

Lab : Automatic Emergency Braking (AEB)

Hyeong-jun Kim, Assistant Professor

ASMR Lab. / Department of Future Automotive Engineering

Gyeongsang National University

실습 목표:

- 고속 주행 시 충돌을 방지하는 Safety Node 개발
- ROS 2 메시지 활용: LaserScan, Odometry
- 핵심 알고리즘: Instantaneous Time to Collision (iTTC) 구현

왜 중요한가?

- 자율주행 차량의 가장 기초적이고 중요한 안전 장치 (Safety Critical System)
- 주행 알고리즘이 실패하더라도 독립적으로 작동하여 차량을 보호해야 함

1. LaserScan (/scan)

- LiDAR 센서 데이터핵심 필드: ranges (장애물까지의 거리 배열)
- 방사형으로 정렬된 거리 데이터를 통해 r (Range) 획득

2. Odometry (/ego_racecar/odom)

- 차량의 위치 및 속도 정보
- 핵심 필드: twist.twist.linear.x (차량의 전진 속도 v_x)

3. AckermannDriveStamped (/drive)

- 차량 제어 명령
- 제동 방법: speed 필드를 0.0으로 설정하여 발행

정의: 차량이 현재의 속도와 진행 방향을 유지한다고 가정할 때, 장애물과 충돌하기까지 남은 시간

기본 공식: $iTTC = \frac{r}{\{-\dot{r}\}_+}$

- r : 순간 거리 측정값 (Range)
- \dot{r} : 거리 변화율 (Range Rate, 상대 속도)

연산자 $\{\}_+$ 의 의미:

- $\max(x, 0)$
- 차량이 장애물로부터 멀어지거나 ($\dot{r} > 0$), 정지해 있을 때 iTTC가 무한대가 되도록 처리 (충돌 위험 없음)

방법: 속도 벡터 투영 (Projection)

- 차량의 전진 속도(v_x)를 각 LiDAR 빔의 각도(θ)로 투영하여 계산
- 공식: $\dot{r} = v_x \cos(\theta)$

부호의 의미:

- 음수 (-): 장애물에 접근 중 (거리가 줄어듦) → 위험
- 양수 (+): 장애물에서 멀어지는 중 (거리가 늘어남) → 안전

주의사항:

- 각도(θ)는 LaserScan 메시지의 angle_min, angle_increment를 이용해 계산해야 함

1. Subscribe:

- /scan: 장애물 거리(r) 및 각도(θ) 데이터 수신
- /ego_racecar/odom: 차량 속도(v_x) 수신

2. Process (Loop):

- LiDAR의 모든 빔에 대해 반복
- 각 빔의 $\dot{r} = v_x \cos(\theta)$ 계산
- iTTC = $\frac{r}{\max(-\dot{r}, 0)}$ 계산
- 배열 내 최소 iTTC (Min iTTC) 추출

3. Act (Publish):

- Condition: Min iTTC < Threshold (임계값)
- Action: /drive 토퍼으로 speed = 0.0 발행 (Emergency Brake)

테스트 환경:

- sim.yaml에서 kb_teleop을 True로 설정
- teleop_twist_keyboard 노드를 실행하여 키보드로 차량 조작

검증 시나리오:

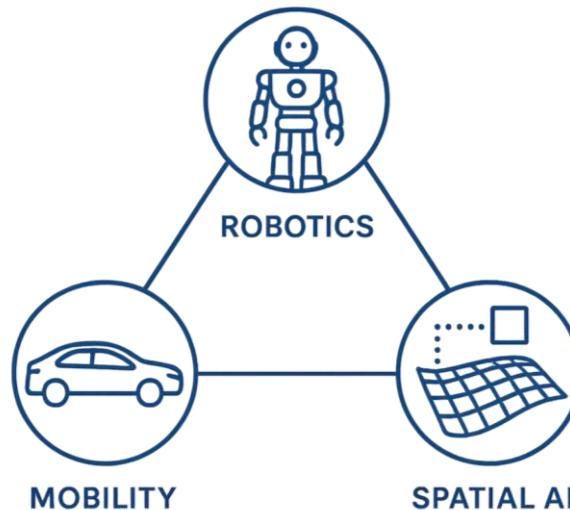
- 복도 주행: 직진 시 벽을 인식해도 멈추지 않아야 함 (False Positive 방지)
- 벽 충돌: 벽을 향해 전속력으로 달릴 때 충돌 전 정지해야 함

데이터 처리 팁:

- inf (무한대) 또는 nan (숫자 아님) 값 예외 처리 필수
- 적절한 Threshold 값 찾기 (너무 크면 일찍 멈추고, 작으면 충돌함)

Thank you for your attention

Questions and feedback are welcome.



ASMR Lab.
Autonomous Systems and Multimodal Reasoning Lab.

hj.kim@gnu.ac.kr