

시스템 설계 (SDD)

프로젝트 제목 : 미아 탐지 로봇

조원 : 김교회, 박상욱, 이건도

날짜 : 2025.01.18

1. 개요 (Introduction)

- 목적: AI 기반 자율 이동 로봇(AMR)을 사용하여 mapping 완료된 구역 내 두 지점(A, B)를 왕복하며 자율 순찰, 목표 객체 탐지 및 추적 후 사용자에게 신호 송신을 제공하는 시스템이며, PC AMR의 하드웨어와 소프트웨어를 필요로 하고
- 있습니다.

2. 시스템 개요 (System Overview)

- 시스템 이름: 미아 탐지 로봇 시스템
- 기능 설명:
 - AMR이 지정된 경로를 이동하며 USB 카메라로 이미지를 수집합니다.
 - AI 모델이 영상을 실시간 분석하여 미아를 탐지합니다.
 - 탐지된 정보를 기반으로 상황을 기록하고, 신고 및 알림을 수행합니다.
- 사용자 시나리오:
 - AMR은 지정된 위치에서 출발하여 목표 지점까지 반복하며 순찰합니다.
 - 이동 중 미아를 감지하면 사용자에게 경고를 발생시키고 외부 신호를 보냅니다.

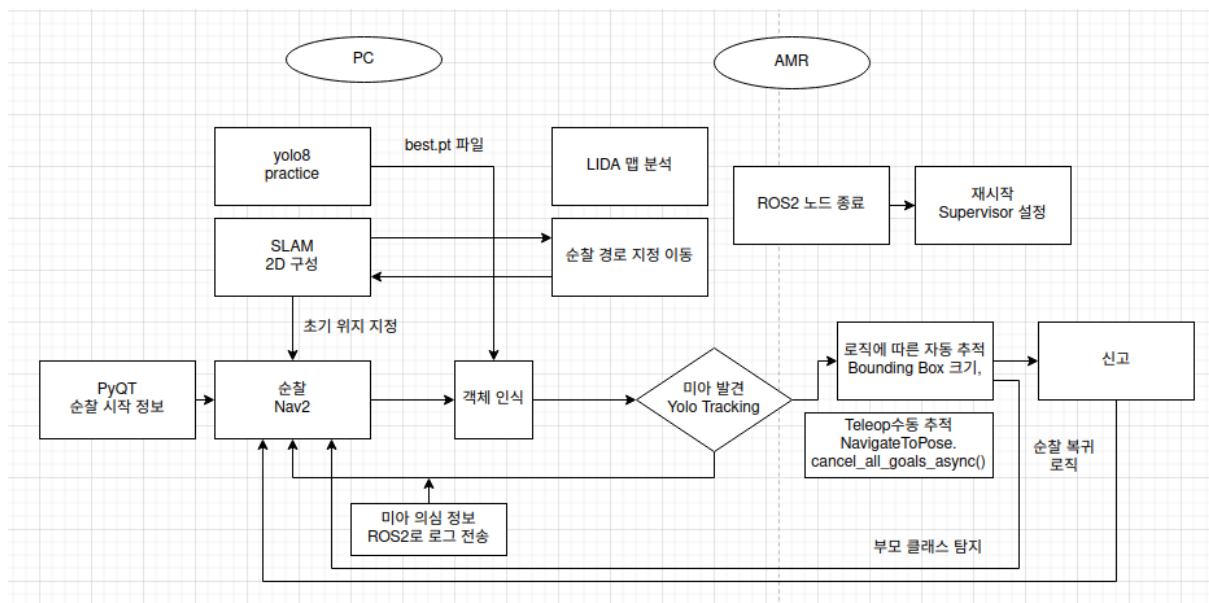
3. 시스템 구성 요소 (System Components)

- 하드웨어:
 - PC:
 - Ubuntu 22.04, USB Camera, Wi-Fi통신
 - AMR(Turtlebot3):
 - Jetson-Orin, Ubuntu 22.04, USB Camera,
 - 배터리: openCV, Jetson-Orin-nano

- 소프트웨어:
 - ROS2 humble: 통신
 - Navigation2 : ROS 2 기반 경로 계획 및 내비게이션 라이브러리
 - OpenCV: 객체 감지 및 추적을 위한 영상 처리 라이브러리, ROS 2 노드와 쉽게 통합 가능
 - YOLOv8 : 실시간 객체 감지, Detection, Tracking을 위한 고성능 딥러닝 모델
 - Gazebo : 테스트 및 검증용
 - Rviz : 실시간 데이터 및 객체 추적 결과 시각화, (센서 데이터와 추적 객체를 함께 표현 가능)
 - PyQt : PC 사용자 인터페이스에서 데이터 시각화 소프트웨어
- 통신 방식:
 - Wi-Fi를 통한 ROS2 메시지 송수신

4. 아키텍처 설계 (Architecture Design)

- 시스템 아키텍처 다이어그램:



5. 상세 설계 (Detailed Design)

기능 설계

1. 네비게이션:
 1. pyQT로 “순찰 개시” popup 유저 GUI

2. Capogragh(구글 자체 개발 SLAM,mapping), 로 mapping 된 특정 범위 내에서 pose를 지정할 수 있다.
3. 처음 goal send
4. ROS2: 이때 AMR은 Nav2와 통신하며 Infinital Pose Data를 설정한 후(map) 액션 클라이언트로 포즈 전송, While loop 를 이용해 계속적으로 목표좌표 전송 가능
5. maping된 장애물을 회피하며 순찰

2. 목표 객체 인식:

- a. 순찰 중 AMR 카메라에 목표 객체 탐지
- b. YOLOv8 알고리즘으로 카메라 데이터에서 객체 인식
- c. 감지 신호, goal pose 취소 메시지를 Nav2 액션 서버에 전송해 네비게이션 중단
- d. bool 메시지를 통해 Teleop 전환 신호 전송
- e. AMR이 감지된 객체의 바운딩 박스 좌표 및 중심점을 계산, OpenCV를 사용하여 바운딩 박스를 이미지에 그림, cv_bridge를 통해 OpenCV 이미지를 ROS 메시지로 변환. img topic을 ROS 2 메시지나 소켓을 통해 PC에서 topic 을 subscribe해 수신

3. 목표 객체 특정화 및 트래킹: 목표 객체 **yolo tracking**

- a. YOLO Model에서 제공하는 kid 클래스만 필터링
- b. ROS2를 사용하여 추적 결과를 퍼블리싱하고 PC노드가 이를 구독
- c. OpenCV로 이를 시각화(바운딩 박스와 ID, 클래스 이름)

4. AMR의 객체 **follow**

- a. 목표 객체의 bounding box 크기와 centerpoint 계산 거리계산-박스의 크기 방향-박스의 center Point
- b. ROS2 AMR 제어 AMR 속도명령 퍼블리셔, (Twist, 'cmd_vel' 대충 10) 카메라 입력 img subscribe,
- c. twist 명령 사용하여 x축 중심 위치 에 따라 로봇 회전 angular speed 약 0.01 (테스트 후 조정 가능) 바운딩 박스의 width 에 따라 linear speed 약 0.02 (테스트 후 조정 가능)
- d. YOLO Tracking과 AMR 제어 노드를 동시에 실행하는 Launch 파일을 description하여 ROS2 시스템 통합

※ 성능에 따라 Teleop수동 추적

5. 추적 중단

- a. AMR 카메라에 목표 해제 클래스(parent) 감지 상태
- b. parent centerpoint와 kid centerpoint 거리 임계값 이상 가까이 있는 상태로 10초 이상 유지되면 보호자 감지 상황(Patrolling)으로 전환
- c. 10초가 지나 추적중단 상황 후 추적 중단 신호를 ROS2 topic 메시지로 “추적 완료, 순찰 복귀” 송신, PC subscribe
- d. AMR에서 Teleop 노드와 Navigation 노드 간 메시지 통신을 통해 Topic메시지 전달
- e. AMR에서 /set_mode를 통해 Navigation 상태로 전환.

6. 순찰 로직 복귀

- a. Nav2 bringup (AMCL 노드 실행) 맵을 로드하고 RViz2에서 시각화
- b. RViz2의 "2D Pose Estimate" 도구를 사용해 초기 위치와 방향 설정, AMCL로 로봇의 위치 업데이트
- c. AMCL로 AMR 현재 위치 확인 후 두 목표지점 좌표를 list로 정의, 두 목표좌표 중 현재 위치에서 가장 가까운 지점을 계산하여 복귀 지점을 설정, 목표가 수락되면 상태를 확인하고 처리.
- d. NavigateToPose 액션 클라이언트를 사용해 목표 좌표로 이동
- e. 목표에 도달하면 다음 목표를 콜백(result_callback)에서 다음 목표를 자동으로 설정
- f. RViz2에서 로봇 위치 시각화, 로봇의 위치와 방향이 올바르게 설정되었는지 확인 "2D Pose Estimate" 도구를 사용하여 수정

● 미아 판별 기준

- g. 한 화면에 미아 클래스만 발견 되면 미아로 의심
- h. 미아로 의심하는 도중 부모 클래스 5초 이상 감지 시 미아 의심 해제.
- i. 10초 이상 감지 시 "미아", 5초~10초는 "미아 의심"으로 표시.
- j. 여러 명의 미아 발견 시 가장 먼저 감지된 아이의 번호를 추적

● yolov8 detective 학습

1. 다각도의 목표객체가 포함된 1000장의 데이터셋 확보
2. Data를 labelImg 를 사용해 라벨링 Class 1.kid 2.dummy 3.parent
3. epoch: 40로 YOLOv8 python3 API 학습.
4. 학습이 완료된 best.pt 모델을 저장
5. Confidence threshold : 0.8, Accuracy : 0.9

- 데이터 처리 흐름:
 - k. AMR이 정해진 두 지점을 왕복 시작
 - l. USB 카메라로 이미지 수집 후 YOLOv8 모델에 전송.
 - m. 탐지 결과에 따라 미아 판별 및 경고 메시지 전송.
 - n. 부모와 동반 시 목적지로 재이동.

Sensor Node --> Object Detection Node --> Object Tracking Node --> Path Planning Node

6. 데이터 설계 (Data Design)

- 입력 데이터: USB 카메라를 통해 수집된 이미지 데이터
 - 출력 데이터: YOLOv8 모델의 탐지 결과 및 탐지된 객체 클래스
 - 로그: 탐지 시각, 탐지 해제 시각, 탐지 위치 기록, 누적 미아 의심수, 누적 미아수, 객체 accuracy, 배터리 상태
-

7. 요구 사항 (Requirements)

- 기능 요구 사항
 - AMR이 장애물을 회피하며 지정된 경로를 자율주행 및 왕복.
 - AMR의 실시간 미아 객체 탐지 후 신고 신호 발행
 - ROS 2 시스템과의 통합 - 추적 정보 퍼블리시
 - 결과 시각화 - OpenCV로 실시간 감지 및 추적 정보 표시.
 - 데이터 저장 - 추적 데이터(객체 ID, 좌표, 클래스 등)를 로컬 데이터베이스에 저장.
 - 성능 요구 사항:
 - 객체 추적- 정확성 :YOLOv8의 Confidence Threshold : 0.8, accuracy : 0.9 설정, 안정성: IOU Threshold: 최소 0.5 이상.
 - 객체 탐지-실시간 처리 최소 15 FPS 이상 유지(GPU)
-

8. 테스트 계획 (Testing Plan)

테스트 모니터링(RViz2)

- 기능 테스트
 - 출발점에서 목표지점까지 순찰 경로 탐색 기능 테스트
 - 출발,목표 지점 토폭 전달 여부 확인과 벽에 부딪히지 않는지 확인

- 자율주행 중 장애물 발견 및 회피 테스트
 - 장애물 발견 시 감지 후 방향전환하는지 확인
- 다양한 조건 미아 객체 탐지 여부 확인
 - Edgecase 데이터에서 객체 탐지 확인
- 각 과정에서 두 하드웨어 사이 ROS2통신 테스트
 - Publish & subscribe 확인
- 목표객체 tracking 시
 - 번호가 매겨진 객체 추적
- 실시간 로그 확인
 - 실제 클래스 판별 시 시간 및 위치 확인
- 성능 테스트
 - 자율주행 목표 정확도
 - AMCL 기반 위치 추정을 활용하여 로봇이 goal 도달 정확도 확인
 - 각 goal까지 이동하는 데 걸린 시간 측정
 - 데이터 수집 속도 성능
 - ROS2 토픽 에서 데이터 초당 수집된 데이터 메시지 수 측정
 - YOLOv8 모델의 실시간 탐지 속도
 - GPU가 장착된 시스템에서 FPS 테스트
 - YOLOv8 모델의 탐지 정확도 평가(누락,오탐 횟수 Accuracy측정)
 - 라벨 데이터와 예측 데이터 비교

9. 예외 처리

- 네트워크 손실: ROS2 네트워크 상태를 모니터링하는 노드를 구현하여 노드 간 연결 상태를 확인, 네트워크 손실 시 네트워크 스크립트를 사용하여 재연결을 시도하고, ROS2 노드를 재등록.
- 방화벽 오류 : ROS2의 통신 포트 범위를 수동으로 설정하여 방화벽 문제를 완화.(ROS2 DDS 포트를 방화벽에서 허용하고, 포트 충돌 방지)
- 하드웨어 손상 **PC,AMR** : 정기적으로 카메라 센서 및 하드웨어 상태를 주기적으로 점검 및 최적화
- 배터리 부족 :
 - 추적 수행 도중: AMR은 추적 완료 까지 완수한 후 좌표 신호 전송 , Nav2에서 다시 Teleop 변환 요청 경고 신호 PC사용자에 송신
 - 순찰 중 : AMR은 자동으로 초기 충전 위치로 복귀
- 가상환경 연결 오류 : 가상환경 활성화 실패를 탐지하고 복구, 패키지 설치 오류 시 재시작