than that of Informed RRT (7.82s). This indicates that the algorithm has superior robustness and search efficiency under different map densities.

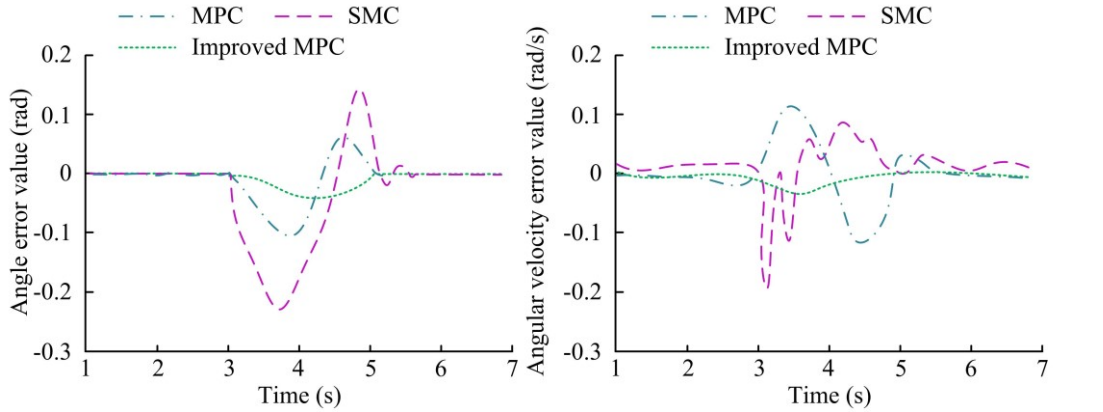
Table 2: Comprehensive Performance Comparison of Different Algorithms in High Density Obstacle Environments

Algorithm model	Planning success rate (%)	Average planning time (s)	Average path length (m)	Smoothing cost (rad)
Traditional RRT	64.0	12.18	95.24	45.22
RRT*	72.0	15.45	78.56	25.41
Informed-RRT*	85.0	7.82	75.13	12.35
MRA-RRT*	98.0	4.15	72.31	6.58

3.2 Trajectory Tracking Control Performance based on Improved MPC

To verify the effectiveness of the improved MPC based on NDO and double closed - loop control in wheeled mobile robot trajectory tracking, the study selected three trajectory tracking scenarios of different levels of complexity, and compared the improved MPC algorithm with standard MPC and SMC algorithms. The study first tested the wheeled mobile robot's tracking ability on a simple straight trajectory and its robustness under sudden lateral interference, as scenario 1. Starting from the origin, a linear motion is made with uniform acceleration in the positive direction of the X-axis, and the target speed is 1.0m/s. During the period from t=3s to t=5s, a 5N lateral wind interference force is applied.

The angle and speed tracking errors are presented in Figure 10. From Figure 10(a), the SMC algorithm was limited by the switching characteristics of the sliding mode surface, and its peak heading angle error reached a maximum of 0.23rad. Followed by MPC, the peak heading angle error was 0.12rad. The peak heading angle error of the improved MPC was only 0.04rad. From Figure 10(b), the angular velocity error curve of the SMC algorithm showed severe high-frequency buffeting, and the peak error reached 0.19rad/s. The peak angular velocity error of the improved MPC was only 0.03rad/s. The improved MPC can converge to the reference trajectory at the fastest speed and with the smallest error while ensuring the smoothness.

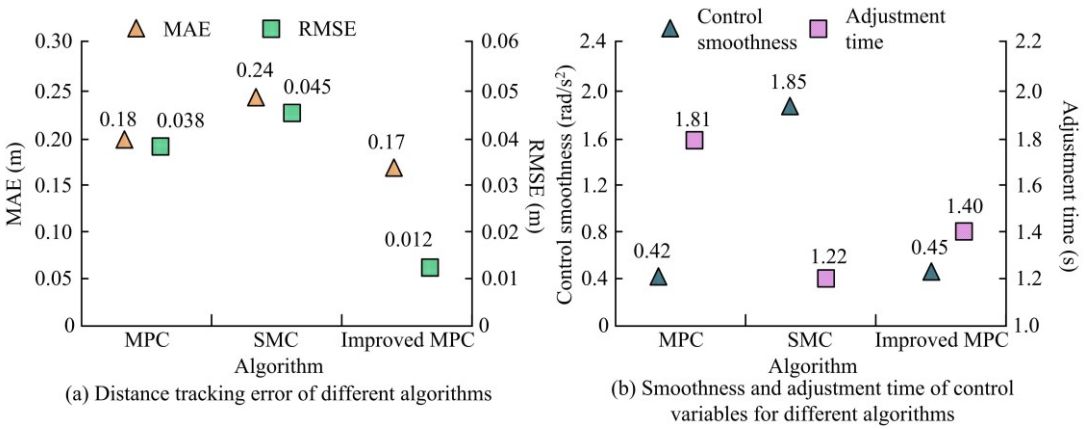


(a) Angle error value of different algorithms (b) Angular velocity error value of different algorithms

Figure 10: Angle and velocity tracking errors of various algorithms in scenario 1

The study compared the maximum absolute error (MAE), RMSE, control smoothness and adjustment time of each algorithm from t=3s to t=5s, as presented in Figure 11. From Figure 11(a), the MAE and RMSE of the SMC algorithm were 0.24m and 0.045m, respectively. The standard MPC lacks prediction and compensation for lateral interference forces, resulting in steady-state drift, with MAE and RMSE of 0.18m and 0.038m, respectively. The improved MPC uses real-time compensation of NDO to reduce the MAE to 0.17m, RMSE to only 0.012m, and the tracking accuracy is significantly improved.

From Figure 11(b), SMC had the worst control smoothness, at 1.85rad/s^2, indicating that its control action was extremely violent. Although the standard MPC had the optimal smoothness, its adjustment time was the longest, reaching 1.81s. The control smoothness of the improved MPC was 0.45rad/s^2, which was far better than that of SMC, and the adjustment time was only 1.40s. The improved MPC has the ability to quickly suppress sudden interference while ensuring the smooth operation of the actuator.

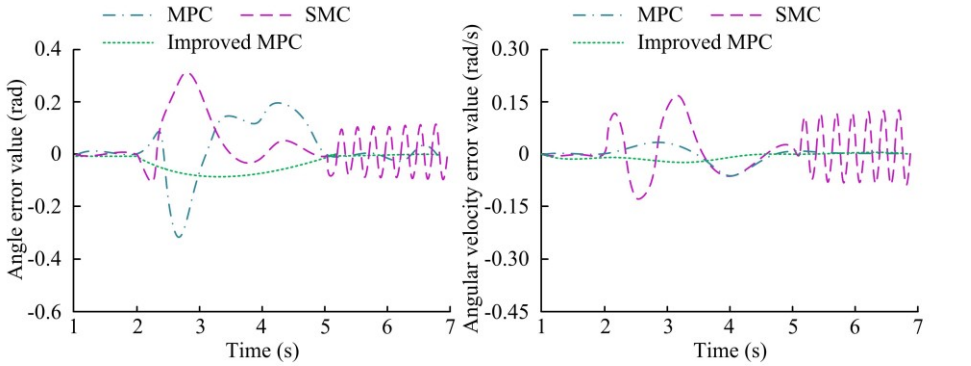


(a) Distance tracking error of different algorithms

(b) Smoothness and adjustment time of control variables for different algorithms

Figure 11: Tracking distance error, smoothness of control variables, and adjustment time of various algorithms

The study simulated the scenario of a wheeled mobile robot performing continuous steering on the ground with different friction coefficients, as scenario 2. The running trajectory is a circular trajectory with a radius of 2m, and the linear speed is 0.8m/s. From t=2s to t=5s, the ground friction coefficient suddenly halved, causing the tire lateral grip to decrease. The angle error and angular velocity error are presented in Figure 12. From Figure 12(a), when side-slip interference caused by a sudden drop in the friction coefficient occurred at t=2s, the SMC algorithm experienced severe jitter, and the maximum angle error peak exceeded 0.3rad. However, the improved MPC algorithm always maintained the tracking error within a very small range throughout the disturbed interval, with a maximum error of only 0.08 rad. From Figure 12(b), the maximum angular velocity error value of the improved MPC algorithm was only 0.04rad/s. The improved MPC algorithm has optimal dynamic anti-interference ability and control stability.



(a) Angle error value of different algorithms

(b) Angular velocity error value of different algorithms

Figure 12: Angle and velocity tracking errors of various algorithms in scenario 2

During the operation of a real mobile robot chassis, there is inevitably measurement noise from its wheel odometer and positioning sensor. To closely simulate the actual physical working conditions, a more challenging scenario 3 was designed: requiring the robot to track a complex "8" - shaped reference trajectory, which simultaneously tests the robot's continuous steering and variable acceleration capabilities. In the control loop, Gaussian white noise $\mathrm{N}(0,0.02^2)$ with a standard deviation of 0.02 was injected into the position state feedback signal, and Gaussian white noise $\mathrm{N}(0,0.05^2)$ with a standard deviation of 0.05 was injected into the heading angle. An unmodeled composite external disturbance $d(t)=3\sin(0.5t)+2\$N$ that varied over time was applied from t=5 s to t=15 s.

The error statistics of the three controllers are shown in Table 3. According to Table 3, when high-frequency measurement noise and time-varying interference were introduced into the system, the traditional SMC algorithm was extremely sensitive to noise. High-frequency noise excites severe high-frequency chattering of the sliding surface, resulting in the worst smoothness of its control variable, with a value of 4.85 rad/s². Although the standard MPC has certain filtering characteristics, due to the lack of explicit processing for unmodeled composite interference, the MAE can reach up to 0.254m. The improved MPC had a heading angle RMSE of only 0.035rad, which was about 77.0% lower than that of SMC, verifying the robustness of the dual closed-loop control architecture under harsh conditions.

Table 3: Comparison of Tracking Error Indicators of Various Control Algorithms in Noise and Composite Interference Environments

Control algorithm	MAE (m)	Heading angle RMSE (rad)	Angular velocity RMSE (rad/s)	Control smoothness (rad/s ^{2})
SMC	0.185	0.152	0.224	4.85
MPC	0.254	0.095	0.128	1.42
Improved MPC	0.082	0.035	0.041	0.65

4. Conclusions and Recommendations

To address the autonomous navigation of wheeled mobile robots in complex unstructured environments, the study constructed an improved RRT-MPC control framework that integrated global path planning and local trajectory tracking. In a sparse obstacle environment, the path length generated by the MRA-RRT* was only 63.55m, the smoothing cost was only 4.21rad, and the planning time was only 1.86s, which was about 35% shorter than that of the traditional RRT. The Nash equilibrium strategy based on Pareto front found the best balance point between path length and smoothness. In the U-shaped trap scenario, MRA-RRT maintained a 100% planning success rate, while traditional RRT was only 82.31%. In the experiment where straight-line trajectory tracking was interfered by 5N lateral wind, the heading angle error of the improved MPC was only 0.012m. In the variable working condition test where the road friction coefficient was halved, the SMC algorithm experienced severe buffeting, and the angle error exceeded 0.30rad. In contrast, the improved MPC used NDO to quickly estimate the lumped disturbance and directly compensated it by the inner loop, limiting the maximum angular error to within 0.08rad, and the peak angular velocity error was only 0.04rad/s. The improved RRT-MPC model has obvious performance advantages in the autonomous navigation. Although the proposed improved RRT-MPC model performs well under various rigorous simulation conditions, validation work is still limited to simulation environments. Non-ideal factors such as actuator dead zones, communication delays, and dynamic obstacles in real physical environments have not been fully considered. Therefore, future research considers introducing dynamic obstacle trajectory prediction into MPC to achieve dynamic obstacle avoidance. The algorithm can be deployed on a real mobile robot platform based on ROS and tested on real vehicles in complex unstructured scenarios to further verify the engineering applicability of the navigation framework.

References

[1] Cong, V.D. (2023). Path following and avoiding obstacle for mobile robot under dynamic environments using reinforcement learning. *Journal of Robotics and Control (JRC)*, 4(2), 157-164. https://doi.org/10.18196/jrc.v4i2.17368

[2] Wang, J., Fader, M.T.H., & Marshall, J.A. (2023). Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization. *Journal of Field Robotics*, 40(5), 1014-1033. https://doi.org/10.1002/rob.22165

[3] Wellendorf, A., Tichelmann, P., & Uhl, J. (2023). Performance analysis of a dynamic test bench based on a linear direct drive. *Archives of Advanced Engineering Science*, 1(1), 55-62. https://doi.org/10.47852/bonviewAAES3202902

[4] Bouaziz, N.I., Mechali, O., Besseghieur, K.L., & Achour, N. (2024). Trajectory planning for autonomous formation of wheeled mobile robots via modified artificial potential field and improved PSO algorithm. *Unmanned Systems*, 12(06), 1085-1104. https://doi.org/10.1142/S2301385025500372

[5] Mohanraj, T., Dinesh, T., Guruchandhramavli, B., Sanjai, S., & Sheshadhri, B. (2023). Mobile robot path planning and obstacle avoidance using hybrid algorithm. *International Journal of Information Technology*, 15(8), 4481-4490. https://doi.org10.1007/s41870-023-01475-5

[6] Zou, A., Wang, L., Li, W., Cai, J., Wang, H., & Tan, T. (2023). Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach. *The Journal of Supercomputing*, 79(8), 8340-8367. https://doi.org/10.1007/s11227-022-04998-z

[7] Paital, C., Kumar, S., & Muni, M.K. (2023). Navigation of a wheeled mobile robotic agent using modified grey wolf optimization controller. *International Journal of Intelligent Unmanned Systems*, 11(2), 197-213. https://doi.org/10.1108/IJIUS-06-2020-0023

[8] Zhang, J.X., Ding, J., & Chai, T. (2024). Fault-tolerant prescribed performance control of wheeled mobile robots: A mixed-gain adaption approach. *IEEE Transactions on Automatic Control*, 69(8), 5500-5507. https://doi.org/10.1109/TAC.2024.3365726

[9] Li, J., Wang, J., Wang, S., & Yang, C. (2023). Human–robot skill transmission for mobile robot via learning by demonstration. *Neural Computing*

and Applications, 35(32), 23441-23451. https://doi.org/10.1007/s00521-021-06449-x

[10] Hassan, I.A., Abed, I.A., Al-Hussaibi, W.A. (2024). Path planning and trajectory tracking control for two-wheel mobile robot. *Journal of Robotics and Control (JRC)*, 5(1), 1-15. https://doi.org/10.18196/jrc.v5i1.20489

[11] Yuan, L., Zhao, J., Li, W., & Hou, J. (2023). Improved informed-RRT* based path planning and trajectory optimization for mobile robots. International Journal of Precision Engineering and Manufacturing, 24(3), 435-446. https://doi.org/10.1007/s12541-022-00756-6

[12] Yu, L., Han, J., Ma, B., Liu, B., & He, Z. (2025). Integration of improved APF and RRT algorithms for enhanced path planning in mobile robotics. *Measurement and Control*, 58(4), 427-434. https://doi.org/10.1177/00202940241268612

[13] Li, X., Lu, Y., Zhao, X., Deng, X., & Xie, Z. (2024). Path planning for intelligent vehicles based on improved D* Lite. The Journal of Supercomputing, 80(1), 1294-1330. https://doi.org/10.1007/s11227-023-05528-1

[14] Kordi, F., Mobayen, S., Rezaalikhani, H., & Nikoukar, J. (2025). Hybrid trajectory tracking control of wheeled mobile robots using predictive kinematic control and dynamic robust control. *Asian Journal of Control*, 27(3), 1203-1223. https://doi.org/10.1002/asjc.3503

[15] Moudoud, B., Aissaoui, H., & Diany, M. (2023). Adaptive integral-type terminal sliding mode control: Application to trajectory tracking for mobile robot. *International Journal of Adaptive Control and Signal Processing*, 37(3), 603-616. https://doi.org/10.1002/acs.3540

[16] Meng, Q., Qian, C., Chen, K., Sun, Z.Y., Liu, R., & Kang, Z. (2024). Variable step MPC trajectory tracking control method for intelligent vehicle. *Nonlinear Dynamics*, 112(21), 19223-19241. https://doi.org/10.1007/s11071-024-10042-x

[17] Tang, J., Wu, S., Lan, B., Dong, Y., Jin, Y., Tian, G., & Shi, L. (2024). GMPC: Geometric model predictive control for wheeled mobile robot trajectory tracking. *IEEE Robotics and Automation Letters*, 9(5), 4822-4829. https://doi.org/10.1109/LRA.2024.3381088

[18] Dai, S.L., Liang, J., Lu, K., & Jin, X. (2023). Adaptive image-based moving-target tracking control of wheeled mobile robots with visibility maintenance and obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 32(2), 488-501. https://doi.org/10.1109/TCST.2023.3331553

[19] Hung, N., Rego, F., Quintas, J., Cruz, J., Jacinto, M., Souto, D., & Pascoal, A. (2023). A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments. *Journal of Field Robotics*, 40(3), 747-779. https://doi.org/10.1002/rob.22142

[20] Su, X., Qing, F., Chang, H., & Wang, S. (2023). Trajectory tracking control of human support robots via adaptive sliding-mode approach. *IEEE Transactions on Cybernetics*, 54(3), 1747-1754. https://doi.org/10.1109/TCYB.2023.3253171