

차동 구동형 로봇의 경로 추종 및 장애물 회피 알고리즘

이동현

금오공과대학교 IT융복합공학과

donglee@kumoh.ac.kr

Path following and obstacle avoidance algorithm for differential-drive robot

Lee Dong-Hyun

Department of IT Convergence Engineering, Kumoh National Institute of Technology

요약

본 논문은 차동 구동형 로봇의 운동역학을 고려하여 주어진 글로벌 경로를 효율적으로 추종함과 동시에 인접한 장애물을 회피할 수 있는 알고리즘을 제안한다. 제안하는 알고리즘의 입력은 글로벌 경로, 로봇의 현재 상태, 거리 측정 센서이며, 출력은 선속도 및 각속도 명령이다. 해당 알고리즘이 생성한 속도 명령은 차동 구동형 로봇 제어기에 전송되어 로봇의 바퀴를 제어함으로써 로봇을 이동시킨다.

I. 서론

일반적인 경로 계획 알고리즘 또는 연산량을 최소화하기 위한 경로 계획 알고리즘은 로봇의 운동역학을 고려하지 않으며, 어느 방향으로도 이동이 가능(홀로노믹 제약)하다고 가정하고 경로를 생성한다. 따라서 로봇 제어기에 장애물 충돌이 없는 글로벌 경로가 주어지더라도 이를 추종하는 과정에서 로봇의 운동 역학적인 제약으로 인해 예러 없는 경로 추종이 어려우며, 이 과정에서 지도상에 존재하는 장애물과 충돌할 수 있다. 또한 이동 중에 지도상에 없는 동적 장애물이 글로벌 경로상에 위치하여 글로벌 경로의 추종을 방해할 수도 있다. 본 논문은 위와 같이 로봇의 제약을 고려하지 않고 생성된 글로벌 경로에 대해 차동 구동형 이동로봇의 운동역학을 고려하여 장애물과 충돌 없이 효율적으로 주어진 글로벌 경로를 추종할 수 있는 속도 명령을 생성하는 알고리즘을 제안한다. 해당 알고리즘이 생성한 속도 명령은 차동 구동형 로봇 제어기에 전송되어 로봇의 바퀴를 제어함으로써 로봇을 이동시킨다.

본 알고리즘과 동일한 목적으로 활용되는 기존의 알고리즘으로는 샘플링을 기반으로 한 dynamic window approach(DWA), 주어진 글로벌 경로를 변형시키으로써 최적화를 수행하는 timed elastic band(TEB), vector field histogram기반의 알고리즘(VFH+, VFH*) 등이 있다. DWA의 경우 샘플링 개수에 따라 성능의 영향을 크게 받으며, 각 샘플에 대한 비용함수를 계산하기 위한 파라미터 튜닝이 필요하다.[1] TEB의 경우 비선형 최적화를 위한 그래프 최적화 알고리즘을 적용해야 하며, 이 과정에서 많은 연산량과 파라미터 튜닝이 필요하다.[2] VFH 기반의 알고리즘의 경우 로봇의 최소회전반경을 0보다 큰 값으로 제한함으로써 제자리회전에 대한 명령을 생성할 수 없으며, 또한 로봇의 현재 속도, 최대속도, 최대가속도에 대한 고려를 하지 않는다.[3][4] 본 논문에서 제안하는 알고리즘은 위와 같은 기존의 알고리즘에 비해 낮은 연산량으로 차동 구동 이동로봇의 운동역학을 고려함과 동시에 효율적으로 장애물 회피가 가능한 속도 명령을 생성할 수 있다.

II. 본론

본 논문에서 제안하는 알고리즘은 그림1과 같이 8단계의 과정을 반복적

으로 수행한다.

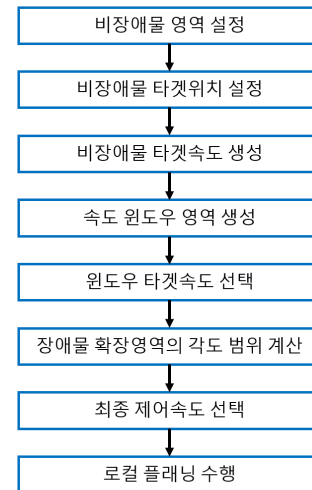


그림1. 제안하는 알고리즘의 8단계 과정

첫 번째 단계에서는 로봇과 주변 장애물 간의 최단 거리를 반경으로 한 비장애물 영역을 설정한다. 로봇을 중심으로 한 비장애물 영역은 라이다와 같은 거리센서를 기반으로 로봇 주변의 장애물과의 거리를 측정하고 그 중에 로봇과 가장 가까운 장애물과의 최단 거리를 반경으로 한 원이다.

두 번째 단계에서는 주어진 글로벌 경로와 비장애물 영역과의 교점을 비장애물 타겟 위치로 결정한다. 비장애물 영역 내에는 장애물이 존재하지 않으므로 로봇과 비장애물 타겟 위치를 연결하는 글로벌 경로 상에는 장애물이 존재하지 않는다. 따라서 로봇은 이동 과정에서 지속적으로 변하는 비장애물 타겟 위치를 목적지로 하여 이동하게 된다.

세 번째 단계에서는 차동 구동 로봇의 선속도와 각속도 간의 허용범위를 고려하여 비장애물 타겟 위치로 이동하기 위한 비장애물 타겟 속도를 생성한다. 이를 위하여 $g(q, q_t)$ 를 정의한다.

$$g(q, q_t) = \begin{cases} \frac{1}{2} (\cos(\frac{\pi}{q_t} q) + 1) & \text{If } |q| < q_t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

선속도 $v_T(d, \theta)$ 는 식(1)을 바탕으로 다음과 같이 계산된다.

$$v_T(d, \theta) = v(d)s(\theta), \quad (2)$$

$$v(d) = \begin{cases} (v_{cruise} - v_{goal})g(d - d_i, d_i) + v_{goal} & \text{if } d < d_i \\ v_{cruise} & \text{otherwise} \end{cases}$$

$$s(\theta) = g(\theta, \theta_i^v), \quad 0 < \theta_i^v < \pi$$

여기서 $v_{cruise}, v_{goal}, d, d_i, \theta, \theta_i^v$ 는 각각 크루징 속도, 목표지점 도달 시의 목표 속도, 목표지점과의 거리, 목표지점 도달 시의 속도로 변경하기 위한 임계거리, 타겟과 로봇 간의 각도, 속도 조절을 위한 임계각도 이다.

각속도 $w_T(\theta)$ 는 다음과 같이 계산된다.

$$w(\theta) = \begin{cases} w_{cruise}(1 - g(\theta, \theta_i^v)) & \text{if } 0 \leq \theta \leq \theta_i^v \\ w_{cruise}(g(\theta, \theta_i^v) - 1) & \text{if } -\theta_i^v < \theta < 0 \\ w_{cruise} \text{sgn}(\theta) & \text{otherwise} \end{cases} \quad (3)$$

여기서 θ_i^v 는 임계 각도로서 θ 의 크기가 임계 각도보다 클 경우 θ 의 부호에 w_{cruise} 를 곱한 값을 각속도로 하여 빠른 각속도로 로봇이 비장애물 타겟 위치로 향하도록 회전하며, θ 가 0에 가까워질수록 각속도가 0에 수렴함으로써 로봇의 전진방향을 비장애물 타겟 위치 방향으로 고정시키도록 한다.

네 번째 단계에서는 로봇의 현재 속도와 최대 허용 속도 및 가속도를 고려한 속도 윈도우 영역을 생성한다. 속도 윈도우 영역은 각속도-선속도 좌표계에서 로봇의 현재 각속도와 선속도를 중심으로 하며, 로봇의 최대 허용 속도, 최대 허용 가속도, 제어 주기 시간 및 차동 구동형 이동로봇의 선속도와 각속도의 허용 범위를 고려하여 속도 윈도우 영역을 생성한다. 속도 윈도우 영역 내에 존재하는 가속도와 속도 쌍을 선택함으로써 로봇의 속도와 가속도에 대한 제약을 고려할 수 있다.

다섯 번째 단계에서는 속도 윈도우 영역과 비장애물 타겟 속도를 고려하여 윈도우 타겟 속도를 선택한다. 앞서 구한 비장애물 타겟 속도가 속도 윈도우 영역 내에 있다면 해당 각속도와 선속도 쌍을 윈도우 타겟 속도로 선택한다. 그렇지 않을 경우 속도 윈도우 영역 내에서 비장애물 타겟 속도와 가장 가까운 속도를 윈도우 타겟 속도로 선택한다.

여섯 번째 단계에서는 장애물을 로봇의 반지름만큼 확장한 영역(장애물 확장영역)과 비장애물 영역 간의 겹침으로 발생하는 장애물 확장영역의 각도 범위를 계산한다. 이를 위하여 라이다와 같은 거리센서를 사용하여 비장애물 영역과 장애물 확장영역이 겹치는 공간의 시작 각도와 끝 각도를 측정하여 장애물의 각도 범위를 계산한다. 따라서 비장애물 영역에서 장애물 확장영역의 시작 각도와 끝 각도 사이에 존재하는 호(장애물 호)와 만나는 로봇의 이동경로는 장애물 충돌이 예상되며, 이러한 이동경로는 로봇의 속도명령에 의해 결정된다. 장애물 시작 각도와 끝 각도가 각각 α, β 이고 $-\pi/2 < \alpha < \beta < \pi/2$ 의 부등식이 성립된다면 각속도와 속도 좌표계에서 장애물과 충돌이 예상되는 장애물 충돌 속도영역은 다음과 같다.

$$\frac{R_o}{2\cos(\pi/2 - \alpha)} w \leq v \leq \frac{R_o}{2\cos(\pi/2 - \beta)} w \quad (4)$$

여기서 R_o 는 비장애물 영역의 반지름이며 $|a| < \epsilon$ (ϵ 은 0에 가까운 미소값)

인 경우 $w=0$, 즉 v 축과 $v = \frac{R_o}{2\cos(\pi/2 - \beta)} w$ 사이의 공간이 장애물 충돌 속도영역이 된다. 장애물 시작 각도와 끝 각도의 범위가 $-\pi/2 \sim \pi/2$ 인 이유는 로봇과 타겟 간의 각도 차이가 $\pi/2$ 보다 크거나 같을 경우 식(2)의 $s(\theta)$ 가 0이 되므로 제자리 회전을 하게 되며, 이 경우 로봇과 장애물 간의 충돌은 발생하지 않기 때문이다. 따라서 $-\pi/2 \sim \pi/2$ 의 범위가 아닌 장애물 시작 각도와 끝 각도는 무시한다.

일곱 번째 단계에서는 최종 제어 속도를 선택한다. 만약 윈도우 타겟 속도가 장애물 충돌 속도영역에 속하지 않을 경우 윈도우 타겟 속도를 최종 제어 속도로 선택하여 로봇 제어기에 전달한다. 만약 윈도우 타겟 속도가 장애물 충돌 속도영역에 속할 경우 속도 윈도우 영역과 장애물 확장영역의 차집합에 속한 영역 내에서 윈도우 타겟 속도와 가장 가까운 각속도와 선속도 쌍을 최종 제어 속도로 사용하여 로봇 제어기에 전달한다.

마지막 여덟 번째 단계에서는 로컬 플래닝을 수행한다. 만약 2의 비장애물 타겟위치가 장애물 확장영역 내에 존재하거나 글로벌 경로와 비장애물 영역과의 교점이 존재하지 않을 경우 (즉, 비장애물 타겟위치가 정의되지 않은 경우) 로컬 지도 내에서 경로 생성 알고리즘을 실행하여 코일랩의 테두리와 글로벌 경로 간의 교점을 목표지점, 로봇의 위치를 시작 지점으로 하는 로컬 경로를 생성한다.

III. 결론

본 논문에서는 로봇의 제약을 고려하지 않고 생성된 글로벌 경로에 대해 차동 구동형 이동로봇의 운동역학을 고려하여 장애물과 충돌 없이 효율적으로 주어진 글로벌 경로를 추종할 수 있는 속도 명령을 생성하는 알고리즘을 제안하였다. 해당 알고리즘은 차동 구동형 이동로봇의 자율주행 시 장애물 회피 알고리즘으로 활용 가능하다. 추후 해당 알고리즘을 차동 구동형 로봇에 구현하여 스마트 팩토리에서 무인 운반차(Automated Guided Vehicle, AGV)를 운용할 경우 작업자 또는 다른 무인 운반차(Automated Guided Vehicle, AGV)와의 충돌을 회피하여 안전하게 이동할 수 있는 시스템을 개발하고자 한다.

ACKNOWLEDGMENT

이 논문은 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2021R111A3050100). 또한 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터)의 연구결과로 개발한 결과물임(IITP-2022-2020-0-01612).

참 고 문 헌

- [1] D. Fox, W. Burgard, S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics and Automation Magazine, pp. 23-33, 1997.
- [2] C. Rösmann, F. Hoffmann and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," Robotics and Autonomous Systems, Vol. 88, pp. 142-153, 2017.
- [3] Ulrich, I., Borenstein, J., "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots," Proceedings of International Conference on Robotics and Automation IEEE, Leuven Belgium, 1572-1577, 1998.
- [4] Ulrich, I., Borenstein, J., "VFH*: Local Obstacle Avoidance with Look-Ahead Verification," IEEE International Conference on Robotics and Automation, pp. 2505-2511, 2000.