

실패 인지 기반 LLM-DWA Replanning을 이용한 이동 로봇 내비게이션

김다빈¹, 이영민¹, 최재용^{1,*}

¹가천대학교 인공지능학과

e-mail : kdb724@gachon.ac.kr, dudals4087@gachon.ac.kr, andrewjchoi@gachon.ac.kr

Failure-Aware LLM-DWA Replanning for Mobile Robot Navigation

Dabin Kim¹, Youngmin Lee¹, and Andrew Jaeyong Choi^{1,*}

¹ School of Computing, Gachon University

I. 서론

Abstract

This study proposes a Failure-Aware LLM-DWA Replanning algorithm for mobile robot navigation in complex static environments. Conventional LLM-DWA generates high-level waypoints only once before navigation, making it difficult to recover when the initial waypoints are inappropriate or the robot becomes stuck in local minima such as U-shaped obstacle environments. The proposed method monitors local planner failure signals, including low progress, low velocity, and recovery behavior, and selectively recalls the LLM to regenerate waypoints from the current robot pose. A waypoint validation module checks whether the generated waypoints are located in free space and do not intersect obstacles. Experiments in a U-shaped obstacle scenario show that the proposed method improves navigation success while reducing recovery behavior and unnecessary LLM calls compared with one-shot and periodic replanning approaches.

자율 로봇의 성공적인 임무 수행을 위해서는 복잡한 정적 환경에서 장애물을 회피하며 목표 지점까지 안정적으로 도달하는 내비게이션 능력이 필요하다. 일반적인 내비게이션 시스템은 global planner와 local planner의 계층적 구조를 따르며, local planner는 센서 정보를 기반으로 실시간 장애물 회피 및 속도 명령 생성을 수행한다.

Dynamic Window Approach(DWA)는 대표적인 local planner로, 로봇의 속도 및 가속도 제약을 고려하여 안전한 주행 궤적을 선택한다[1]. 그러나 DWA는 제한된 국소 정보에 기반하기 때문에 local minima 문제에 취약하다. 특히 U자형 장애물과 같이 local minima가 발생하기 쉬운 구조에서는 로봇이 정체되거나 비효율적인 경로를 반복적으로 탐색하여 목표 지점에 도달하지 못할 수 있다.

최근 LLM을 활용하여 경로 계획의 전역적 탐색이나 중간 waypoint 생성을 보조하려는 연구가 진행되고 있다[2,4,6]. LLM-DWA는 LLM이 high-level waypoint를 생성하고 DWA가 이를 지역적으로 추종하는 방식이다. 그러나 One-shot LLM-DWA는 초기 waypoint가 부적절하거나 주행 중 local minima가 발생할 경우 복구가 어렵고, 생성

waypoint의 유효성 검증도 필요하다.

본 연구에서는 One-shot LLM-DWA의 한계를 보완하기 위해 실패 인지 기반 LLM-DWA replanning 구조를 제안한다. 제안 방식은 진행률, 전속도, recovery count 등 local planner의 실패 신호를 모니터링하고, 기존 waypoint 추종이 어렵다고 판단될 때 현재 로봇 위치를 기준으로 LLM waypoint를 재생성한다. 생성된 waypoint는 검증 모듈을 통해 맵 경계, 장애물 영역, 경로 유효성을 확인한 뒤 주행에 사용된다.

본 연구의 주요 기여는 local planner 실패 신호 기반의 선택적 LLM-DWA replanning 구조를 제안하고, waypoint 검증 모듈을 통해 생성 경로의 유효성을 확인한 점이다. 또한 U자형 장애물 환경에서 기존 방식들과 비교하여 제안 방식의 주행 안정성과 LLM 호출 효율성을 검증하였다.

II. 본론

2.1 DWA 기반 local planner

Dynamic Window Approach(DWA)는 로봇의 속도 및 가속도 제약을 고려하여 실시간 속도 명령을 생성하는 대표적인 local planner이다[1]. 본 연구에서는 DWA를 LLM이 생성한 waypoint를 추종하는 기본 local planner로 사용한다. DWA의 평가 함수는 일반적으로 목표 방향성, 장애물과의 거리, 속도 유지, 충돌 가능성 등을 포함하며 다음과 같이 표현할 수 있다.

$$J(v, \omega) = \alpha H(v, \omega) + \beta D(v, \omega) + \gamma V(v, \omega) + \delta O(v, \omega)$$

여기서 $H(v, \omega)$ 는 목표 방향과의 정렬 정도, $D(v, \omega)$ 는 장애물과의 거리, $V(v, \omega)$ 는 전진 속도, $O(v, \omega)$ 는 장애물 충돌 위험도를 의미하며, $\alpha, \beta, \gamma, \delta$ 는 각 항의 가중치이다. 그러나 DWA는 제한된 지역 정보에 기반하기 때문에 복잡한 정적 환경에서 local minima에 빠질 수 있다. 특히 U자형 장애물 환경에서는 목표 방향으로의 단기 이동이 막힌 경로를 유도하여, 로봇이 정체되거나 반복적인 복귀 동작을 수행할 수 있다.

2.2 LLM 기반 waypoint 생성과 한계

LLM은 환경 설명, 목표 위치, 지도 정보 등을 입력받아 로봇이 추종할 고수준 waypoint나 위상적 경로 단서를 생성하는 데 활용될 수 있다[2,5,6]. 이러한 고수준 정보는 LLM-DWA 구조에서 중간 waypoint 시퀀스로 사용되며, DWA가 각 waypoint를 지역적으로 추종함으로써 DWA의 전역 정보 부족

문제를 완화한다.

One-shot LLM-DWA는 주행 전 한 번만 LLM을 호출하므로, 초기 waypoint가 부적절하거나 주행 중 정체가 발생할 경우 수정이 어렵다. 또한 LLM waypoint가 장애물 내부나 지도 외부에 생성될 수 있으므로 replanning과 유효성 검증이 필요하다.

2.3 실패 인지 기반 LLM-DWA replanning

본 연구에서는 One-shot LLM-DWA의 한계를 보완하기 위해 실패 인지 기반 LLM-DWA replanning 구조를 제안한다. 그림1은 제안 방식의 전체 흐름을 나타낸다.

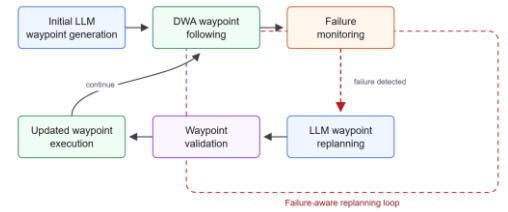


그림 1. 실패 인지 기반 LLM-DWA replanning 구조

초기 단계에서는 LLM이 시작 위치와 목표 위치, 지도 정보를 기반으로 waypoint 시퀀스를 생성한다. 이후 DWA는 생성된 waypoint를 순차적으로 추종한다. 주행 중 로봇이 기존 waypoint를 안정적으로 따라가지 못하는 경우, 실패 감지 모듈은 local planner의 주행 상태를 기반으로 실패 여부를 판단한다.

본 연구에서는 다음과 같은 신호를 실패 판단에 활용한다.

$$\begin{aligned} \Delta d_i &= d_i(t - T_s) - d_i(t) \\ \Delta d_i &< \varepsilon_p \\ v(t) &< v_{min} \\ N_{rec}(T_r) &\geq N_{rec,min} \end{aligned}$$

여기서 $d_{i(t)}$ 는 현재 시점에서 목표 waypoint까지의 거리, Δd_i 는 시간 구간 T_s 동안 감소한 목표 거리, ε_p 는 최소 진행률 임계값, v_{min} 은 저속 판단 임계값이다. 또한 $N_{rec}(T_r)$ 는 시간 구간 T_r 동안 발생한 recovery 횟수이며, $N_{rec,min}$ 은 recovery 기반 failure 판단 임계값이다. 즉, 일정 시간 동안 목표 waypoint까지의 거리가 충분히 감소하지 않고 로봇의 전속도가 낮게 유지되거나, recovery가 반복적으로 발생하면 waypoint 추종 실패로 판단한다.

실패 감지 파라미터는 일시적 정체로 인한 불필요한 replanning을 방지하기 위해 DWA Only baseline의 대표 성공 주행 결과를 기준으로 보수적으로 설정하였다.

Parameter	Symbol	Value	Basis
stuck-time-sec	T_s	60 s	DWA Only 주행 시간 90 s의 약 2/3
progress-epsilon	ϵ_p	0.2 m	60 s 동안 기대 이동량 9.84 m의 약 2%
low-speed-threshold	v_{min}	0.016 m/s	DWA Only 평균 속도 0.164 m/s의 약 10%
recovery-trigger-count	$N_{rec,min}$	7	성공한 DWA baseline의 recovery 6회보다 큰 값
recovery-window-sec	T_r	60 s	DWA Only 주행 시간 90 s의 약 2/3
replan-cooldown-sec	T_c	30 s	새 waypoint 추종 시간을 보장하기 위한 cooldown

표 1. 실패 감지 파라미터 설정

따라서 제안 방식은 순간적인 속도 저하가 아니라, 진행을 저하와 저속 상태가 지속되거나 recovery가 반복되는 경우에만 현재 로봇 위치를 새로운 시작점으로 설정하고 LLM을 재호출한다. 재호출된 LLM은 최종 목표까지의 새로운 waypoint 시퀀스를 생성한다. 이 방식은 단순 주기적 replanning과 달리 실제 실패 상황에서만 LLM을 호출하므로 불필요한 호출을 줄일 수 있다.

2.4 LLM waypoint 검증 모듈

LLM이 생성한 waypoint는 실제 지도에서 유효한지 확인되어야 한다. 본 연구에서는 waypoint 검증 모듈을 통해 생성된 waypoint가 주행 가능한 위치에 있는지 검사한다. 각 waypoint $w_i = (x_i, y_i)$ 에 대해 다음 조건을 확인한다.

첫째, waypoint가 지도 경계 내부에 위치해야 한다.

$$x_{min} \leq x_i \leq x_{max}, \quad y_{min} \leq y_i \leq y_{max}$$

둘째, waypoint는 점유 격자 지도에서 자유 공간에 위치해야 하며, 로봇 반경과 안전 여유를 고려한 inflated obstacle 영역에 포함되지 않아야 한다.

셋째, 연속된 waypoint w_i 와 w_{i+1} 를 연결하는 경로가 장애물을 관통하지 않아야 한다. 이를 위해 두 waypoint 사이의 격자를 샘플링하여 점유 여부를 확인한다.

$$Line(w_i, w_{i+1}) \cap M_{obs} = \emptyset$$

여기서 M_{obs} 는 장애물 셀 집합을 의미한다. 검증 과정에서 무효한 waypoint가 발견되면 해당 waypoint를 제거하거나, 필요한 경우 LLM 재호출을 통해 새로운 waypoint를 요청한다. 이를 통해 LLM의 공간 추론 오류가 실제 주행 실패로 이어지는 것을 방지하고 내비게이션 안전성을 향상시킨다.

III. 구현

3.1 시스템 구현

본 연구에서 제안하는 시스템은 ROS 2 기반 내비게이션 환경에서 구현하였다. 시뮬레이션 환경은 Gazebo를 사용하였으며, 로봇 모델은 TurtleBot3 Burger를 기반으로 구성하였다. 로봇은 LiDAR 센서를 이용하여 주변 장애물 정보를 획득하고, DWA 기반 local planner를 통해 waypoint를 순차적으로 추종한다.

LLM 연동 모듈은 Python 기반으로 구현하였으며, 초기 waypoint 생성 및 replanning 요청 시 LLM API를 호출한다. LLM 입력 프롬프트에는 경로 계획 문제를 구조화하기 위해 현재 로봇 위치, 목표 위치, 현재 waypoint, 실패 감지 정보가 포함된다. LLM은 새로운 waypoint 시퀀스를 JSON 형식으로 반환하며, 시스템은 이를 파싱한 뒤 waypoint 검증 모듈을 거쳐 내비게이션 목표로 전달한다. 실패 인지 모듈은 2.3의 failure monitor 조건에 따라 replanning 여부를 판단하고, 검증된 waypoint를 목표로 전달한다.

3.2 비교군 설정

제안 방식의 성능을 평가하기 위해 **DWA Only**, **One-shot LLM-DWA**, **Periodic LLM-DWA Replanning**, **Proposed**를 비교하였다. Periodic LLM-DWA Replanning은 12 s와 25 s의 두 주기 설정으로 평가하였다. DWA Only는 LLM waypoint 없이 DWA만 사용하는 기준 방식이며, One-shot LLM-DWA는 주행 전 한 번만 LLM을 호출하는 방식이다. Periodic LLM-DWA는 일정 주기마다 waypoint를 갱신하고, Proposed는 failure monitor가 waypoint 추종 실패를 감지한 경우에만 LLM을 재호출한다.

3.3 실험 시나리오

실험은 복잡한 정적 환경에서 로봇의 주행 성능을 평가하기 위해 U자형 장애물 환경에서 수행하였다. U자형 장애물은 DWA와 같은 local planner가 local minima에 빠지기 쉬운 대표적인 구조로, 로봇이 목표 방향으로 이동하려 할 때 장애물 내부 방향으로 유도되어 정체되거나 반복적인 회피 동작을 수행할 수 있다. 본 시나리오에서 로봇은 시작점에서 목표점까지 이동해야 하며, U자형 장애물 주변에서 각 방법이 정체 상황을 얼마나 효과적으로 회피하거나 복구하는지를 평가하였다. 시작 위치와 목표 위치는 모든 실험에서 동일하게 설정하였으며, 각 방식에 대해 동일한 조건에서 반복 실험을 수행하였다.

3.4 평가지표

본 연구에서는 제안 방식의 성능을 평가하기 위해 주행 성공률, 경로 길이, 주행 시간, recovery count,

LLM call count, replanning count를 사용하였다. 주행 성공률은 다음과 같이 정의한다.

$$Success Rate = \frac{Successful Trials}{Total Trials} \times 100$$

주행 성공률은 로봇이 목표 지점 반경 0.25m 이내에 도달한 비율로 정의한다. 경로 길이는 로봇이 실제 이동한 총 거리이며, 주행 시간은 시작부터 목표 도달 또는 주행 종료까지 소요된 시간이다. recovery count는 Nav2 recovery 동작이 발생한 횟수이며, LLM call count는 주행 중 LLM이 호출된 총 횟수를 의미한다. replanning count는 LLM 호출 이후 waypoint가 갱신된 횟수로 정의한다. 이러한 지표를 통해 제안 방식이 주행 성공률을 향상시키는지, recovery behavior를 줄이는지, 그리고 주기적 replanning 방식보다 적은 LLM 호출로 안정적인 주행을 수행하는지를 평가한다.

3.5 실험 결과 및 고찰

표 2는 U자형 장애물 환경에서 각 방식의 주행 성공률을 비교한 결과를 나타낸다.

Method	Success/Trials	Success Rate
DWA Only	17/20	85.0%
One-shot LLM-DWA	8/20	40.0%
Periodic LLM-DWA, T=12s	18/20	90.0%
Periodic LLM-DWA, T=25s	20/20	100.0%
Proposed Failure-Aware LLM-DWA	20/20	100.0%

표2. U자형 장애물 환경에서의 주행 성공률 비교

DWA Only 방식은 20회 중 17회 성공하여 85.0%의 성공률을 보였으며, One-shot LLM-DWA는 8회만 성공하여 40.0%의 성공률을 보였다. 이는 LLM waypoint 자체의 한계라기보다, 초기 waypoint가 U자형 장애물 내부 방향 또는 장애물 근처로 설정될 경우 이를 주행 중 수정할 수 없는 one-shot 구조의 한계를 보여준다. 이 경우 DWA는 부적절한 waypoint를 지속적으로 추종하면서 local minima에 빠질 수 있으며, 결과적으로 DWA Only보다 낮은 성공률을 보일 수 있다. Periodic LLM-DWA는 12 s 주기에서 90.0%, 25 s 주기에서 100.0%의 성공률을 보였으며, 제안 방식 역시 100.0%의 성공률을 달성하였다. 성공률 비교만으로는 각 방식의 replanning 효율성과 주행 안정성을 충분히 확인하기 어렵기 때문에, 추가적으로 각 방식의 대표 주행 사례에서 경로 길이, 주행 시간, LLM 호출 수, replanning 횟수, recovery 횟수를 비교하였다.

Method	Result	Path Length (m)	Navigation Time (s)	LLM Calls	Replans	Recoveries
DWA Only	Success	14.718	89.988	0	0	6
One-shot LLM-DWA	Failure	37.264	296.450	1	0	16
Periodic LLM-DWA, T=12s	Success	19.183	907.142	76	75	1
Periodic LLM-DWA, T=25s	Success	32.894	608.828	25	24	1
Proposed Failure-Aware LLM-DWA	Success	14.986	79.804	2	1	0

표3. 대표 주행 지표 비교

표 3은 각 방법의 대표 주행 사례에서 경로 효율 및 replanning 관련 지표를 비교한 결과이다. DWA Only는 목표 도달에는 성공하였으나 6회의 recovery가 발생하였고, One-shot LLM-DWA는 초기 waypoint를 갱신하지 못한 상태에서 해당 waypoint를 계속 추종하였고, U자형 장애물 주변에서 정체가 누적되어 목표 도달에 실패하였다. Periodic replanning 방식은 목표 도달에는 성공하였지만, 12 s 주기에서 76회의 LLM 호출과 75회의 replanning이 발생하였다. 반면 제안 방식은 2회의 LLM 호출과 1회의 replanning만으로 목표에 도달하였으며, recovery count는 0으로 나타났다. 즉, Proposed는 Periodic replanning과 동일한 100.0% 성공률을 달성하면서도, LLM 호출 수를 25~76회가 아니라 2회로 줄였고 recovery 없이 목표에 도달하였다. 이는 failure monitor가 Nav2 recovery 이전에 실패 신호를 감지하고 waypoint를 재계획했기 때문이다.

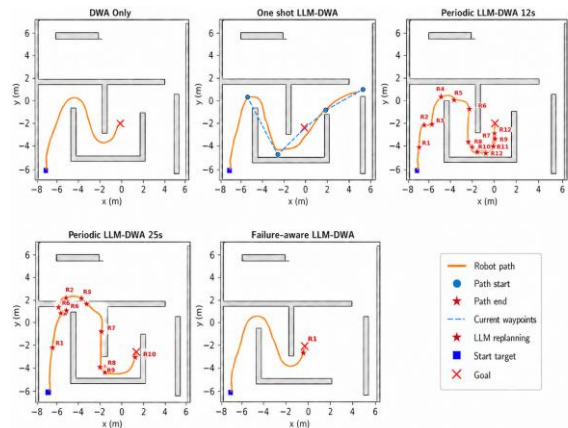


그림 2. 각 방법의 대표 주행 경로 비교

그림 2는 각 방법의 대표 주행 경로를 시각적으로 비

교한 결과이다. DWA Only와 One-shot LLM-DWA는 U자형 장애물 주변에서 목표 방향으로 안정적으로 수렴하지 못하는 경향을 보였으며, Periodic LLM-DWA는 목표 도달에는 성공하였으나 다수의 replanning 지점이 발생하였다. 반면 제안 방식은 1회의 replanning만으로 경로를 갱신하여 목표 지점까지 도달하였다.

IV. 결론 및 향후 연구 방향

4.1 결론

본 논문에서는 복잡한 정적 환경에서 기존 One-shot LLM-DWA의 한계를 보완하기 위한 실패 인지 기반 LLM-DWA Replanning 구조를 제안하였다. 제안 방식은 로봇의 위치 변화, waypoint 진행률, 선속도, recovery count 등 local planner의 실패 신호를 모니터링하고, 실패가 감지된 경우에만 LLM을 재호출하여 현재 위치 기준의 waypoint를 재생성한다. 또한 waypoint 검증 모듈을 통해 지도 경계, 자유 공간 여부, 장애물 관통 여부를 확인함으로써 LLM 기반 waypoint의 안전성을 향상시켰다.

U자형 장애물 환경 실험에서 제안 방식은 20회 중 100.0%의 성공률을 보였으며, 대표 주행 사례에서도 2회의 LLM 호출과 1회의 replanning만으로 목표에 도달하였다. 이를 통해 제안 방식이 One-shot LLM-DWA보다 높은 주행 안정성을 보이고, periodic replanning보다 효율적으로 LLM 호출을 사용할 수 있음을 확인하였다.

4.2 향후 연구

향후 연구에서는 미로형 구조와 같은 더 복잡한 정적 환경 및 동적 장애물 환경으로 실험 범위를 확장할 예정이다. 또한 실제 모바일 로봇 플랫폼에 적용하여 시뮬레이션과 실제 환경 간 성능 차이를 분석하고, 구조화된 지도 정보 또는 VLM 기반 인식 정보를 결합하여 LLM의 공간 추론 정확도를 향상시킬 계획이다.

참고문헌

- [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] 서정희, 최재용, "LLM-DWA: LLM을 활용한 경로 Waypoints 생성 하이브리드 알고리즘", 대한전자공학회 2025년도 하계학술대회 논문집, pp. 2425-2428, 2025.
- [3] X. Gong, Y. Gao, F. Wang, D. Zhu, W. Zhao, F. Wang, and Y. Liu, "A local path planning algorithm for robots based on improved DWA", Electronics, vol. 13, no. 15, article no. 2965, 2024.
- [4] S. Meng, Y. Wang, C.-F. Yang, N. Peng, and K.-W. Chang, "LLM-A*: Large language model enhanced incremental heuristic search on path planning", Findings of the Association for Computational Linguistics: EMNLP, pp. 1119-1132, 2024.
- [5] H. Deguchi, K. Shibata, and S. Taguchi, "Language to map: Topological map generation from natural language path instructions", IEEE International Conference on Robotics and Automation (ICRA), pp. 9556-9562, 2024.
- [6] M. Aghzal, E. Plaku, and Z. Yao, "Look further ahead: Testing the limits of GPT-4 in path planning", IEEE 20th International Conference on Automation Science and Engineering (CASE), 2024.
- [7] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of ChatGPT: The history, status quo and potential future development", IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 5, pp. 1122-1136, 2023.
- [8] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models", Advances in Neural Information Processing Systems, vol. 35, pp. 24824-24837, 2022.