ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Research on adaptive control of path planning and trajectory tracking in complex environment

Fengzhen Chen[1,*], *Wanfang Lin*[1] *and Tingting Wu*[1]

[1] Department of Information Engineering of Meizhouwan Vocational Technology College
PuTian 351119, Fujian, China

SUMMARY: *In order to solve the problems of low efficiency of path planning in complex environments, disconnection between local obstacle avoidance and trajectory execution, and easy amplification of tracking error in dynamic disturbances, this paper proposes an integrated adaptive control method for path planning and trajectory tracking. In the planning layer, guided sampling and risk cost field modeling were introduced to improve the search efficiency, path smoothness and reachability under complex obstacle distribution. In the control layer, an adaptive tracking structure including feedforward compensation, error feedback and online gain adjustment is constructed to enhance the system's ability to suppress model mismatch and external disturbance. The experimental results show that in the scene with 25% obstacle density, the average planning time of the proposed method is 0.43 s, the average path length is 78.4 m, and the task success rate reaches 96.7%. When the obstacle density increases to 35%, the success rate still maintains 91.5%. In the field test, the average tracking error in the dynamic obstacle scene is 0.038 m, the maximum lateral error is 0.074 m, and the replanning recovery time is 0.93 s. The results show that the proposed method achieves a good balance between path quality, control ride comfort and environmental adaptability.*

KEYWORDS: *complex environment; Path planning; Trajectory tracking; Adaptive control*

1 Introduction

With the continuous landing of mobile robots, unmanned vehicles and intelligent inspection equipment in scenarios such as storage and transportation, park services, complex manufacturing and emergency operations, the requirements for autonomous movement ability of the system no longer stop at "being able to reach the target position", but further shift to "reaching the target position stably, in real time and safely in complex environments". Complex environments are usually characterized by irregular distribution of obstacles, narrow access areas, frequent local dynamic disturbances, and enhanced model uncertainty. This makes path planning and trajectory tracking no longer two isolated problems, but a computational process involving environmental modeling, state estimation, planning decision and closed-loop control. If only the geometric reachability of the path is emphasized, and the dynamic constraints and external disturbances of the vehicle or robot execution layer are ignored, the planning results are often difficult to be smoothly reproduced in the actual operation. On the other hand, if the controller only tracks around a given trajectory and lacks the fast response ability to environmental changes and path reconstruction, the system is prone to obstacle hysteresis, local oscillation and even tracking instability.

*minyi0730@163.com
https://doi.org/10.65102/is2026220
Published: 30 April, 2026

Around this problem, domestic and foreign research has formed a rich technical accumulation in the direction of graph search, sampling planning, artificial potential field, dynamic window, reinforcement learning and model predictive control [1]. The existing methods can achieve good results in structured scenes or single task conditions. However, when facing the application scenarios with dense obstacles, incomplete environmental information, and high requirements for path curvature continuity, some common shortcomings are still exposed. The other kind of method has strong real-time performance, but is easily affected by factors such as local optimum, speed jitter and control unsmoothness [2]. More notably, many studies deal with path planning and trajectory tracking in segments, and the discrete paths output by the planning layer lack deep coupling with the control layer, resulting in an obvious fracture between path feasibility, tracking accuracy and control stability, which is also an important reason why the system performance is difficult to be further improved in complex environments [3].

Based on this, this thesis focuses on the integrated goal of "path planning, trajectory generation, adaptive tracking control", and attempts to construct a unified method framework for complex environments from the perspective of computer modeling and control cooperation. In the research, the environment grid expression, obstacle constraint judgment, path cost evaluation and trajectory state update are integrated into the same computing link, which improves the feasibility and continuity of path search in the planning layer. The adaptive adjustment mechanism is introduced in the control layer to correct the speed error, heading deviation and external disturbance online. Thus, the tracking stability of the system is enhanced under the condition of dynamic disturbance and parameter perturbation. The focus of this paper is not simply to shorten the path length or reduce the instantaneous error, but to achieve the coordination and unification between planning quality, control ride comfort and real-time response capability in a multi-constrained environment.

The purpose of this paper is to verify whether the co-design of the path planning model and the trajectory tracking adaptive control strategy can improve the path quality, tracking accuracy and operation robustness of the system more effectively than the traditional separated method under complex environmental constraints. The follow-up content will focus on the control structure design and parameter optimization, the path planning model construction in complex environments, the integrated implementation mechanism and performance verification, in order to provide engineering feasible technical reference for the autonomous movement of mobile agents in non-ideal environments.

2 Related Research

Path planning and trajectory tracking control in complex environments have gradually developed from "single path search" to the coupled research direction of "environment modeling-decision generation-closed-loop execution". The core difficulty is not whether a feasible path can be obtained, but whether the system can maintain high planning quality, small tracking error and sufficient real-time response ability when the obstacle distribution, scene topology, vehicle motion constraints and external disturbances change simultaneously. Around this problem, the existing research mainly advances along three technical routes: path planning optimization, trajectory tracking control enhancement, and planning and control collaborative design, and has formed a rich application results in mobile robots, unmanned vehicles, and complex operating platforms.

At the path planning level, the research focuses on global search efficiency, local obstacle avoidance flexibility and path smoothness improvement. Liu et al. [4] conducted a systematic review of mobile robot path planning technologies, and pointed out that traditional graph search and heuristic methods have stability advantages in clearly structured scenes, but are often limited by problems such as high frequency of re-planning and insufficient local adaptation in dynamic environments. Yang et al. [5] combined the improved dynamic window method with the A* algorithm to establish a connection between local obstacle avoidance and global guidance, and enhanced the online planning ability in indoor environments. Hou et al. [6] further introduced the improved sparrow search algorithm into DWA to optimize the local trajectory evaluation process and improve the path search quality under complex obstacle distribution conditions. In addition to heuristic methods, deep reinforcement learning has also been used for local path generation. Park et al. [7] and Tao et al. [8] implement path decision in dynamic environment based on DDPG and deep reinforcement learning framework respectively, so that the system has certain self-learning ability, but these methods usually rely on high-quality training samples and long training period. There is still instability in generalization across scenarios. Zhang et al. [9] combined SAC with LSTM to deal with the local planning problem in dynamic indoor scenes, which shows that the learning method has the potential to deal with time-dependent environmental changes, but its interpretability and engineering deployment cost are still worthy of attention.

For trajectory tracking control, research has been extended from classical PID, sliding mode control to model predictive control, adaptive control, and data-driven compensation control. Sun et al. [10] proposed an autonomous vehicle path following system based on model predictive control, which can balance lateral error correction and control ride comfort under constraints. Liu et al. [11] introduced an adaptive MPC mechanism to solve the model mismatch problem caused by scene changes and improved the tracking stability of autonomous vehicles under variable driving conditions. Wang et al. [12] used the state expansion method to construct an adaptive trajectory tracking controller, which still maintained good error convergence performance in the presence of external disturbances and parameter perturbations. Compared with the control method relying on accurate models, Liu et al. [13] used Gaussian process regression to compensate the unmodeled dynamics of wheeled mobile robots, so that the controller could correct the prediction deviation in the uncertain environment, which indicates that data-driven modeling is gradually becoming an important means to enhance the robustness of control. However, the controller optimization alone cannot solve the inconsistency problem between the planning path and the execution trajectory. When the planning result has curvature mutation or local untraceability segments, the upper bound of the control layer performance will still be significantly constrained.

In recent years, the integration of planning and control has gradually become a research hotspot. Marashian et al. [14] simultaneously dealt with the path planning and path following problems of mobile robots from the perspective of dynamic programming, trying to weaken the separation error between them. Feng et al. [15] integrated obstacle avoidance path planning and trajectory tracking into the operation process of automated guided vehicles for the scenario of automated terminals, reflecting the actual demand for collaborative design in engineering systems. Zhang et al. [16] combined the improved RRT* and PSO-LQR for path planning and tracking control of autonomous vehicles, improving the execution stability while ensuring the feasibility of the path. Zhang et al. [17] improved the combination of artificial potential field and sliding mode control to realize the linkage of local obstacle avoidance and trajectory tracking. Zhang et al. [18] further constructed an adaptive planning and tracking system based on MPC, which made local obstacle avoidance and trajectory execution complete under a unified prediction framework, showing good dynamic adaptive ability. This kind of research has obviously broken through the series mode of "planning before, control after", but there are still two common problems. First, the connection between the path cost function and the control performance index is not close enough, and the planning layer often pays more attention to the length or safety margin, but does not consider the traceability enough. Second, the scene update in complex environment is uncertain and sudden, and the existing methods are still not sufficient in real-time reconstruction of control parameters and planning strategies. To summarize the representative work more clearly, Table 1 provides an inductive comparison of related studies.

Table 1: Comparison of representative methods of related studies

Reference	Core Method	Research Object	Main Advantages	Main Limitations
[5]	Improved DWA + A*	Indoor navigation of mobile robots	Balances global guidance and local obstacle avoidance, with relatively good real-time performance	Limited path smoothness and dynamic adaptation capability
[9]	Deep reinforcement learning for local planning	Mobile robots in dynamic environments	Possesses online decision-making and self-learning capability	High training cost, and generalization stability is affected by the scenario
[14]	Dynamic programming for path planning and tracking	Mobile robots	Focuses on the coordination between planning and tracking	Relatively heavy computational burden, with limited scalability in complex scenarios
[16]	Improved RRT* + PSO-LQR	Autonomous vehicles	Improves path feasibility and control stability	Strong parameter coupling, making online adjustment more complex
[18]	Adaptive planning-tracking MPC	Local obstacle avoidance for autonomous vehicles	Strong predictive capability and suitable for constrained control	Depends on model accuracy, with relatively high pressure for real-time solving
[13]	Gaussian process regression + robust adaptive control	Wheeled mobile robots	Can compensate for unmodeled dynamics and provides good robustness	Requires high sample quality and regression efficiency

Based on the existing results, it can be seen that the key contradiction of motion control in complex environments is no longer the pursuit of shorter paths or smaller errors, but how to achieve a balanced improvement of planning quality, tracking accuracy and computational efficiency under the condition of environmental uncertainty, dynamic changes of obstacles and system parameter perturbation. Based on this understanding, this paper puts the path planning model construction and trajectory tracking adaptive control into a unified framework, and tries to narrow the gap between "path planning" and "trajectory executable" by introducing traceability constraints in the planning layer and online parameter adjustment mechanism in the control layer. Such a research idea also constitutes the direct starting point for the design of the method below.

3 Path planning and trajectory tracking adaptive control method in complex environment

3.1 Trajectory tracking adaptive control structure design and control parameter optimization

Trajectory tracking control in complex environments is not only to make the moving object move forward along a given curve, but also to maintain the cooperative convergence of position error, heading error and velocity error under the conditions of obstacle disturbance, ground attachment change, local curvature mutation and actuator response lag. If the controller still adopts a fixed parameter structure, although a relatively stable output can be obtained under a single working condition, once the reference trajectory is switched from a straight line to a sharp curve, or temporary obstacles in the environment cause local replanning, the system is prone to problems such as steering lag, lateral swing and speed adjustment discontinuity. Based on this, a four-layer coupling structure of "reference trajectory feedforward - error feedback correction - parameter online adaptation - disturbance compensation correction" is constructed under the digital control framework, so that trajectory tracking does not rely on a single set of static control parameters, but is continuously adjusted according to the current operating state.

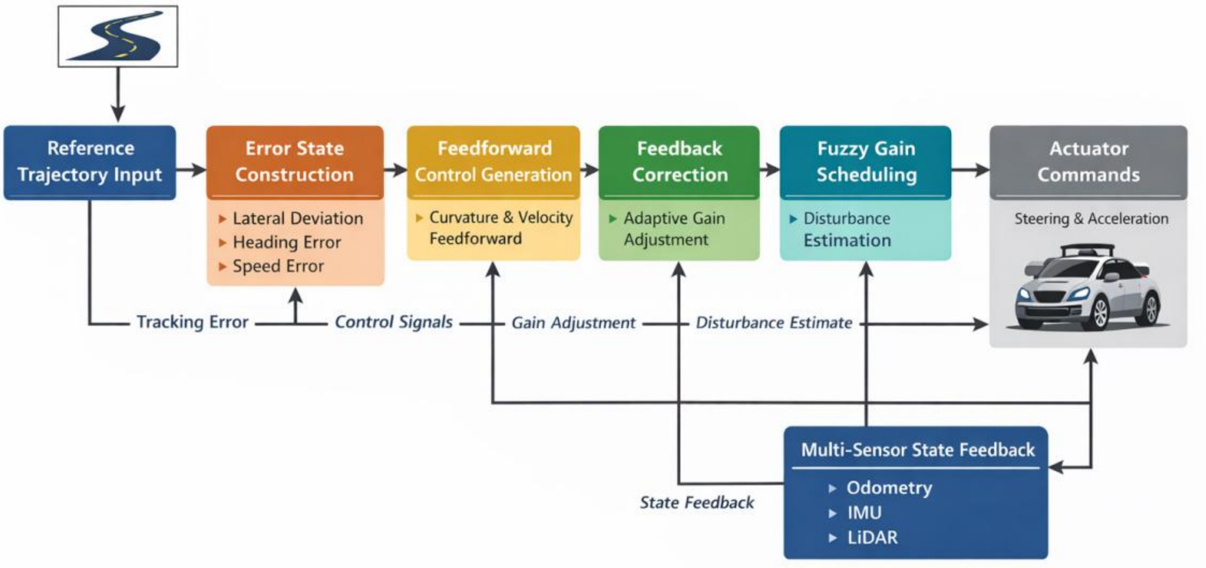

Figure 1: Overall structure of adaptive control for trajectory tracking in complex environments

As shown in Figure 1, the system takes the discrete reference trajectory sequence output by the path planning layer as input, and combines the fusion positioning results of the on-board odometer, inertial measurement unit and lidar to construct the tracking error state in real time. In order to make the controller reflect the geometric deviation and the kinematic deviation at the same time, the lateral error, the heading Angle error and the longitudinal velocity error are uniformly written as the state vector:

$$e_k = [e_y(k) \quad e_\psi(k) \quad e_v(k)]^T \tag{1}$$

Here, $e_y(k)$ represents the lateral offset at time k, $e_\psi(k)$ represents the Angle deviation between the actual heading of the vehicle body and the reference heading, and $e_v(k)$ represents the difference between the actual speed and the desired speed. Considering that the system dynamics changes with speed, curvature and road condition in a complex environment, the discrete error propagation relation is expressed as in this paper:

$$e_{k+1} = A_k e_k + B_k u_k + w_k \tag{2}$$

where, A_k and B_k are time-varying state matrices and control matrices, $u_k = [\delta_k, a_k]^T$correspond to steering control quantity and acceleration control quantity respectively, and w_k represents the comprehensive disturbance term composed of model mismatch, sensor noise and external disturbance. The significance of this expression is not to establish a completely accurate physical model, but to provide a unified computational carrier for subsequent parameter tuning and error prediction.

In the design of control law, this paper does not use a single feedback correction, but constructs a compound control form of feedforward and feedback parallel. The feedforward part directly uses the reference trajectory curvature and the desired velocity to generate the reference control signal, so that the system has a certain pretuning ability before entering the large curvature section. The feedback part modifies the reference input according to the current error state, which is used to suppress the execution deviation and the burst disturbance. Its control law can be written as:

$$u_k = u_k^{ff} - K_k e_k + \hat{d}_k \tag{3}$$

Here, u_k^{ff} is the feedforward quantity calculated from the reference trajectory, K_k is the time-varying feedback gain matrix, and $\hat{d}_k$ is the disturbance compensation term. Different from the traditional fixed gain control, K_k is not a constant, but is determined by the off-line optimization initial value and the on-line adjustment increment. The reason for this treatment is that the importance of different error components is not constant in complex environments. In the stage of narrow channel low-speed obstacle avoidance, lateral deviation suppression is more critical than velocity recovery. In the acceleration phase of open area, the speed error and heading comfort become the dominant factors. If the same set of gains is used in all scenarios, it is often difficult to balance stability and flexibility.

In order to obtain better initial control parameters, the constrained particle swarm optimization method is introduced to optimize the reference gain in the offline stage. The optimization objects include the lateral error feedback gain k_y, the heading error feedback gain k_ψ, the velocity error feedback gain k_v, and the control increment penalty weight. Considering that trajectory tracking should not only pursue the minimum error, but also avoid sudden turning and sharp acceleration fluctuations, this paper constructs the following performance indicators:

$$J = \sum_{k=1}^{N} \left(q_1 e_y^2(k) + q_2 e_\psi^2(k) + q_3 e_v^2(k) + r_1 \Delta\delta^2(k) + r_2 \Delta a^2(k) \right) \tag{4}$$

Among them, q_1, q_2, q_3 are used to measure the importance of the three types of errors, and r_1, r_2 are used to constrain the change rate of the control quantity. The objective function makes the optimization process pay attention to both tracking accuracy and control ride comfort, avoiding the degradation of one type of error only by reducing the other type of performance. The particle update process follows the following iterative relationship:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g^t - x_i^t), x_i^{t+1} = x_i^t + v_i^{t+1} \tag{5}$$

where, x_i^t represents the parameter position of the ith particle at the TTH iteration, v_i^t represents the particle velocity, p_i^t and g^t correspond to the individual and global optimal positions, ω is the inertia weight, and c_1 and c_2 are the learning factors. In this paper, the number of particles is set to 30, the maximum number of iterations is set to 120, the initial value of ω is set to 0.72, and both c_1 and c_2 are set to 1.6. The parameter search range is limited to $k_y \in [0.2,3.5]$, $k_\psi \in [0.5,4.5]$, $k_v \in [0.1,2.5]$, so as to ensure that the parameters obtained have sufficient adjustment amplitude, and do not lead to control overshoot under local high noise conditions. Figure 2 shows the overall process of offline optimization and online update.

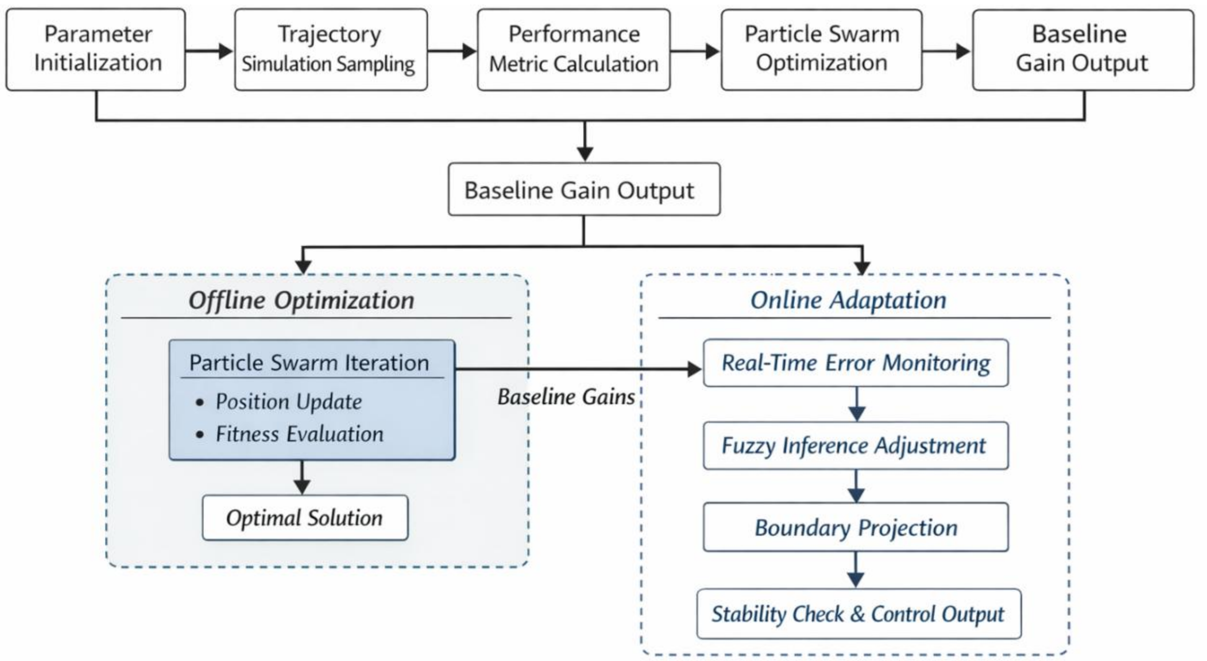

Figure 2: Process of offline optimization and online update of control parameters

Relying only on initial values obtained from offline optimization is still not sufficient to cover real-time changes in complex environments. To this end, this paper further introduces a fuzzy adaptive gain scheduler in the execution layer, which takes the normalized tracking error en and the error change rate ecn as inputs, and generates the gain correction factor online according to the current deviation intensity and the deviation evolution trend. Instead of directly reconstructing all the control parameters, a small range of constrained continuous correction is carried out near the offline optimal values, so as to give consideration to both adaptive ability and numerical stability. Table 2 shows the fuzzy rules for the gain correction factor. It can be seen that when the error and the error change rate are in a large positive interval at the same time, the output tends to be strongly positive regulation to enhance the correction strength. When the error is already close to zero and the change rate is small, the output falls back to near zero, avoiding the repeated swing of the controller due to excessive sensitivity.

Table 2: Fuzzy rule table of gain correction factor

$ec_n \backslash e_n$	NB	NM	ZO	PM	PB

NB	PB	PB	PM	PM	ZO
NM	PB	PM	PM	ZO	ZO
ZO	PM	PM	ZO	NM	NM
PM	ZO	ZO	NM	NM	NB
PB	ZO	NM	NM	NB	NB

Based on this, the online update of the feedback gain matrix is written as:

$$K_{k+1} = \Pi_{\Omega}(K_k + \Lambda\,\Delta\lambda_k) \tag{6}$$

Here, $\Delta\lambda_k$ is the gain correction of the fuzzy inference output, Λ is the diagonal matrix that scales different error channels, and $\Pi_{\Omega}(\cdot)$ represents the boundary projection operator used to restrict the updated parameters to the allowable interval Ω. There are two considerations in using the projection mechanism. First, the error peak may be caused by instantaneous occlusion or positioning drift in complex environments. Secondly, online control programs are usually deployed on embedded platforms, and controlling the numerical range is beneficial to maintain the execution efficiency and thread safety of the algorithm.

In order to further weaken the influence of external disturbances on the control results, this paper adds a low-order disturbance estimation module to the control loop, and updates the unknown disturbance recursively according to the error residual of the latest period. When the local path is replanned due to temporary obstacles, there is often a short-term discontinuity between the old and new reference trajectory. If the system completely relies on the error feedback to absorb this mutation, the steering amount will spike. After introducing the disturbance estimation, a part of the sudden changes can be interpreted as compensable terms, and the control pressure can be released in advance by feedforward correction. At the same time, in order to verify the convergence trend of the parameter update, the Lyapunov candidate function is constructed:

$$V_k = \frac{1}{2}e_k^T P e_k + \frac{1}{2}\tilde{K}_k^T \Gamma^{-1} \tilde{K}_k \tag{7}$$

Here, $\tilde{K}_k$ represents the estimated deviation between the current parameter and the ideal parameter, and both P and Γ are positive definite matrices. Under the condition of projection constraint and small step update, the system can maintain the non-increasing trend of V_k along the time direction, so as to ensure that the trajectory error will not unboundedly enlarge in successive iterations. This treatment does not pursue strict analytical optimality in form, but puts more emphasis on stable availability in real-time control of complex environments.

Taken together, the adaptive control structure for trajectory tracking designed in this section has three characteristics. First, the controller no longer regards the path planning result as a static input, but incorporates the reference curvature, velocity expectation and local trajectory switching information into the control calculation link, which strengthens the connection between the planning layer and the execution layer. Secondly, a two-stage strategy of "offline optimization to give the benchmark and online adjustment to complete the correction" is adopted, so that the parameters not only have the optimal initial values obtained by global search, but also have the dynamic adaptation ability for scene changes. Thirdly, through the combination of incremental constraint, disturbance compensation and stability judgment, the common problems of overshoot, oscillation and command mutation in complex environment are reduced, which lays a foundation for the integration of subsequent path planning model and controller.

3.2 Path planning model construction and integration of planning and control in complex environment

The trajectory tracking adaptive controller, constructed in the previous section, solves the problem of stable execution under the condition that the reference trajectory is known, but autonomous motion in complex environments does not stop there. In order to maintain continuous operation in scenes with dense obstacles, narrow channels and frequent local dynamic disturbances, the system must also have the overall ability to cooperate with path generation, local correction and control. In other words, just because the controller can "follow" doesn't mean that the system is "going right." When the environment topology changes with time, or the global path has unexecutable segments in the local area, simply relying on control gain adjustment cannot fundamentally eliminate the structural defects of the path layer. Based on this understanding, based on the adaptive control framework described in 3.1, this paper further constructs a path planning model for complex environments, and incorporates path search, local re-planning and tracking control into the same closed-loop system to realize the collaborative coupling of planning layer and execution layer.

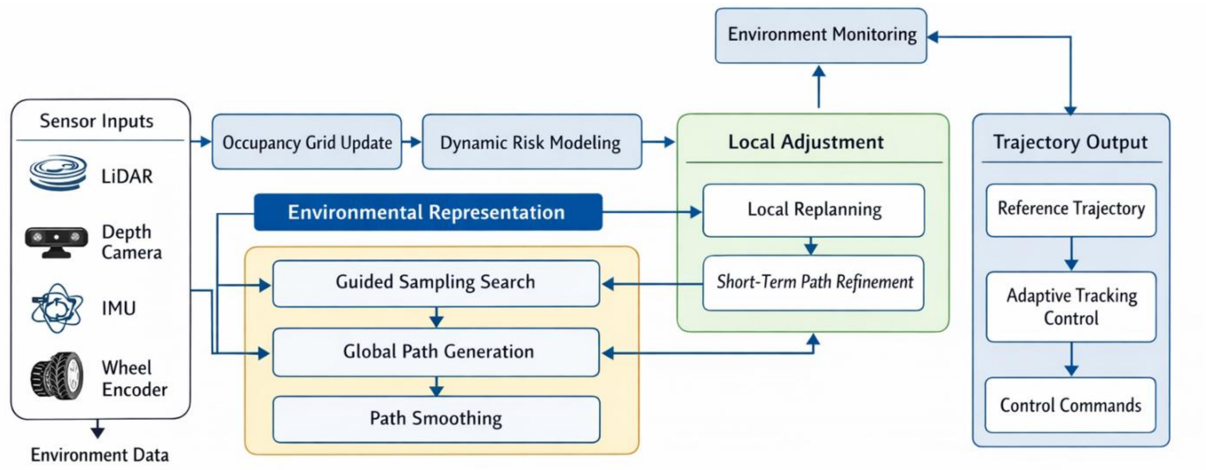

Figure 3: Structure diagram of path planning model in complex environment

As shown in Figure 3, the path planning model constructed in this paper adopts a four-level structure of "environment expression-global planning-local correction-trajectory output". At the bottom of the system, the environment and pose information are obtained by lidar, depth camera, inertial measurement unit and wheel speed encoder, and the real-time environment map is generated by raster mapping and local occupancy update. In the middle layer, the global search method of guided sampling is used to generate the main path from the starting point to the target point. When dynamic obstacle intrusion, local channel contraction or original path curvature does not meet the execution constraints are detected, the local planning module is activated to perform short-term reconstruction of the affected road section. The upper layer then converts the modified path sequence into a reference trajectory with velocity annotation and sends it to the adaptive controller designed in Section 3.1 for closed-loop tracking. Compared with the traditional serial structure, the path planning is no longer regarded as a one-time calculation result, but the path is regarded as a sustainable updating intermediate state, so that the system has the ability of online adaptation for complex environments.

In order to make the planning results closer to the real execution conditions, this paper does not only use the binary occupancy grid to describe the environment, but introduces the dynamic risk and the scene traffic cost in addition to the free space judgment, and constructs a fusion environmental cost field. For any position p=(x,y), the combined risk is defined as:

$$\Phi_t(p) = \lambda_1 O_t(p) + \lambda_2 D_t(p) + \lambda_3 S(p) \tag{8}$$

where $O_t(p)$ represents the static occupied state of position p at time t, and takes a larger value if the position is occupied by obstacles. $D_t(p)$ represents the dynamic risk probability predicted by the movement disorder. $S(p)$ represents difficulty information such as terrain constraints, attenuation of channel width or proximity to boundaries. The $\lambda_1, \lambda_2, \lambda_3$ are the weight coefficients. The meaning of this formula is to distinguish "whether it can pass" from "whether the cost of passing is too high". In this way, the global planning no longer only pursues the shortest geometric distance, but actively avoids high-risk areas and narrow channel edges, and reduces the correction burden of subsequent control layers from the path source.

In the global planning module, this paper adopts the improved sampling search strategy under the guided corridor constraint. Although the traditional random sampling method has good scalability in high-dimensional space, it still often has problems such as too many invalid expansion nodes, slow convergence speed and redundant turning points in complex two-dimensional/quasi-two-dimensional environment. In order to improve the search efficiency, we generate an adaptive sampling distribution according to the current optimal path, target orientation and free space distribution, so that new nodes are more inclined to fall near the existing good path and its natural extension direction, rather than evenly scattered on the whole map. Its sampling probability model is denoted by:

$$p(x) = \eta\, \mathcal{N}(x; \mu_g, \Sigma_g) + (1-\eta)\, U(\Omega_f) \tag{9}$$

Here, $\mathcal{N}(x; \mu_g, \Sigma_g)$ represents the Gaussian distribution with the guided path node mean μ_g and covariance Σ_g as parameters, $U(\Omega_f)$ represents the uniform distribution on the free space Ω_f, and $\eta \in (0,1)$ is the guided sampling weight. This formula retains the necessary exploration of random search, and enhances the ability to focus on potential excellent regions through the guidance term. In the implementation of the program, KD-tree is used to accelerate the nearest neighbor search, and fast collision detection is performed on the newly expanded edges to reduce the time overhead of the node reconnection process.

Node expansion alone is not enough to generate execution-friendly primary paths, and the path evaluation function must also take curvature and safety margin into account. To this end, this paper defines the following global cost for the candidate path $= \{p_1, p_2, \cdots, p_N\}$:

$$J_g(\mathcal{P}) = \sum_{k=1}^{N-1} \left(\alpha \|p_{k+1} - p_k\| + \beta \Phi_t(p_k) + \gamma |\kappa_k| + \zeta \frac{1}{d_k + \varepsilon} \right) \tag{10}$$

Here, $\|p_{k+1} - p_k\|$ represents the Euclidean distance between adjacent waypoints, κ_k represents the discrete curvature near the KTH node, d_k represents the distance from this node to the nearest obstacle boundary, ε is a tiny constant to prevent the denominator from being zero, and $\alpha,\beta,\gamma,\zeta$ are weight parameters. Different from the traditional way of selecting the optimal path only according to its length, the objective function will suppress the excessively curved path segments and encourage the path to stay away from the obstacle boundaries while maintaining the accessibility. The main path generated in this way is geometrically more suitable for subsequent trajectory smoothing and control execution. The discrete node sequence obtained by searching is further continuized by cubic B-spline, and the reference centerline satisfying the curvature continuity constraint is obtained.

However, the global path, even if smoothed, cannot fully cope with the instantaneous changes in complex environments. Especially when dynamic obstacles are traversed, local view is occluded, or narrow channels are temporarily blocked, only relying on global recomputation will bring large time overhead and may cause short-term stagnation of the execution link. Therefore, this paper superimposed a local velocity space planning module on the main path, sampled the linear velocity v and angular velocity ω at the current time, generated multiple short-term prediction trajectories, and selected the optimal obstacle avoidance action from them. The local trajectory evaluation function is written as:

$$G(v,\omega) = aH(v,\omega) + bV(v,\omega) + cD(v,\omega) + dT(v,\omega) \tag{11}$$

where H(v,ω) represents the heading consistency index towards the target direction, V(v,ω) represents the velocity gain term, D(v,ω) represents the minimum safety distance evaluation with obstacles, T(v,ω) represents the tracking consistency term between the local predicted trajectory and the global reference path, and a,b,c,d are the corresponding weights. In particular, T(v,ω) is introduced to prevent the local obstacle avoidance action from going too far away from the main path. Without this item, although local planning may avoid obstacles in a short time, it is easy to produce the problem of too large detour and difficulty in returning to the main path. With the addition of consistency constraints, local actions will be more inclined to select those velocity combinations that can safely avoid obstacles and facilitate reintegration into the global path.

In order to make path planning and control execution truly form linkage, this paper does not regard the local planning results as the final output directly, but further constructs the planning-control joint optimization index. Let the control sequence in the prediction time domain be U, then its integrated cost is denoted by:

$$\mathcal{U}^* = \arg\min_{\mathcal{U}} \sum_{\tau=1}^{N_p} \left(J_g^{\tau} + \xi \|e_{\tau}\|_Q^2 + \nu \|\Delta u_{\tau}\|_R^2 \right) \tag{12}$$

Here, J_g^{τ} represents the path cost on the prediction step τ, e_{τ} is the tracking error vector at the corresponding time, Δu_{τ} represents the control increment, Q and R are the error weight matrix and the control smoothing weight matrix, respectively, ξ and ν are the coordination coefficients. This formula puts the traversability of the path layer, the error convergence of the control layer and the smoothness of the execution layer into the same target, avoids the planning module giving a trajectory that is "theoretically feasible but difficult to track", and avoids the control module from excessive tracking of local geometric optima, which causes high-frequency action jitter. From the perspective of implementation, this joint cost does not require a heavy optimization solution at each time, but starts the local update when the trigger condition is satisfied, and takes the optimal solution at the previous time as the initial value of the current iteration, so as to improve the efficiency of online calculation.

Table 3 shows the key parameter Settings of the proposed integrated planning and control model. The relevant parameters are not arbitrarily selected, but determined jointly according to the size of the simulation environment, the scale of the robot, the control refresh frequency and the local obstacle avoidance requirements. In particular, there is a significant coupling relationship between the safe expansion radius and the local prediction time domain. If the expansion radius is too large, the narrow passage will be judged as impassable prematurely. If the prediction time domain is too short, it is difficult for the system to give stable obstacle avoidance action in time in the high-speed motion stage. In this paper, the parameters are balanced through multiple sets of experiments.

Table 3: Key parameter Settings of the integrated model of path planning and control

Parameter Item	Setting Value	Description
Grid Map Resolution	0.10 m	Environment discretization precision
Obstacle Safety Inflation Radius	0.35 m	Accounts for vehicle shape and safety margin
Maximum Number of Global Sampling Nodes	2500	Upper limit for guided sampling search
Goal Bias Probability	0.18	Increases the expansion frequency toward the target region
Neighborhood Rewiring Radius	1.20 m	Optimization range for the global path
Local Planning Prediction Horizon	2.0 s	Prediction length for velocity-space sampling
Number of Linear Velocity Samples	7	Number of candidate local trajectories
Number of Angular Velocity Samples	9	Number of discretized steering actions
Minimum Safety Distance Threshold	0.45 m	Key criterion for triggering local obstacle avoidance
Planning–Control Refresh Period	50 ms	Update frequency of closed-loop coordination

As shown in Figure 4, the integration implementation process of this paper is not a one-way execution chain, but a continuous rolling closed-loop update process. When the system is running, the sensor thread continuously updates the pose estimation and local environment slice, and the map thread synchronously maintains the occupancy grid and the risk cost field. When the main path is available and there is no local high-risk disturbance, the system advances according to the global reference trajectory, and the steering and driving commands are output by the adaptive controller designed in Section 3.1. When dynamic obstacles are detected in front of the path, the passable gap is narrowed, or the predicted tracking error exceeds the threshold, the local planner is triggered to perform short-term reconstruction near the main path, and the new trajectory segment is passed to the controller to complete the smooth connection. If the reachability cannot be maintained after multiple local corrections, the global planning module is restarted and the main path skeleton is updated. The result of this design is that the system can maintain the global goal orientation, but also has enough local mobility, and will not gradually deviate from the overall task route by a single local obstacle avoidance.

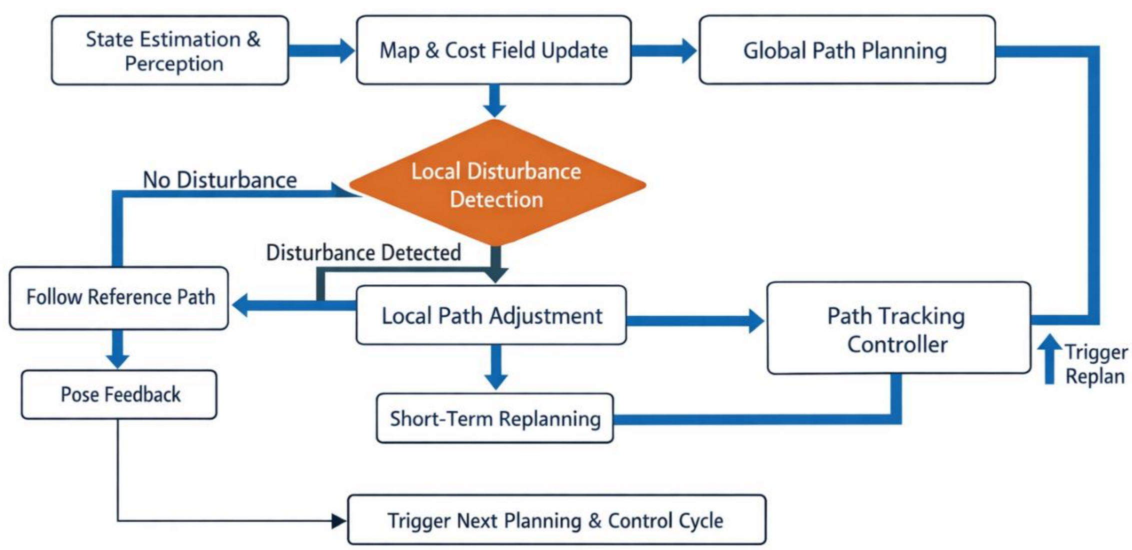

Figure 4: Flowchart of closed-loop realization of integrated planning and control

In general, the path planning model for complex environments constructed in this section has three characteristics. On the one hand, the environment representation is no longer limited to static obstacle occupation, but incorporates dynamic risk and passage difficulty into the cost modeling, which makes the path search closer to the real scene. On the other hand, through the combination of guided sampling, path cost reconstruction and local velocity space correction, the system achieves a good balance between global efficiency and local flexibility. At the same time, the path planning result is not separated from the controller, but forms a closed-loop coupling with the adaptive control structure of Section 3.1 through the joint cost function and the state feedback mechanism. The integrated implementation of planning and control provides a method basis for the experimental analysis of path quality, tracking accuracy and dynamic adaptability in the subsequent performance verification part.

4 Performance verification of adaptive control methods for path planning and trajectory tracking

4.1 Performance evaluation of path planning in complex environments

In order to verify the effectiveness of the path planning model in complex environments proposed in this paper, this section carries out comparative experiments under the unified simulation platform, and makes comprehensive evaluation from five dimensions of planning time, path length, safe spacing, curvature smoothness and task success rate. Considering that the method in this paper is not a single search algorithm, but jointly models the global planning of guided sampling, local dynamic correction and control enforceability constraints, the experimental focus is not only on "whether the path can be found", but also on whether the path is smooth enough to facilitate the stable execution of subsequent trajectory tracking control. RRT*, Hybrid A* and traditional DWA are selected as the comparison methods. RRT* focuses on the global sampling search ability, Hybrid A* takes into account the vehicle kinematics constraints, and DWA emphasizes the real-time performance of local obstacle avoidance. The proposed method is denoted as IPC-APTC (Integrated Planning and Control with Adaptive Path-Tracking Control).

The experimental scene was set as a 60 m×60 m two-dimensional raster map with a resolution of 0.1 m. The density of known static obstacles was set to 15%-35% and randomly generated in different trials. The number of unknown dynamic obstacles was set as 3, and the motion speed range was 0.2-0.8m /s. The initial position of the robot is (2, 2) and the target point is (55, 54). The maximum linear velocity and angular velocity of the system are 1.2 m/s and 1.5 rad/s, respectively. In order to reduce the influence of chance factors, each group of algorithms was run 30 times independently, and the results were counted in the form of mean. The relevant experimental parameters are shown in Table 4.

Table 4: Parameter Settings of path planning experiment in complex environment

Parameter Item	Setting Value
Map Size	60 m × 60 m
Grid Resolution	0.1 m
Static Obstacle Density	15%–35%
Number of Dynamic Obstacles	3
Dynamic Obstacle Speed	0.2–0.8 m/s

Maximum Linear Velocity of the Robot	1.2 m/s
Maximum Angular Velocity of the Robot	1.5 rad/s
Local Planning Refresh Period	0.05 s
Number of Independent Trials per Group	30
Evaluation Metrics	Planning time, path length, safety clearance, curvature variation rate, success rate

In A medium complexity scenario (obstacle density 25%), the average planning time of the proposed method is 0.43 s, which is significantly shorter than 0.86 s of RRT* and 0.61 s of Hybrid A*, and is also lower than 0.57 s of traditional DWA under global-local segmentation processing. In terms of path quality, the average path length generated by the proposed method is 78.4 m, which is 9.7% shorter than RRT* and 6.1% shorter than Hybrid A*. More importantly, the minimum safe distance of the proposed method reaches 0.71 m, and the average rate of path curvature change is 0.184, which indicates that the method still retains good turning continuity when approaching the obstacle boundary, and there is no obvious sharp corner line. The corresponding task success rate reaches 96.7%, which is higher than 88.9% for RRT, 92.3% for Hybrid A and 85.6% for DWA. This indicates that the proposed model can not only give a feasible path faster in complex environments, but also the output trajectory is more suitable as a reference input for the subsequent tracking controller.

Figure 5 further shows the variation of the task success rate of each algorithm under different obstacle densities. It can be seen that when the obstacle density increases from 15% to 35%, the success rate of each method decreases, but the decrease is not consistent. The success rate of the proposed method under the obstacle density of 15%, 20%, 25%, 30% and 35% is 98.6%, 97.9%, 96.7%, 94.8% and 91.5%, respectively, and the overall change is relatively gentle. RRT* decreased from 95.1% to 82.4%, Hybrid A* decreased from 96.0% to 86.1%, and DWA decreased from 91.8% to 76.9%. This difference shows that as the traversable space of the environment is continuously compressed, the path planning methods that only rely on random expansion or simple local response are more susceptible to narrow channels and dynamic interference. However, the proposed method can still maintain a high path reachability with the help of risk cost field constraints, guided sampling mechanism and local re-planning.

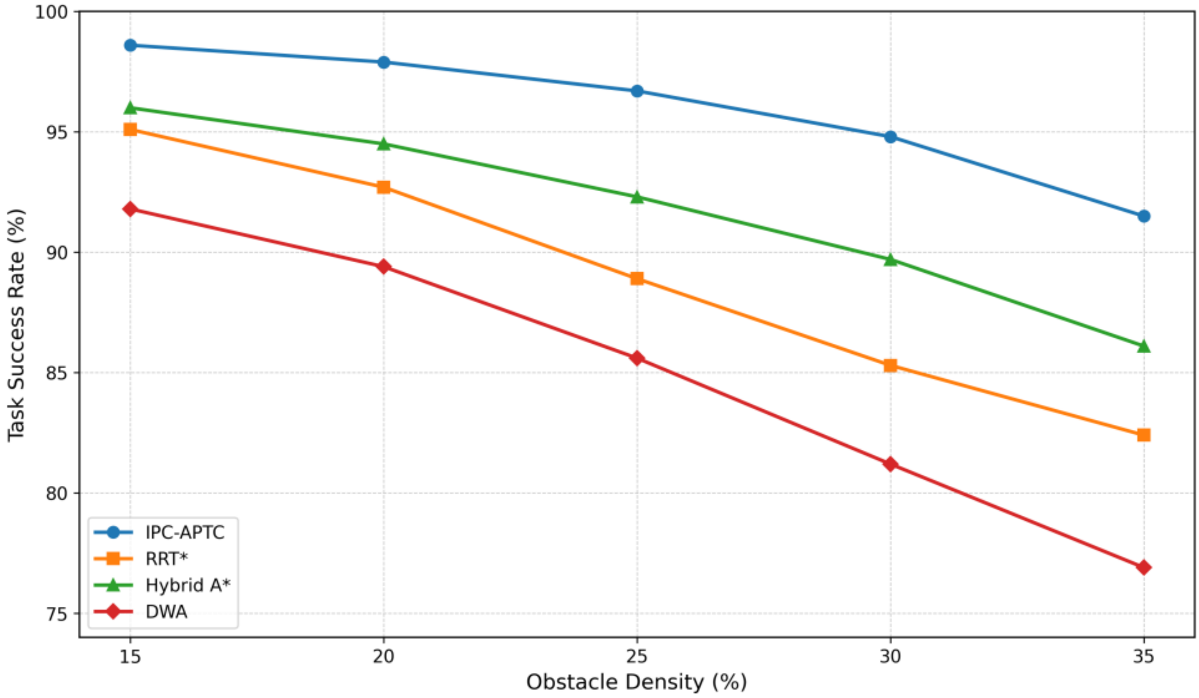

Figure 5: Comparison of path planning success rates of each algorithm under different obstacle densities

Combining Table 4 and Figure 5, it can be found that the advantages of the proposed method are not only reflected in a single indicator. On the one hand, it compressed the invalid search range by guided sampling and reduced the global planning time in complex environments. On the other hand, with the help of local velocity space modification and control executability constraints, the path mutation and near-obstacle jitter are reduced, so that the planning results achieve a better balance between safety and smoothness. For the path-following system, this path quality improvement has direct significance, because the more continuous curvature distribution and more reasonable safety margin can significantly reduce the instantaneous regulation pressure of the control layer. It can be seen that the complex environment path planning model proposed in this paper is superior to the comparison methods in terms of comprehensive performance, and can provide a more stable and higher quality reference trajectory for subsequent trajectory tracking adaptive control.

4.2 Practical application analysis of adaptive control method for trajectory tracking

After the path planning performance verification at the simulation level, this paper further carries out field application experiments to verify the stability and deployability of the proposed trajectory tracking adaptive control method under real execution conditions. Compared with the ideal simulation environment, the sensor sampling in the actual scene has time jitter, the wheel speed feedback is affected by the difference of ground adhesion, and the reference trajectory is also switched for a short time due to partial occlusion and temporary obstacle entry. Therefore, this section no longer simply investigates whether the path is reachable, but focuses on the evaluation of motion ride comfort, trajectory tracking accuracy, error recovery speed, and continuous operation ability under complex disturbance conditions. The control program is deployed by ROS 2 and Python/C++. The planning thread and control thread run at the frequency of 20 Hz and 50 Hz respectively. The experimental site is a 14 m×10 m indoor test area, and a 0.2 m calibration grid is laid on the ground to facilitate subsequent pose verification. The relevant platform parameters are shown in Table 5.

Table 5: Field experimental platform and operating environment parameters

Configuration Item	Parameter Setting
Mobile Platform Dimensions	0.82 m × 0.58 m × 0.46 m
Vehicle Body Mass	38 kg
Maximum Payload	80 kg
Maximum Linear Velocity	1.2 m/s
Maximum Angular Velocity	1.6 rad/s
LiDAR Range	20 m
IMU Sampling Frequency	100 Hz
Control Host	Intel Core i7-12700H, 32 GB RAM
Operating System	Ubuntu 22.04
Software Environment	ROS 2 Humble, Python 3.11, C++17
Control Refresh Frequency	50 Hz
Planning Refresh Frequency	20 Hz

Three kinds of typical disturbance scenarios are set up in the experiment: the first is the conventional detour under the condition of known static obstacles; Second, unknown static obstacles temporarily intrude into the main path; Third, dynamic obstacles cross laterally to form a short - time blocking. In order to reflect the advantages of the method, the proposed method is compared with three combination strategies: RRT*+PID, Hybrid A*+MPC and DWA+ fixed gain control. The same starting point, end point and speed upper limit were used for each method, and each set of scenarios was run 20 times independently. The evaluation metrics include task success rate, average tracking error, maximum lateral error, average replanning recovery time, and average running time. It should be noted that the average tracking error is defined as the average Euclidean distance between the actual position of the robot and the closest point of the reference trajectory in the execution phase of the whole trajectory, and the maximum lateral error is taken as the peak value of the vertical deviation during a single run.

From the field test results, the proposed method shows more stable tracking quality in all three types of scenes. Under the condition of known static obstacles, the system can smoothly pass along the main path without frequent correction of control commands, and the changing curve of steering Angle is more continuous. When the unknown static obstacle suddenly entered the reference path, the local planning module completed the path correction within 0.41 s, and the adaptive controller adjusted the lateral error gain synchronously, so that the vehicle returned to the reference trajectory within 1.3 s without obvious secondary swing. In the dynamic obstacle traversing scenario, the proposed method does not adopt a conservative wide range detour, but combines obstacle velocity prediction and local traversable gap calculation to give a short obstacle trajectory on the premise of ensuring the safety distance, and then suppresses the control impact caused by trajectory switching by online parameter adjustment. Figure 6 shows the trajectory tracking accuracy variation of each method in dynamic obstacle scenarios. It can be seen that the tracking accuracy of the proposed method is improved to 98.7% within 6 s and stabilized at 99.1% after 12 s. In contrast, RRT*+PID shows a significant fluctuation in the middle, with a peak accuracy of only 95.6%, and DWA+ fixed gain control produces a large error spike at the obstacle cut, and the recovery process is also longer.

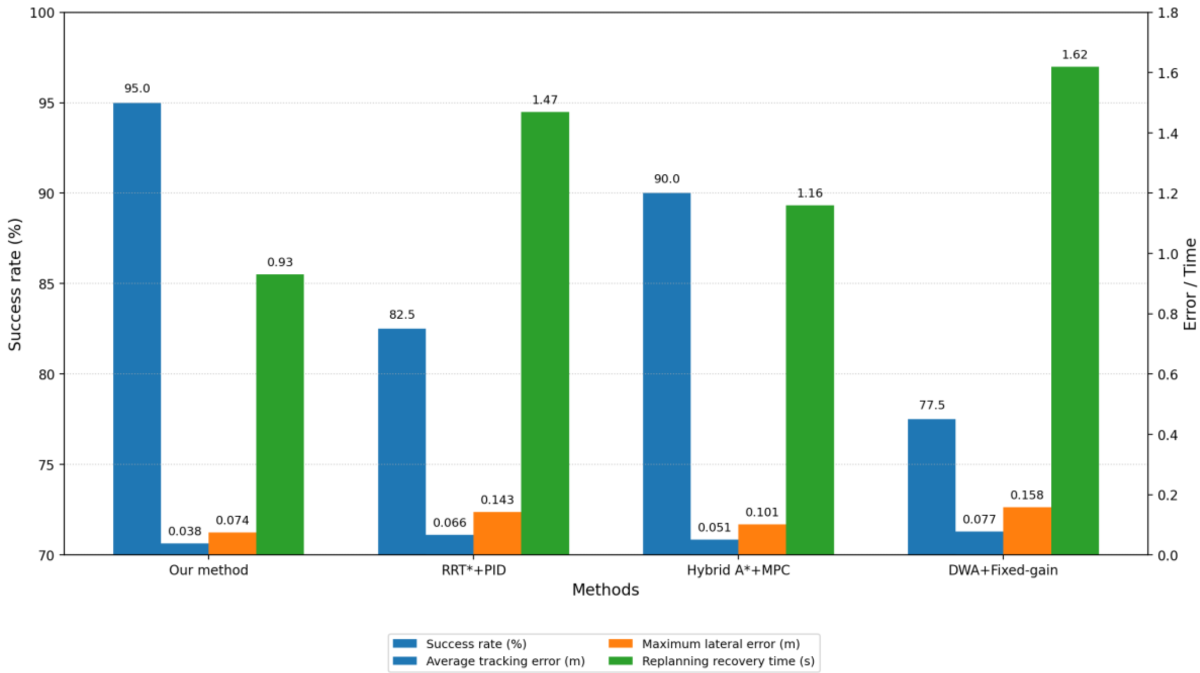

Figure 6: Comparison of trajectory tracking accuracy and lateral error variation of each method in dynamic obstacle scenario

To quantify the differences more intuitively, Table 6 lists the comprehensive performance of different methods under three categories of field scenarios. It can be found that the task success rate of the proposed method is higher than 95% in all scenarios, and the scene with known static obstacles reaches 100.0%. In the dynamic obstacle crossing scenario, the average tracking error of the proposed method is 0.038 m, which is 42.4% lower than that of RRT*+PID and 50.6% lower than that of DWA+ fixed gain control. The average replanning recovery time is only 0.93 s, which indicates that the system can quickly complete the trajectory reconstruction and control convergence after the local path is interrupted. At the same time, the average running time of the proposed method is 31.8 s, which is shorter than the other three methods, indicating that the proposed integrated framework does not significantly increase the execution burden due to the introduction of online adjustment mechanism, but shortens the task completion time by reducing invalid detours and large oscillations.

Table 6: Performance comparison of different methods under field application scenarios

Scenario	Method	Success Rate (%)	Average Tracking Error (m)	Maximum Lateral Error (m)	Replanning Recovery Time (s)	Average Runtime (s)
Known static obstacles	Proposed method	100.0	0.026	0.061	0.00	24.7
	RRT* + PID	95.0	0.054	0.118	0.00	28.9
	Hybrid A* + MPC	100.0	0.039	0.086	0.00	26.8
	DWA + fixed-gain control	90.0	0.063	0.131	0.00	29.6
Unknown static obstacle intrusion	Proposed method	97.5	0.033	0.072	0.41	27.9
	RRT* + PID	90.0	0.061	0.126	0.88	33.5
	Hybrid A* + MPC	95.0	0.046	0.094	0.57	30.8
	DWA + fixed-gain control	87.5	0.074	0.149	0.96	34.2
Dynamic obstacle crossing	Proposed method	95.0	0.038	0.074	0.93	31.8
	RRT* + PID	82.5	0.066	0.143	1.47	39.6
	Hybrid A* + MPC	90.0	0.051	0.101	1.16	35.2
	DWA + fixed-gain control	77.5	0.077	0.158	1.62	40.4

In order to further investigate the necessity of each sub-module in the integrated design, an additional comparative analysis is carried out. After the online gain adjustment is turned off, the average tracking error in dynamic obstacle scene is increased from 0.038 m to 0.052 m, and the maximum lateral error is increased to 0.097 m. After removing the local risk cost term, the system can still complete path generation, but the success rate in dynamic obstacle scenarios drops to 89.2%, and it is easier to operate close to the obstacle boundary after replanning. This shows that the risk constraint of the path planning layer and the adaptive adjustment of the control layer are not independent additional functions, but key components to jointly support the stable operation of the system in a complex environment.

5 Discussion

Based on the above experimental results, it can be found that the performance improvement of the proposed method does not come from the local optimization of a single planner or a single controller, but from the tighter computational coupling between the path planning layer and the trajectory tracking layer. In the path planning experiment of complex environment, the average planning time of the proposed method is 0.43 s when the obstacle density is 25%, which is shorter than 0.86 s of RRT and 0.61 s of Hybrid A. The average path length is 78.4m, and the task success rate reaches 96.7%. When the obstacle density is further increased to 35%, the success rate of the proposed method still remains at 91.5%, while RRT* and DWA decrease to 82.4% and 76.9%, respectively. This indicates that the combination of guided sampling and risk cost field not only compreses the invalid search range, but also weakens the destruction of path reachability by complex obstacle distribution. From the practical application analysis, the average tracking error of the proposed method is 0.038 m, the maximum lateral error is 0.074 m, and the replanning recovery time is 0.93 s in the dynamic obstacle crossing scene, which is better than that of RRT*+PID, Hybrid A*+MPC and DWA+ fixed gain control. Especially after the unknown obstacle intruded into the main path, the system could still complete the local trajectory correction and recover the stable tracking in a relatively short time, which indicates that the online gain adjustment mechanism constructed in the previous section has a direct effect on suppressing the trajectory switching impact. In other words, the proposed method not only improves the ability to "find a path", but more importantly improves the ability to "run stably along an executable path", which is particularly critical for mobile robots in complex environments.

However, the experimental results also show that there is still room for further improvement. Taking the dynamic obstacle traversing scene as an example, although the success rate reaches 95.0%, it is still lower than 100.0% in the static scene, indicating that the current prediction model is not sufficient to describe the high dynamic disturbances when the obstacle speed, direction and local channel constraints change simultaneously. At the same time, the scale of the experimental site in this paper is mainly concentrated in small and medium-sized indoor environments, and the map scale and obstacle complexity are still relatively controllable. If it is extended to a wider range of storage, inspection or mixed traffic scenarios, the cost field update frequency, local replanning overhead and multi-sensor synchronization pressure may increase significantly. The follow-up research can focus on the prediction of high-dynamic obstacle intention, the compressed expression of hierarchical map, and the lightweight online optimization to further enhance the real-time performance and generalization ability of the method in large-scale complex environments.

6 Conclusions

Aiming at the problems of insufficient path planning efficiency, disconnection between local obstacle avoidance and trajectory execution, and easy amplification of tracking error in dynamic disturbances in complex environments, this paper constructs an integrated path planning and trajectory tracking adaptive control method. In the method design, the global planning of guided sampling, local velocity space modification, risk cost field modeling and online gain adjustment are integrated into a unified closed-loop framework, so that path generation, trajectory reconstruction and control execution can be continuously coordinated. The experimental results show that the average planning time is 0.43 s, the average path length is 78.4 m, and the task success rate reaches 96.7% when the obstacle density is 25%. At 35% obstacle density, the success rate remains 91.5%. In the field application test, the average tracking error of the system in the dynamic obstacle crossing scene is 0.038 m, the maximum lateral error is 0.074 m, and the replanning recovery time is 0.93 s. The overall performance of the system is better than the comparison methods. The results show that the proposed method can balance the path accessibility, control smoothness and dynamic adaptability, which is suitable for the autonomous operation of mobile robots in complex environments. In the future, it can continue to be optimized around real-time calculation compression in large-scale scenarios, high dynamic obstacle intention prediction and multi-platform migration ability.

Funding

This work was supported by the Young and Middle-aged Research Project of Fujian Province (JAT251353) and the Research Institute of Meizhouwan Vocational and Technical College (PT2023009).

References

[1] AbuJabal N, Baziyad M, Fareh R, et al. A comprehensive study of recent path-planning techniques in dynamic environments for autonomous robots[J]. Sensors, 2024, 24(24): 8089.

[2] Katona K, Neamah H A, Korondi P. Obstacle avoidance and path planning methods for autonomous navigation of mobile robot[J]. Sensors, 2024, 24(11): 3573.

[3] Liu J, Yap H J, Khairuddin A S M. Review on motion planning of robotic manipulator in dynamic environments[J]. Journal of Sensors, 2024, 2024(1): 5969512.

[4] Liu L, Wang X, Yang X, et al. Path planning techniques for mobile robots: Review and prospect[J]. Expert Systems with Applications, 2023, 227: 120254.

[5] Yang H, Teng X. Mobile robot path planning based on enhanced dynamic window approach and improved A∗ algorithm[J]. Journal of robotics, 2022, 2022(1): 2183229.

[6] Hou J, Jiang W, Luo Z, et al. Dynamic path planning for mobile robots by integrating improved sparrow search algorithm and dynamic window approach[C]//Actuators. MDPI, 2024, 13(1): 24.

[7] Park M, Lee S Y, Hong J S, et al. Deep deterministic policy gradient-based autonomous driving for mobile robots in sparse reward environments[J]. Sensors, 2022, 22(24): 9574.

[8] Tao B, Kim J H. Deep reinforcement learning-based local path planning in dynamic environments for mobile robot[J]. Journal of King Saud University-Computer and Information Sciences, 2024, 36(10): 102254.

[9] Zhang Y, Chen P. Path planning of a mobile robot for a dynamic indoor environment based on an sac-lstm algorithm[J]. Sensors, 2023, 23(24): 9802.

[10] Sun Z, Wang R, Meng X, et al. A novel path tracking system for autonomous vehicle based on model predictive control[J]. Journal of Mechanical Science and Technology, 2024, 38(1): 365-378.

[11] Liu X, Ou J, Yan D, et al. Path tracking control of automated vehicles based on adaptive MPC in variable scenarios[J]. IET Intelligent Transport Systems, 2024, 18(6): 1031-1044.

[12] Wang J, Yu Y, Song Y, et al. A novel adaptive trajectory tracking control for autonomous vehicles based on state expansion[J]. Journal of Mechanical Science and Technology, 2024, 38(6): 3143-3154.

[13] Liu D, Tang M, Fu J. Robust adaptive trajectory tracking for wheeled mobile robots based

on Gaussian process regression[J]. Systems & Control Letters, 2022, 163: 105210.

[14] Marashian A, Razminia A. Mobile robot's path-planning and path-tracking in static and dynamic environments: Dynamic programming approach[J]. Robotics and Autonomous Systems, 2024, 172: 104592.

[15] Feng J, Yang Y, Zhang H, et al. Path planning and trajectory tracking for autonomous obstacle avoidance in automated guided vehicles at automated terminals[J]. Axioms, 2023, 13(1): 27.

[16] Zhang Y, Gao F, Zhao F. Research on path planning and tracking control of autonomous vehicles based on improved RRT* and PSO-LQR[J]. Processes, 2023, 11(6): 1841.

[17] Zhang Y, Liu K, Gao F, et al. Research on path planning and path tracking control of autonomous vehicles based on improved APF and SMC[J]. Sensors, 2023, 23(18): 7918.

[18] Zhang L, Ge W, Pan F, et al. Adaptive path planning and tracking system based on model predictive control for autonomous vehicle local obstacle avoidance[J]. Asian Journal of Control, 2024, 26(3): 1506-1516.

[19] Li X, Wang L, An Y, et al. Dynamic path planning of mobile robots using adaptive dynamic programming[J]. Expert Systems with Applications, 2024, 235: 121112.

[20] Bulut V. Path planning of mobile robots in dynamic environment based on analytic geometry and cubic Bézier curve with three shape parameters[J]. Expert Systems with Applications, 2023, 233: 120942.

[21] Kobayashi M, Zushi H, Nakamura T, et al. Local path planning: Dynamic window approach with Q-learning considering congestion environments for mobile robot[J]. IEEE Access, 2023, 11: 96733-96742.

[22] Yu J. Research on mobile robot path planning and tracking control[J]. International Journal of Computational Science and Engineering, 2023, 26(4): 349-360.