

Failure-Aware LLM-DWA Replanning for Mobile Robot Navigation in Dynamic Obstacle Environments

Dabin Kim, Youngmin Lee, Jeonghee Seo, Andrew Jaeyong Choi*

Artificial Intelligence Major, Department of AI-Software, Gachon University
Seongnam, 13120, Korea ({kdb724, dudals4087, sjh0621, andrewjchoi}@gachon.ac.kr) * Corresponding author

Abstract: This paper presents a feasibility study of failure-aware LLM-DWA replanning for mobile robot navigation in dynamic obstacle environments. Existing LLM-DWA navigation has shown promise in static maze-like settings, but moving obstacles can invalidate initially generated waypoints and cause local planners to oscillate, time out, or fall into local-minimum states. We propose an event-triggered strategy that requests new LLM-generated waypoints when either a low-speed no-progress condition or repeated recovery evidence is detected, subject to a replanning cooldown. In ten trials per method, the proposed strategy achieved the highest observed success rate (5/10), compared with NavFn-DWA (0/10), one-shot LLM-DWA (1/10), and fixed-periodic LLM-DWA (0/10 for both intervals). Exploratory two-sided Fisher exact tests favor the proposed method over each 0/10 baseline ($p = 0.0325$) but not over one-shot LLM-DWA ($p = 0.1409$), and no comparison remains significant after correction for four comparisons. These results demonstrate the feasibility of selective failure-aware replanning in dynamic obstacle environments and motivate further evaluation across diverse environments with comprehensive trial-level metrics.

Keywords: Mobile robot navigation, LLM-DWA, failure-aware replanning, dynamic obstacles, waypoint generation

1. INTRODUCTION

Reliable navigation is a fundamental capability for autonomous mobile robots operating in cluttered and changing environments. Classical navigation systems typically combine a global planner with a local planner that generates safe short-horizon motions. The Dynamic Window Approach (DWA) is widely used because it evaluates admissible velocity commands under robot dynamics and obstacle constraints [1]. However, as a local planner, DWA is vulnerable to local minima in U-shaped or maze-like environments, where the robot may repeatedly oscillate, stop, or fail to reach the goal.

Recent advances in large language models (LLMs) have motivated their use as high-level planning modules for mobile robots. Instead of directly controlling low-level velocities, an LLM can generate semantic hints or intermediate waypoints that guide a conventional local planner [2]-[5]. The LLM-DWA framework proposed by Seo et al. combines LLM-generated waypoints with DWA and demonstrates improved performance in static U-shaped environments [5]. Nevertheless, extending this idea to dynamic obstacle environments remains challenging. Moving obstacles can rapidly invalidate a previously generated waypoint sequence, while frequent replanning can cancel active goals and destabilize the navigation stack.

This paper studies whether LLM-DWA navigation can be extended to dynamic maze environments through failure-aware replanning. The goal is not to claim fully robust dynamic navigation, but to provide an initial feasibility analysis showing when event-triggered LLM replanning helps and where it still fails. The main contributions are: (1) a failure-aware LLM-DWA replanning strategy based on local planner failure

signals, (2) a comparative evaluation of a NavFn-DWA baseline (NavFn global planner with DWA local planner), one-shot LLM-DWA, fixed-periodic LLM-DWA, and the proposed replanning method in a dynamic maze, and (3) a failure-mode analysis of local minima, excessive replanning, stale waypoints, and obstacle-induced timeout behavior.

2. RELATED WORK

LLM-based navigation has recently been studied as a way to provide high-level planning cues for embodied agents and mobile robots. Instead of replacing classical planners, LLMs are often used to generate semantic hints, subgoals, or waypoint sequences from textual or structured map descriptions [2]-[4]. These approaches show that language models can support long-horizon reasoning, but they also reveal a persistent limitation: LLM-generated plans may not be physically grounded in the actual geometry and dynamics of the environment.

The LLM-DWA framework is directly related to this study because it combines LLM waypoint generation with the Dynamic Window Approach [5]. This hybrid structure is attractive because the LLM can provide global route guidance while DWA handles short-horizon collision avoidance. However, prior evaluations mainly focus on static or relatively simple obstacle settings. In contrast, our study examines whether the same idea can remain feasible when obstacles move and the validity of waypoints changes during execution.

Replanning has also been studied as a way to recover from plan failures in embodied systems [6], [7]. Periodic replanning is simple, but it may waste

computation or destabilize navigation by repeatedly canceling active goals. Event-triggered replanning is more selective because it updates the plan only when the robot shows evidence of failure. Our work follows this direction by using local planner signals, such as progress stagnation, low speed, and recovery frequency, to decide when LLM-based waypoint regeneration should occur.

3. FAILURE-AWARE LLM-DWA REPLANNING FRAMEWORK

3.1 LLM-assisted waypoint navigation

The system follows a hierarchical structure. The LLM receives the current robot pose, final goal, and map-level information and produces a sequence of intermediate waypoints. The local planner then follows each waypoint while reacting to nearby obstacles. This division preserves the real-time obstacle avoidance capability of DWA while using the LLM to provide global route hints.

However, LLM-generated waypoints are not guaranteed to be geometrically valid or dynamically appropriate. Prior work reports that language models can provide useful semantic heuristics but may suffer from grounding errors and spatial hallucinations [2], [3], [10]. Therefore, in dynamic environments, the waypoint sequence must not be treated as a fixed plan. The navigation system must decide when the current plan is no longer useful and when LLM replanning is worth the cost.

3.2 Event-triggered failure-aware replanning

The proposed strategy monitors local navigation signals and requests new waypoints only when failure-like behavior persists. Let $d(t)$ be Nav2's remaining path distance, b the best recorded value of d , t_p the last progress time, $N_{rec}(t; W_r)$ the number of recovery-count increments in the preceding window W_r , and t_{rep} the most recent replanning time. The thresholds ϵ_{d} , v_{min} , T_s , N_r , W_r , and T_c denote the progress, speed, stagnation-time, recovery-count, recovery-window, and cooldown settings, respectively. The implemented monitor is defined by (1)-(5).

$$\text{if } d(t) < b - \epsilon_d: \quad b \leftarrow d(t), \quad t_p \leftarrow t \quad (1)$$

$$s(t) = \sqrt{v_x(t)^2 + v_y(t)^2} + 0.1 |\omega_z(t)| \quad (2)$$

$$S(t) = 1[t - t_p \geq T_s] \quad (3)$$

$$L(t) = 1[s(t) < v_{min}]$$

$$Q(t) = 1[N_{rec}(t; W_r) \geq N_r] \quad (4)$$

$$C(t) = 1[t - t_{rep} \geq T_c]$$

$$R_{mon}(t) = C(t) \wedge$$

$$[(L(t) \wedge S(t)) \vee Q(t)] \quad (5)$$

Equation (1) updates the progress reference only after an improvement greater than ϵ_{d} . Equation (2) matches the scalar speed used by the implementation. Equations (3)-(5) show that the monitored trigger has two alternative branches: current low speed together with a sufficiently long interval since the last progress update, or a threshold number of recent recovery events. Both branches are blocked during the replanning cooldown. A request discards the remaining waypoints and regenerates a sequence from the current pose to the final goal. The implementation also requests replanning after Nav2 goal rejection, a non-success action result other than an intentional cancellation, or an action-feedback timeout; these are safeguard paths outside $R_{mon}(t)$.

4. DYNAMIC MAZE SIMULATION SETUP

The robot starts at (-8.0, -8.0) and must reach (8.0, 6.0) in a maze-like map. Three box-shaped dynamic obstacles move with constant reciprocating motion at 0.15 m/s. Obstacle A moves along $y = -5$ from $x = -2.2$ to $x = 2.2$ and starts at $x = -2.2$ moving right. Obstacle B moves along $y = 0$ over the same x interval and starts at $x = 2.2$ moving left. Obstacle C moves along $y = 5$ and starts at $x = 0.0$ moving right.

The LLM used in this study was GPT-4o-mini with a temperature of 0.5. The LLM input consisted of the robot's current pose, the final goal position, and map-level obstacle information extracted from the simulation world file. The obstacle information was represented as rectangular objects, where each obstacle was described as (x , y , width, depth, height). Dynamic obstacle positions were not directly provided to the LLM at every control step. Instead, the LLM generated high-level waypoints based on the static map structure, while dynamic obstacle interactions were handled by the local motion controller and the failure-aware replanning trigger.

The LLM output was a JSON-formatted waypoint list. Each waypoint was represented as a two-dimensional coordinate $[x, y]$ in the map frame. The number of waypoints was not fixed and depended on the LLM response. However, the first and last points were constrained to correspond to the start and goal positions. The generated waypoints were not directly used as a continuous trajectory. Instead, each waypoint was sequentially assigned as an intermediate navigation goal. The navigation system then generated and executed a local path between consecutive waypoints using a global planner and a DWA-based local motion controller.

For the main Proposed-40 setting, $T_s = 40$ s, $\epsilon_d = 0.2$ m, $v_{min} = 0.016$ m/s, $N_r = 5$, $W_r = 60$ s, and $T_c = 40$ s. $R_{mon}(t)$ is evaluated on each Nav2 feedback update, while a 1 s monitor checks for a 15 s feedback timeout. Recovery events are recorded whenever Nav2's

cumulative recovery count increases. After $R_{mon}(t)$ becomes true, one replanning request is issued, t_{rep} is updated, and monitored requests are suppressed for T_c . The contact-based trigger implemented for the ablation was disabled in Proposed-40. The maximum numbers of replanning attempts and navigation time were 50 and 600 s, respectively.

A trial was considered successful if the robot reached the final goal within the 600 s navigation limit and the final distance to the goal was less than 0.30 m. A waypoint was considered reached when the robot was within 0.30 m of the waypoint position or when the navigation system reported successful arrival at the corresponding intermediate goal. A trial was counted as a failure if the robot failed to reach the final goal within 600 s, if the number of replanning attempts exceeded 50, or if the navigation action was aborted by the motion planning system.

5. COMPARATIVE EXPERIMENTAL RESULTS

In this study, the baseline denoted as NavFn-DWA uses the NavFn global planner [11] and the DWA local planner [1]. This configuration serves as a classical navigation baseline without LLM-generated waypoint replanning.

Fig. 1 compares the five representative navigation outcomes. The NavFn-DWA baseline does not produce meaningful progress. One-shot LLM-DWA generates an initially plausible route, but in most trials the fixed waypoint sequence becomes stale as obstacle timing changes, leading to unstable navigation or trapping. Periodic LLM-DWA with a 12 s interval creates dense replanning markers and becomes trapped in the left-side dead-end, while the 25 s interval travels farther but still degenerates into a timeout loop. In contrast, the proposed method produces a complete route in the representative successful case by replanning only after accumulated failure evidence. Fig. 1 visualizes representative trajectories rather than all trials; the quantitative success rates over ten trials are summarized in Table 1.

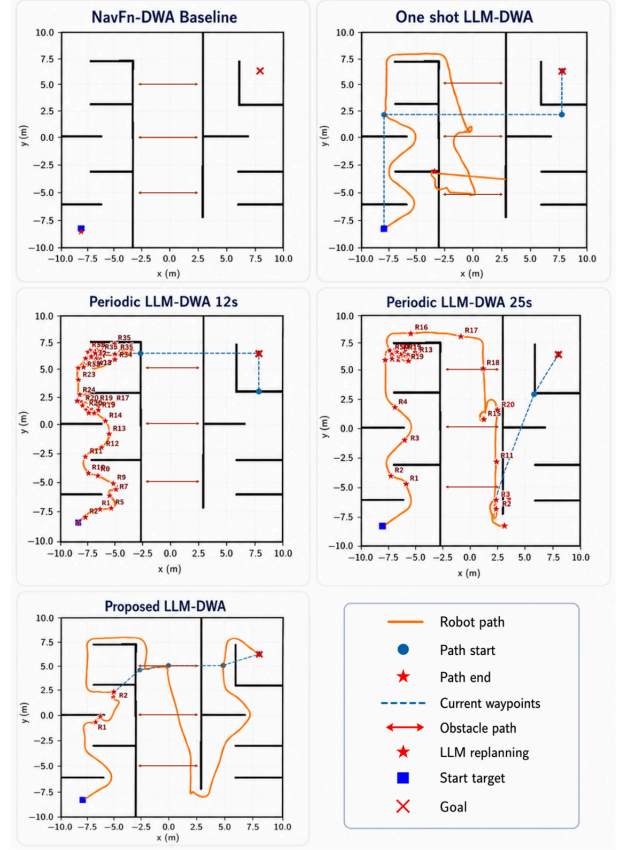


Fig. 1. Representative trajectories of the NavFn-DWA baseline, one-shot LLM-DWA, periodic LLM-DWA with 12 s and 25 s intervals, and the proposed failure-aware LLM-DWA.

Table 1. Comparative results of navigation strategies in the dynamic maze.

Method	Success rate	Representative outcome	Dominant observation
NavFn-DWA baseline	0/10 success	No meaningful progress	Start-stage planning failure
One-shot LLM-DWA	1/10 success	Initial route becomes invalid	Stale global waypoint sequence
Periodic LLM-DWA 12 s	0/10 success	Dense replanning markers	Replanning explosion and left dead-end lock-in
Periodic LLM-DWA 25 s	0/10 success	Timeout trajectory	Delayed recovery and timeout loop
Proposed LLM-DWA	5/10 success	Goal reached in multiple trials	Failure-aware selective replanning

Table 1 summarizes the comparative results and associated failure modes. The proposed method had the highest observed success count. To quantify uncertainty with the available binary outcomes, we performed

exploratory two-sided Fisher exact tests comparing Proposed-40 (5/10) with each baseline. The p-value was 0.0325 against NavFn-DWA, periodic LLM-DWA 12 s, and periodic LLM-DWA 25 s (each 0/10), and 0.1409 against one-shot LLM-DWA (1/10). The 95% Wilson intervals were 0.237-0.763 for Proposed-40, 0-0.278 for each 0/10 baseline, and 0.018-0.404 for one-shot LLM-DWA. With Bonferroni correction for four comparisons ($\alpha = 0.0125$), none of the pairwise differences remained statistically significant. Nevertheless, the proposed method achieved the highest observed success count, supporting its feasibility under the present experimental setting.

Table 2. Ablation results of failure-aware replanning variants over 10 trials.

Variant	Trigger setting	Success rate	Main observation
Proposed-45	45 s / 5 / 45 s	2/10	Conservative trigger; delayed necessary replanning
Proposed-35	35 s / 4 / 35 s	1/10	Faster trigger; timeout failures remained
Proposed-40	40 s / 5 / 40 s	5/10	Best setting; highest success rate; stable selective replanning
Contact-aware	Contact-based trigger	0/10	Over-reactive replanning after physical contact
Temperature-0	Deterministic LLM output	0/10	Deterministic output alone did not resolve failures

In Table 2, the trigger setting represents stagnation time, recovery-trigger count, and replanning cooldown, respectively. Proposed-45 was too conservative and delayed necessary replanning, whereas Proposed-35 reacted faster but retained timeout failures. Proposed-40 achieved the largest observed success count (5/10), suggesting a better balance between premature replanning and delayed recovery in this environment. A representative best run had a 71.552 m path length, 436.315 s navigation time, three LLM calls, and two replanning events. These continuous metrics correspond to a representative successful run; complete trial-level logging will be used in future work to characterize performance distributions. The contact-aware and temperature-zero variants further show that physical-contact triggering can be over-reactive and that deterministic LLM output alone does not resolve local failures.

6. FAILURE ANALYSIS AND DISCUSSION

The remaining failures were mainly associated with physical interactions, particularly cases in which the robot was pushed or trapped between a moving obstacle and a wall. This suggests that high-level waypoint

replanning should be complemented by physically aware local recovery. Although the present evaluation focused on a single controlled dynamic maze, the results provide initial evidence that failure-triggered replanning can outperform fixed or periodic waypoint strategies. Future work will evaluate diverse layouts and obstacle dynamics, collect complete trial-level performance distributions, and integrate map- or VLM-grounded perception with local trap-escape behaviors [8], [9].

7. CONCLUSION

This paper presented a failure-aware LLM-DWA replanning strategy for mobile robot navigation in a dynamic maze. The proposed method regenerates LLM-based waypoints when either low-speed no-progress behavior or repeated recovery evidence is detected, while a cooldown prevents excessive replanning. In the conducted experiments, the proposed method achieved the highest observed success rate of 5/10, compared with 0/10 or 1/10 for the tested baselines. The results show that selective event-triggered replanning provides a more effective balance than one-shot or fixed-periodic waypoint generation in the evaluated dynamic environment. Future work will extend the evaluation to diverse obstacle configurations and motion patterns, report complete trial-level performance distributions, and integrate high-level replanning with physically aware local recovery and improved spatial grounding.

REFERENCES

- [1] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, 1997.
- [2] D. Shah, M. Equi, B. Osinski, F. Xia, B. Ichter, and S. Levine, "Navigation with large language models: Semantic guesswork as a heuristic for planning," *arXiv:2310.10103*, 2023.
- [3] M. Aghzal, E. Plaku, and Z. Yao, "Can large language models be good path planners? A benchmark and investigation on spatial-temporal reasoning," *arXiv:2310.03249*, 2023.
- [4] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "LLM-Planner: Few-shot grounded planning for embodied agents with large language models," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, 2023.
- [5] J. Seo, E. Kim, and A. J. Choi, "LLM-DWA: a hybrid path planning framework combining large

- language models with the dynamic window approach," *Scientific Reports*, vol. 16, Article no. 9898, 2026, doi: 10.1038/s41598-026-39524-1.
- [6] M. Skreta, Z. Zhou, J. L. Yuan, K. Darvish, A. Aspuru-Guzik, and A. Garg, "RePLan: Robotic replanning with perception and language models," *arXiv:2401.04157*, 2024.
- [7] A. Mei, G.-N. Zhu, H. Zhang, and Z. Gan, "ReplanVLM: Replanning robotic tasks with visual language models," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10201-10208, 2024.
- [8] B. Chen et al., "SpatialVLM: Endowing vision-language models with spatial reasoning capabilities," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2024.
- [9] A. J. Sathyamoorthy, K. Weerakoon, M. Elnoor, and D. Manocha, "CoNVOI: Context-aware navigation using vision language models in outdoor and indoor environments," *arXiv:2403.15637*, 2024.
- [10] W. Chen, M.-W. Su, N. Mehjabin, and M. L. Cummings, "Can LLMs plan paths in the real world?" *arXiv preprint*, 2024.
- [11] ROS.org, "navfn: API documentation," ROS Noetic. [Online]. Available: <https://docs.ros.org/en/noetic/api/navfn/html/>. Accessed: Jun. 16, 2026.