Robot Dynamic Environment Path Planning Based On Kalman-filter-based Prediction And Complexity-Adaptive Collaboration

Ruiyang Wang and Qi Yang*

School of Mechanical Engineering, Shenyang Ligong University, Shenyang, Liaoning, China

* *Corresponding author. E-mail: yangqi@sylu.edu.cn*

Received: Apr. 23, 2026; Accepted: May 16, 2026

To tackle the critical challenges confronted by mobile robots in dynamic environments, such as the demand for short replanning time, high computational complexity, and unreliable planning results arising from environmental uncertainty, the majority of existing methods merely implement local obstacle avoidance in the post-processing phase, which leads to limited improvements in global path optimization. Moreover, the fixed coordination mechanism between global and local planning is difficult to adapt to varying scenario pressures. To this end, this paper proposes a planning method with a predictive dynamic cost field and a complexity-adaptive coordination mechanism (PADP), which employs Kalman-filter-based prediction to construct a direction-aware dynamic cost field and integrates it into global path generation. An online complexity indicator composed of obstacle density and other metrics is designed to dynamically adjust planning weights. Meanwhile, a lightweight residual correction module is introduced to reduce multi-step prediction errors while retaining the core Kalman filtering framework. Experimental results based on MATLAB grid simulations demonstrate that PADP achieves an overall success rate of 80.56% and has the minimum average collision count among all comparison methods in 36 dynamic scenarios. The hybrid PADP-KFRes reduces the prediction RMSE from 1.053 to 0.457.

Keywords: Mobile robot; Kalman filter; Dynamic path planning; Adaptive coordination mechanism

http://dx.doi.org/10.6180/jase.202609_32.061

1. Introduction

Path planning for mobile robots in dynamic environments requires a balanced trade-off among reachability, safety, and real-time performance. This problem arises extensively in scenarios such as warehouse logistics, indoor services, inspection, and collaborative manufacturing [1, 2]. With the navigation environment evolving from sparse to crowded, robot interaction becomes more frequent. This problem is no longer merely a shortest-path search on a static map. The planner must judge whether future obstacle motions affect route selection in the early path generation stage. It also needs to decide whether the coupling between global and local planning should change with scenario pressure [3, 4]. Therefore, the core issue lies not only in the success of local obstacle avoidance, but also in whether future risks can be incorporated into path generation sufficiently early and whether the coordination mechanism can respond in a timely manner. To address this problem, heuristic search, local trajectory optimization, and potential field methods remain the most common engineering-oriented technical routes [5, 6].

Some studies improve path smoothness and local feasibility by refining the combination of A* and DWA [7–10]; others rely on APF, TEB, or perception constraints to enhance near-field obstacle avoidance [11–13]; a few works attempt to unify global path design and local obstacle avoidance [14]. These methods are attractive due to

their clear structure and ease of online deployment. However, when obstacles intersect, interleave, or compete at close range, risk modeling is still primarily based on current positions or instantaneous geometric relationships. Under such conditions, global paths often fail to anticipate potential conflict zones in a timely manner [11, 12, 15–17]. In recent years, learning-based methods, motion prediction models, and navigation in crowded environments have expanded robots representation ability for dynamic scenarios [18, 19]. Reinforcement learning is often adopted for continuous decision-making in unknown environments [19, 20], while velocity obstacle prediction, motion prediction networks, and multimodal prediction are used to strengthen interactive modeling capabilities [21, 22]. Currently, most prediction modules focus solely on local obstacle avoidance and fail to introduce predictive information as explicit costs into global planning, thus resulting in limited improvement in global path optimization.

To address these issues, this paper integrates future risk awareness into global path planning and employs scenario complexity to adapt the coordination between global replanning and local evaluation. This paper presents three methodological innovations:

- **A predictive dynamic cost field is introduced to explicitly incorporate future obstacle motion into global path generation.**
- **A complexity-adaptive coordination mechanism is developed to dynamically balance global replanning and local trajectory evaluation.**
- **A residual Kalman prediction framework is proposed to improve short-horizon motion estimation while preserving model interpretability**

2. Method

2.1. Overall Framework

The overall framework of the proposed PADP method is illustrated in Fig. 1. The static map is first converted into a static risk base layer via clearance cost mapping, while dynamic observations are used to construct a predictive dynamic cost field through short-term Kalman-filter-based prediction. In the hybrid extension developed in this study, a lightweight residual corrector is embedded inside this prediction module, with the outer planning framework remaining unchanged. The static risk and dynamic risk together form a unified risk-aware cost map. Meanwhile, complexity evaluation and hysteresis-based mode switching jointly regulate global replanning and local trajectory evaluation, forming a closed-loop planning pipeline for dynamic environments.

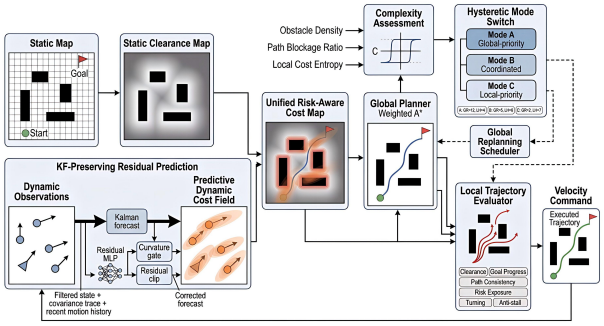

Fig. 1. Overall framework of the PADP method

2.2. System Modeling and Problem Formulation

Let the planar position, heading angle, linear velocity, and angular velocity of the robot at discrete time k be p_k, θ_k, v_k, and ω_k, respectively. Then its discrete kinematic model is given as follows:

$$p_{k+1} = p_k + v_k \Delta t \begin{bmatrix} \cos\theta_k \\ \sin\theta_k \end{bmatrix}, \quad \theta_{k+1} = \theta_k + \omega_k \Delta t \tag{1}$$

where Δt is the sampling period. In the present experiments, $\Delta t = 0.40$s, and the maximum number of simulation steps is 280. Let the goal position be p_g. The robot is considered to have reached the goal when $\|p_k - p_g\| \le 1.25$.

For the i-th dynamic obstacle, the state vector is defined as $\mathbf{s}_{i,k} = [x_{i,k}, y_{i,k}, \dot{x}_{i,k}, \dot{y}_{i,k}]^{\mathrm{T}}$, and the observation vector is denoted by $\mathbf{z}_{i,k}$. Obstacle state estimation adopts a constant-velocity model:

$$\begin{cases} \mathbf{s}_{i,k+1} = \mathbf{F}\mathbf{s}_{i,k} + \mathbf{q}_{i,k} \\ \mathbf{z}_{i,k} = \mathbf{H}\mathbf{s}_{i,k} + \mathbf{r}_{i,k} \end{cases} \tag{2}$$

where F denotes the state transition matrix, H represents the observation matrix, and $q_{i,k}$ and $r_{i,k}$ denote the process noise and observation noise, respectively. Since state estimation, cost map update, and path replanning must be performed online within each planning cycle, the predictor is required to capture the short-term motion trendwithout introducing excessive closed-loop overhead.

Based on Eq. (2), the base predicted position $\mathbf{o}_{i,k+\tau}$ and motion direction $\psi_{i,k+\tau}$ over the next N_p steps can be obtained, where N_p is the prediction horizon and $\tau \in \{1, 2, \ldots, N_p\}$. To preserve the interpretability and stability of the Kalman predictor while compensating systematic multi-step errors near turning and reversal motions, a lightweight residual corrector is further introduced. Let $\Delta\mathbf{o}_{i,k+\tau}$ denote the residual sequence predicted by a small

multilayer perceptron from the filtered state, covariance traces, and a short position-history window. The corrected forecast is written as

$$\tilde{o}_{i,k+\tau} = \hat{o}_{i,k+\tau} + \alpha c_{i,k} \Delta o_{i,k+\tau}, \quad c_{i,k} = \mathbb{I}\left(\chi_{i,k} \geq \chi_{\text{th}}\right) \tag{3}$$

where α is the residual gain, $c_{i,k}$ is a curvature gate, $\chi_{i,k}$ is the recent motion curvature, and χ_{th} is the activation threshold. When recent obstacle motion remains nearly linear, $c_{i,k} = 0$ and the predictor reverts to the standard Kalman prediction. In that way, the Kalman filter stays as the primary estimator, while learning only supplies a bounded correction term for the short-horizon prediction block.

2.3. Predictive dynamic cost field

This paper constructs an anisotropic risk kernel aligned with motion direction around the predicted position. For a spatial point p, we first transform its coordinates into a local coordinate system aligned with the obstacle's motion direction to obtain the offset.

$$\left[\Delta x_{i,k+\tau},\, \Delta y_{i,k+\tau}\right]^T = R\left(-\psi_{i,k+\tau}\right)\left(p - \delta_{i,k+\tau}\right) \tag{4}$$

Here, $R(\cdot)$ is the two-dimensional rotation matrix, $\Delta x_{i,k+\tau}$ denotes the relative displacement along the direction of motion, and $\Delta y_{i,k+\tau}$ denotes the lateral offset. This transformation rotates the principal axis of the risk kernel with the obstacle motion direction, so that the forward risk band and the lateral yielding width can be distinguished within the same coordinate frame. Based on Eq. (4), the predictive dynamic cost field is defined as

$$J_{\text{dyn}}(p) = \sum_{i=1}^{N_o} \sum_{\tau=1}^{N_p} \gamma^{\tau} \exp\left(-\frac{(\Delta x_{i,k+\tau})^2}{2\sigma^2_{\parallel,i,k+\tau}} - \frac{(\Delta y_{i,k+\tau})^2}{2\sigma^2_{\perp,i,k+\tau}}\right) \tag{5}$$

where N_o is the number of dynamic obstacles, γ is the temporal decay factor, and $\sigma_{\parallel,i,k+\tau}$ and $\sigma_{\perp,i,k+\tau}$ denote the diffusion scales along and normal to the direction of motion, respectively These two parameters serve different roles. The longitudinal scale $\sigma_{\parallel,i,k+\tau}$ mainly controls the length of the forward risk band and reflects the space that the obstacle may occupy if it keeps moving ahead. By contrast, $\sigma_{\perp,i,k+\tau}$ controls the lateral yielding width and captures the safety margin needed in crossing and passing interactions. The physical meaning and selection criteria of each parameter are as follows.N_o: Number of dynamic obstacles. Determined by sensor range and computation.N_p: Prediction steps. $N_p = 7$ balances prediction ability and error. γ: Time attenuation weight. $\gamma = 0.80$ weakens distant future uncertainty. $\sigma_{\parallel}$: Forward risk range. Reflects velocity uncertainty; set to 1.60. $\sigma_{\perp}$: Lateral safety margin. Ensures passing space; set to 1.00. All parameters are selected to balance prediction accuracy, safety margin, computational efficiency and motion uncertainty. In the current implementation, both scales increase with prediction steps, and the longitudinal scale increases more rapidly because long-horizon uncertainty tends to accumulate along the motion trend. In that sense, Eq. (5) is not just a softened obstacle inflation. It is a direction-aware approximation of future occupied regions.

Static obstacle risk is represented by the clearance cost $J_{\text{sta}}(p)$, and the unified cost map is written as

$$J(p) = \kappa_s J_{\text{sta}}(p) + \kappa_d J_{\text{dyn}}(p) \tag{6}$$

where κ_s and κ_d are the static and dynamic cost weights, respectively. By inserting the predictive risk directly into Eq. (6), the global path can respond to future interactions during path generation rather than being corrected only afterward at the local control stage.

2.4. Complexity metric and mode switching

By introducing dedicated complexity variables as online coordination indicators, the local pressure imposed on the planner can be quantitatively characterized. Let ρ_k denote the density of dynamic obstacles within the sensing range, β_k represent the blockage rate ahead of the current global path, and H_k be the local cost entropy that reflects the conflict uncertainty of local cost distribution. The formulation is defined as follows:

$$H_k = -\sum_{b=1}^{B} p_{b,k} \ln p_{b,k}, \Phi_k \quad = \lambda_1 \rho_k + \lambda_2 \beta_k + \lambda_3 H_k, \tag{7}$$

where B is the number of cost bins, $p_{b,k}$ is the probability of the b-th bin, Φ_k is the integrated complexity score, and $\lambda_1, \lambda_2, \lambda_3$ are the corresponding weights. They are combined in nondimensional form for online use. So the linear combination in Eq. (7) should be read as a coordination score, not as a universal definition of scenario complexity.

On this basis, the planning mode is updated through a hysteresis-based switching function:

$$m_k = \Gamma\left(\Phi_k, m_{k-1}; \theta_L, \theta_H, \delta_\theta\right) \tag{8}$$

where m_k is the planning mode at time k, θ_L and θ_H are the lower and upper complexity thresholds, and δ_θ is the hysteresis width. The hysteresis term is used to prevent frequent switching when Φ_k fluctuates near a threshold, thereby maintaining continuity of the control output. The three modes correspond to different coordination tendencies. Mode A uses a longer global replanning interval and a stronger preference for efficiency, and is suitable for low-pressure regions. Mode B strikes a balance among global path consistency, local risk avoidance, and online computational cost. Mode C shortens the replanning interval and

increases the weight of local risk constraints so as to cope with near-field interactions in high-pressure regions.

2.5. Experimental Scenario Design and Comparison Method

All experiments were implemented in MATLAB. The environment was modeled as a 60×60 grid map, with a fixed start point at $[4, 4]$ and a fixed goal point at $[56, 56]$. To ensure consistent comparison conditions, all algorithms utilized the identical static map, dynamic obstacle trajectories, start/goal configurations, and random seed. For the simple scenario, the controlled variables included a 60×60 grid map and a time step of 0.40 s. For the moderate scenario, the controlled variables were a maximum step count of 280 and a perception range of 11.5. For the complex scenario, the controlled variables consisted of the same start point, goal point, and random seed set.

Eight methods are compared in this paper:

1. **PADP:** with predictive dynamic cost field and complexity-adaptive coordination;
2. **PADP-w/oPred:** without prediction, only using current dynamic risk;
3. **PADP-w/oAdapt:** with prediction but fixed coordination, no adaptive adjustment;
4. **PADP-Base:** without prediction and adaptive coordination;
5. **A*:** static global path tracking;
6. **DWA:** local obstacle avoidance baseline;
7. **A*+APF:** combination of static global path and local artificial potential field avoidance;
8. **D*Lite:** incremental dynamic replanning.

2.6. Experimental Parameter Design and Data Collection

The parameters described below are uniformly adopted as the default operating configuration across all comparison and ablation experiments to ensure fairness and reproducibility.

In the map module, the static inflation radius and clearance kernel standard deviation are set to 2 and 1.8, providing a balance between safety margin and map smoothness. For the robot model, the radius is 0.90, while the maximum linear velocity, angular velocity, and acceleration are set to 1.45, 1.25, and 1.10, respectively, ensuring kinematic feasibility.

In the prediction module, the horizon is fixed at $N_p = 7$, and the temporal decay factor is $\gamma = 0.80$, prioritizing near-term predictions. The lateral and longitudinal diffusion scales are set to 1.00 and 1.60 to capture anisotropic motion uncertainty.

For the cost function, the static and dynamic weights are $\kappa_s = 1.40$ and $\kappa_d = 2.35$, emphasizing dynamic obstacle influence while maintaining environmental constraints. In the complexity regulation module, the weights are $[\lambda_1, \lambda_2, \lambda_3] = [0.38, 0.37, 0.25]$, with thresholds $[\theta_L, \theta_H] = [0.38, 0.64]$ and hysteresis width $\delta_\theta = 0.05$ to reduce frequent mode switching.

For global replanning, the intervals of Modes A/B/C are set to 12/5/2. Correspondingly, the local rollout horizons for Modes A/B/C are 4/6/7, enabling adaptive planning under different environment complexities.This unified configuration ensures consistency and reproducibility, allowing performance differences to be attributed solely to algorithmic design.

For the KF residual extension module, all baseline PADP parameters are fully retained. The residual predictor adopts a lightweight MLP with two hidden layers, each containing 64 neurons. It takes the filtered obstacle states, covariance traces, and the most recent five position increments as inputs, and outputs a 7-step residual correction term. The residual gain and curvature gating threshold are set to 0.75 and 0.22 rad, respectively, while the maximum single-step correction is limited to 2.4 grid cells. Since obstacle motion tends to be nearly linear in simple scenarios, the residual model is trained and evaluated solely on moderate and complex scenarios.

For the learning-based residual module, random seeds 1–8, 9–10, and 11–12 are used for training, validation, and testing. To ensure fairness, seeds are reset per episode so that paired runs share identical observation noise. Since path length is computed only for successful runs, large differences in failure rates across complex scenarios may introduce sample selection bias; thus, path length is reported only as supplementary information and not used for main performance conclusions.

3. Results and discussion

3.1. Overall comparison

Table 1 summarizes the performance on the benchmark scenarios. PADP achieves the highest overall success rate and the lowest average number of collisions among the compared methods. D*Lite, DWA, and A*+APF represent incremental global replanning, reactive local avoidance, and typical global and local hybrid strategies, respectively, while A* serves as a static baseline. Compared with DLite,

Table 1. A comprehensive comparison of 36 scenarios across all methods

Method	Success rate (%)	Mean collisions	Replanning count	Planning time (ms)
PADP	80.56	0.194	52.69	6.98
PADP-w/oPred	61.11	0.389	33.86	5.57
PADP-w/oAdapt	77.78	0.222	60.00	8.99
PADP-Base	61.11	0.333	49.89	6.46
A*	33.33	0.667	0.00	0.02
DWA	41.67	0.583	0.00	4.30
A*+APF	30.56	0.694	0.00	0.04
D*Lite	52.78	0.472	152.64	19.29

Table 2. Results of the PADP variants across scenario categories

Scenario	Method	Success rate (%)	Mean collisions	Replanning count	Planning time (ms)
Simple	PADP	100.00	0.000	12.00	2.06
Simple	PADP-w/oPred	83.33	0.167	11.67	1.32
Simple	PADP-w/oAdapt	100.00	0.000	35.50	2.47
Simple	PADP-Base	100.00	0.000	35.42	1.63
Medium	PADP	100.00	0.000	20.25	8.89
Medium	PADP-w/oPred	75.00	0.250	19.83	6.62
Medium	PADP-w/oAdapt	100.00	0.000	44.83	13.33
Medium	PADP-Base	66.67	0.333	40.50	8.92
Complex	PADP	41.67	0.583	125.83	9.98
Complex	PADP-w/oPred	25.00	0.750	70.08	8.75
Complex	PADP-w/oAdapt	33.33	0.667	99.67	11.17
Complex	PADP-Base	16.67	0.667	73.75	8.83

PADP increases the success rate by 27.78 percentage points, reduces collisions from 0.472 to 0.194, and decreases planning time from 19.29 ms to 6.98 ms. It also shows higher success rates than DWA and A*+APF. Overall, these results suggest that PADP improves the balance among success rate, safety, and computational cost under the current experimental settings.

Fig. 2 tells a fairly consistent story. Differences are minor in simple scenarios but become significant as interactions grow denser and more frequent. PADP stays at 100% success in simple and medium scenarios, then drops to 41.67% in complex ones. D* Lite remains comparatively competitive in medium scenarios, but it does so with many more replanning events and higher planning time. By contrast, A*, DWA, and A*+APF deteriorate sharply in complex scenes. Notably, the separation among methods becomes most visible when dynamic interference intensifies and path intersections accumulate.

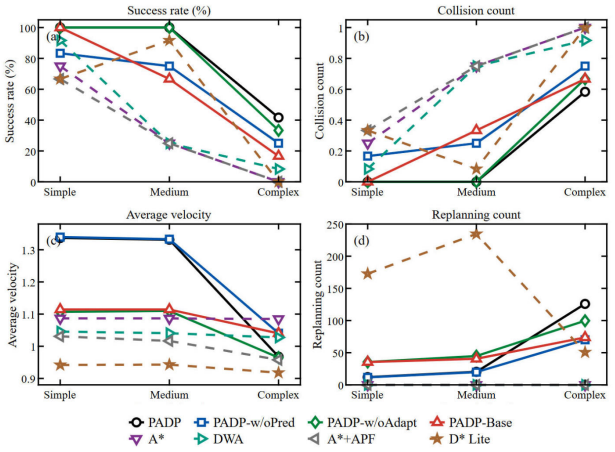

Fig. 2. Trends of core metrics for different methods across various scenarios

3.2. Analysis of the Roles of Prediction and Mode Switching

As shown in Table 2, although the roles of the two modules cannot be completely isolated, it is clear that the prediction mechanism is more closely associated with the reachability of complex scenarios, while the complexity-adaptive coordination mainly affects the resource allocation strategy of the planner. The prediction mechanism determines the upper bound of reachability, whereas the coordination mechanism optimizes online resource distribution. Based on the current descriptive statistics, compared with PADP-w/oPred, PADP improves the success rate in complex scenarios from 25.00% to 41.67% and reduces the average collision number to 0.583. In medium scenarios, the success rate

increases from 75.00% to 100.00%, demonstrating that relying only on current position risk fails to guarantee reachability under intensified interactive motions.

After removing the adaptive mechanism, PADP reduces the replanning count to 20.25 and increases the average velocity to 1.331 in medium scenarios, without affecting success rate. In complex scenarios, it shows a slight improvement in success rate at the cost of more frequent replanning. These results suggest that adaptive coordination primarily regulates resource allocation between global and local planning rather than directly reducing replanning frequency.

Mode switching varies notably across scenarios. PADP remains in Mode A for simple cases. In medium scenarios, Mode A dominates (96.57%), with limited activation of Mode B (3.43%). In complex scenarios, Mode B becomes dominant (91.72%), with occasional use of Mode C (5.14%). This indicates that the adaptive mechanism is mainly triggered under higher scene complexity, adjusting the balance between global and local planning as interaction intensity increases.

3.3. Ablation Experiment Analysis of the KFRes Module

To evaluate the necessity of the residual module and its auxiliary designs, four variants are compared on moderate and complex scenarios: PADP without residual correction, PADP-KFRes-w/oGate, PADP-KFRes-w/oClip, and the full PADP-KFRes. Simple scenarios are excluded due to near-linear obstacle motion, which limits the effect of residual correction.

At the prediction level, the validation RMSE decreases from 1.053 to 0.457 after training. A gain scan shows that setting the residual gain to 0.75 maintains the success rate 75.00% and collision mean 0.250, while reducing planning time from 21.53 ms to 18.50 ms. This setting is therefore adopted in the subsequent ablation.

Table 3 shows that curvature gating is the more critical internal design. When gating is removed, residual correction is also applied to approximately linear motion segments, and the success rate in complex scenarios drops from 58.33% to 50.00%. This reflects the noise disturbance of the learning model under non-specific distributions. The average number of collisions rises from 0.417 to 0.500, and the overall success rate decreases from 79.17% to 75.00%. These results indicate that the learned residual is not equally beneficial at all stages and must therefore be used selectively through conditional activation.

Compared with the full PADP-KFRes, the clipping mechanism plays a secondary yet non-negligible role. Removing clipping does not affect success rate or collision statistics on the full subset, but increases the average planning time from 17.20 ms to 17.46 ms, and from 20.86 ms to 21.52 ms in complex scenarios. This indicates that clipping is not critical for reachability, but effectively constrains correction magnitude and enhances online stability.

Similar trends are observed on the held-out test set. The success rate of PADP-KFRes-w/oGate decreases to 75.00%, with a drop to 50.00% in complex scenarios, whereas the full PADP-KFRes maintains 100% success and zero collisions. Although PADP-KFRes-w/oClip remains safe on this small test set, the full model demonstrates more consistent performance across both the test set and the full subset.

3.4. Performance Validation and Analysis in Complex Dynamic Scenarios

Fig. 3 visually presents the path planning results in complex scenarios. It can be observed that only PADP and PADP-w/oAdapt successfully reached the target, while all other methods failed before completing the task. In the representative complex environment, PADP generated a path deviation earlier before entering the obstacle intersection area. This indicates to a certain extent that the predictive risk field has affected the upstream path selection, rather than only functioning in the final stage of local obstacle avoidance.

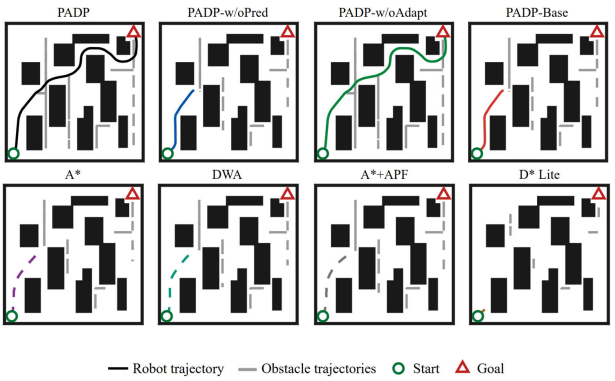

Fig. 3. Trajectory overview of different methods in complex scenarios

3.5. Stability Analysis Under Complex Scenarios

Fig. 4 shows that significant random seed fluctuations still exist in complex scenarios. PADP exhibits certain dispersion in success rate, speed, and planning time, indicating that the method cannot fully eliminate the uncertainty introduced by complex dynamic interactions. However, its collision distribution is more concentrated than those of PADP-w/oPred, PADP-Base, and traditional baseline meth-

Table 3. Ablation study results of the Kalman filter residual module

Level	Method	Success rate (%)	Mean collisions	Replanning count	Planning time (ms)
Medium	PADP	100.00	0.000	20.33	13.35
Medium	PADP-KFRes-w/oGate	100.00	0.000	20.42	15.12
Medium	PADP-KFRes-w/oClip	100.00	0.000	20.42	13.40
Medium	PADP-KFRes	100.00	0.000	20.42	13.53
Complex	PADP	58.33	0.417	130.17	20.14
Complex	PADP-KFRes-w/oGate	50.00	0.500	127.67	23.00
Complex	PADP-KFRes-w/oClip	58.33	0.417	129.50	21.52
Complex	PADP-KFRes	58.33	0.417	129.75	20.86
Overall	PADP	79.17	0.208	75.25	16.74
Overall	PADP-KFRes-w/oGate	75.00	0.250	74.04	19.06
Overall	PADP-KFRes-w/oClip	79.17	0.208	74.96	17.46
Overall	PADP-KFRes	79.17	0.208	75.08	17.20

ods, and successful runs cover a wider range of random seeds. This confirms its comparative advantages in stability, though such benefits are still scenario-dependent.

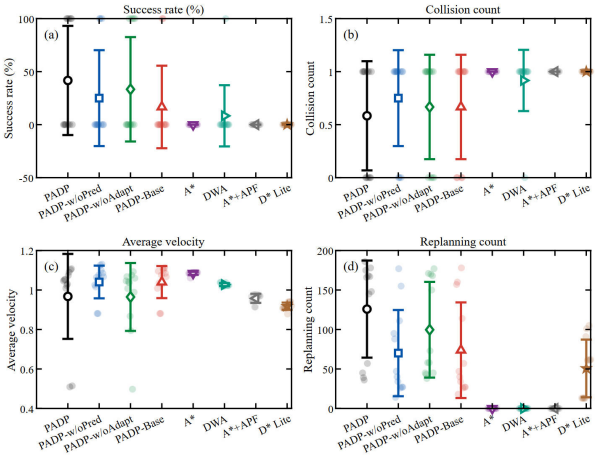

Fig. 4. Means and standard deviations over multiple random seeds in complex scenarios

4. Conclusions

This paper presents a path planning approach for mobile robots in dynamic grid environments, integrating Kalman-based prediction with complexity-adaptive coordination, and extending it with a residual prediction module. The method incorporates predicted future risk into global planning and adjusts global and local interaction via online complexity metrics, forming a unified closed-loop scheme. Across 36 simulated scenarios, PADP shows competitive performance in success rate, collision avoidance, and computational cost. Ablation results suggest that prediction mainly affects reachability in complex scenarios, while adaptive coordination regulates runtime resource allocation.

For the KF-Res residual extension, curvature gating enables selective activation, and residual clipping improves stability. PADP-KFRes reduces prediction error and planning time without degrading success rate or collision performance. These results are obtained under grid-based simulation settings, which define the scope of applicability.

5. Acknowledgments

This work was supported by the Liaoning Provincial Graduate Education and Teaching Reform Research Project grant number LNYJG2023077 and LJ212410144022.

References

[1] G. Fragapane, R. de Koster, F. Sgarbossa, et al., (2021) *"Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda"* **European Journal of Operational Research *294***(2): 405–426. DOI: 10.1016/j.ejor.2021.01.019.

[2] M. Aizat, A. Azmin, and W. Rahiman, (2023) *"A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding"* **IEEE Access *11***: 33934–33955. DOI: 10.1109/ACCESS.2023.3263734.

[3] R. Moller, A. Furnari, S. Battiato, et al., (2021) *"A survey on human-aware robot navigation"* **Robotics and Autonomous Systems *145***: 103837. DOI: 10.1016/j.robot.2021.103837.

[4] C. Mavrogiannis, F. Baldini, A. Wang, et al., (2023) *"Core challenges of social robot navigation: A survey"* **ACM Transactions on Human-Robot Interaction *12***(3): 1–39. DOI: 10.1145/3583741.

[5] X. Deng, R. Li, L. Zhao, et al., (2021) *"Multi-obstacle path planning and optimization for mobile robot"* **Expert Systems with Applications *183***: 115445. DOI: 10.1016/j.eswa.2021.115445.

[6] K. Katona, H. Neamah, and P. Korondi, (2024) *"Obstacle avoidance and path planning methods for autonomous navigation of mobile robot"* **Sensors *24***(11): 3573. DOI: 10.3390/s24113573.

[7] Y. Li, R. Jin, X. Xu, et al., (2022) *"A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach"* **IEEE Access *10***: 57736–57747. DOI: 10.1109/ACCESS.2022.3179397.

[8] M. Kobayashi, H. Zushi, T. Nakamura, et al., (2023) *"Local path planning: Dynamic window approach with Q-learning considering congestion environments for mobile robot"* **IEEE Access *11***: 96733–96742. DOI: 10.1109/ACCESS.2023.3311023.

[9] Y. Sun, W. Wang, M. Xu, et al., (2023) *"Local path planning for mobile robots based on fuzzy dynamic window algorithm"* **Sensors *23***(19): 8260. DOI: 10.3390/s23198260.

[10] L. Xiang, X. Li, H. Liu, et al., (2022) *"Parameter fuzzy self-adaptive dynamic window approach for local path planning of wheeled robot"* **IEEE Open Journal of Intelligent Transportation Systems *3***: 1–6. DOI: 10.1109/OJITS.2021.3137931.

[11] M. Shoeib, J. Lewandowski, and A. Omara, (2024) *"A novel methodology for vision-based path planning and obstacle avoidance in mobile robot applications"* **Advanced Robotics *38***(12): 802–817. DOI: 10.1080/01691864.2024.2315591.

[12] K. Zheng, (2024) *"Autonomous obstacle avoidance and trajectory planning for mobile robot based on dual-loop trajectory tracking control and improved artificial potential field method"* **Actuators *13***(1): 37. DOI: 10.3390/act13010037.

[13] S. Shao, J. Zhang, T. Wang, et al., (2024) *"Dynamic obstacle-avoidance algorithm for multi-robot flocking based on improved artificial potential field"* **IEEE Transactions on Consumer Electronics *70***(1): 4388–4399. DOI: 10.1109/TCE.2023.3340327.

[14] C. Gong, Y. Yang, L. Yuan, et al., (2022) *"An improved ant colony algorithm for integrating global path planning and local obstacle avoidance for mobile robot in dynamic environment"* **Mathematical Biosciences and Engineering *19***(12): 12405–12426. DOI: 10.3934/mbe.2022579.

[15] J. Hou, W. Jiang, Z. Luo, et al., (2024) *"Dynamic path planning for mobile robots by integrating improved sparrow search algorithm and dynamic window approach"* **Actuators *13***(1): 24. DOI: 10.3390/act13010024.

[16] Y. Li, J. Zhao, Z. Chen, et al., (2023) *"A robot path planning method based on improved genetic algorithm and improved dynamic window approach"* **Sustainability *15***(5): 4656. DOI: 10.3390/su15054656.

[17] W. Chen, L. Liu, L. Zhang, et al., (2025) *"Path planning of mobile robots with an improved grey wolf optimizer and dynamic window approach"* **Applied Sciences *15***(7): 3999. DOI: 10.3390/app15073999.

[18] M. Pei, H. An, B. Liu, et al., (2022) *"An improved Dyna-Q algorithm for mobile robot path planning in unknown dynamic environment"* **IEEE Transactions on Systems, Man, and Cybernetics: Systems *52***(7): 4415–4425. DOI: 10.1109/TSMC.2021.3096935.

[19] R. Huang, C. Qin, J. Li, et al., (2023) *"Path planning of mobile robot in unknown dynamic continuous environment using reward-modified deep Q-network"* **Optimal Control Applications and Methods *44***(3): 1570–1587. DOI: 10.1002/oca.2781.

[20] Y. Zhang and P. Chen, (2023) *"Path planning of a mobile robot for a dynamic indoor environment based on an SAC-LSTM algorithm"* **Sensors *23***(24): 9802. DOI: 10.3390/s23249802.

[21] Y. Chen, Y. Wang, B. Li, et al., (2024) *"Multi-robot navigation based on velocity obstacle prediction in dynamic crowded environments"* **Industrial Robot *51***(4): 607–616. DOI: 10.1108/IR-12-2023-0337.

[22] H. Wang, L. He, S. Zhang, et al., (2025) *"Mobile robot path planning considering obstacle gap features"* **Applied Sciences *15***(11): 5979. DOI: 10.3390/app15115979.