

# ROS2 를 활용한 AI & ICT 기반 자율주행 공기청정기 연구

오현준<sup>1</sup>, 하은지<sup>2</sup>, 김기흥<sup>2</sup>, 이경용<sup>3</sup><sup>1</sup> 전북대학교 기계공학과<sup>2</sup> 동아대학교 AI 학과<sup>3</sup> 부경대학교 컴퓨터공학과

ohj\_980918@jbnu.ac.kr, ejha2002@donga.ac.kr, 2143678@donga.ac.kr, ceo@apptools.co.kr

## A Study on AI & ICT-based Autonomous Air Purifier Using ROS2

Hyoun-Jun Oh<sup>1</sup>, Eun-ji Ha<sup>2</sup>, Gi-Heung Kim<sup>2</sup>, Kyung-Yong Lee<sup>3</sup><sup>1</sup>Dept. of Mechanical Engineering, Jeonbuk National University<sup>2</sup>Dept. of Artificial Intelligence, Dong-A University<sup>3</sup>Dept. of Computer Science and Engineering, Pukyong National University

### 요 약

공기질에 대한 인식이 제고됨에 따라 실내공기질 규제가 강화되고 있다. 따라서 기존 고정형 공기청정기를 대체할 효율적인 기술로 자율주행 공기청정기를 연구하였다. 또한 자율주행 공기청정기에 AI를 적용하여 오염도를 예측하는 방법으로 기존 자율주행 공기청정기와 차별성을 둔다.

### 1. 서론

최근 발생한 코로나 19 와 지속적인 대기오염은 공기질에 대한 인식을 제고하였다. 환경부에서도 이러한 인식에 발맞춰 공기질 개선을 위한 중장기 대책을 마련하고 있는 실정이다. 특히 어린이집, 학교, 요양원 같은 다중이용시설의 민감계층 관리시설은 더 높은 관리수준을 지켜야 한다고 명시되어 있다. 하지만 설비를 구축하기 위한 영세시설의 부담, 지자체 여건 등의 문제점이 존재한다. 따라서 기존의 방식보다 재정부담이 덜 하며, 효율성이 증가된 정화시스템의 개발이 필요하다.[1]

본 논문에서 소개하는 자율주행 공기청정기는 개발의 필요성에 부합한다. 같은 공간을 정화하는데 여러대의 고정형 공기청정기가 필요했다면, 자율주행 공기청정기는 한 대로 정화가 가능하다. 또한 누적된 데이터를 바탕으로 AI 지도학습을 진행한다. AI 지도학습의 결과데이터는 오염도가 높을 것으로 예측되는 구역의 체류시간을 길게 잡는다. 본 논문의 자율주행 공기청정기는 단순히 시계열 요소만이 아닌, 실내의 특정 위치의 오염도를 예측한다는 점에서 기존 연구와 차별점을 둔다.

자율주행 공기청정기에 대한 기존 연구는 대부분 ROS1 을 기반으로 이루어졌다. 하지만 ROS1 은 TCPROS(TCP/IP) 프로토콜 메시지 통신을 기반으로 정보를 주고받기 때문에 보안에 취약하다. 또한 핵심 코드의 개발은 중지된 채 2025 년 EOL(end-of-life)을 앞두고 있다. 이러한 이유로 ROS2 를 활용한 연구의 필요성이 있다. ROS2 는 DDS(Data Distribution System)의 DDSI-RTPS(Real Time Publish Subscribe)를 사용하여

산업용 표준을 추종하며, DDS-Security 도입으로 보안 측면에서도 이점을 가진다.[2]

### 2. 세부 설계

#### 2.1 개발환경

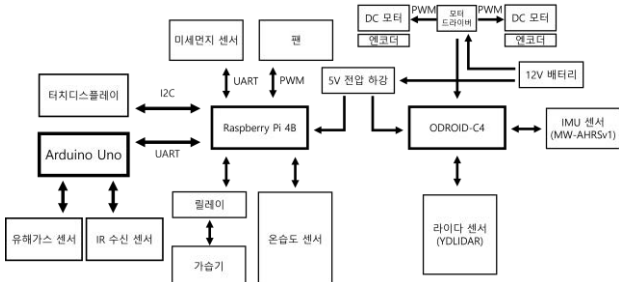
본 논문의 자율주행 공기청정기 개발환경은 다음과 같다.

용도	마스터 PC	데이터 수집	주행	아날로그 데이터 수집
기기	MSI Leopard GP76	Raspberry Pi 4 4GB	Odroid-C4	Arduino Uno
운영 체제	Ubuntu 20.04 LTS	Ubuntu 20.04 LTS	Ubuntu Mate 20.04	None
ROS2 버전	Galactic	Foxy	Foxy	None

<표 1> 자율주행 공기청정기의 개발환경

#### 2.2 하드웨어

본 논문의 자율주행 공기청정기의 하드웨어는 크게 세 개의 파트로 분류된다. 모든 외형은 직접 3D 모델링하였다. 하단부는 주행을 담당한다. 두 개의 엔코더 모터와 IMU 센서, Odroid-C4, 모터드라이버가 탑재되어 있다. 중단부는 정화를 담당한다. H13 등급의 헤파 필터와 원심형 팬이 장착되어 있다. 상단부는 환경데이터를 수집을 담당한다. 온도 센서, 미세먼지 센서, 유해가스 감지센서, 라이다 센서, 가습기, 터치형 디스플레이가 장착되어 있다. 하드웨어의 전체적인 구성도는 다음과 같다.



<그림 1> 자율주행 공기청정기의 하드웨어 구성도

2.3 통신

주행을 담당하는 Odroid-C4 와 센서 데이터 수집을 담당하는 Raspberry Pi 4 는 SSH 통신 프로토콜로 연결한다. 그리고 Raspberry Pi 4 와 아날로그 데이터를 받기 위한 Arduino Uno 는 Serial 통신 중 UART 통신방식을 이용한다.

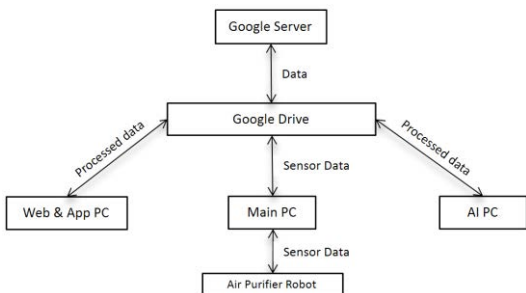
센서로 수집한 환경데이터는 ROS2 를 이용하여 유기적으로 통신한다. ROS2 는 DDS 통신방식을 이용한다. 각 센서는 수집한 정보를 서로 다른 노드에 Publish 한다. 또한 기구 작동을 담당하는 노드는 센서 데이터를 Subscribe 하여 작동에 참고한다. 사용자 정의 Interface 를 활용하여 필요한 데이터를 원하는 포맷으로 가공하여 전송한다.

ROS2 에는 크게 Topic, Service, Action 세 가지 통신방식이 존재한다. 본 논문의 자율주행 공기청정기는 센서데이터를 Topic 방식을 이용하여 노드를 구성한다. RQT 로 출력한 노드 구성도는 다음과 같다.



<그림 2> 자율주행 공기청정기의 ROS2 노드 구성도

Google Server 의 간접 활용 도구인 Google Drive 와 여러 PC 사이의 서버 통신 방법은 다음과 같다.



<그림 3> 서버 통신 방법

2.4 인공지능

시간별 공간의 오염도에 따른 체류시간을 예측하는데 목적이 있다. 오염도가 높은 지역의 체류시간을 길게 하여 주행하도록 한다. 예측모델 적용 전 정확도는 약 36%이고, 목표한 정확도는 70%이상이다.

인공지능 설계는 다음과 같다. 환경 데이터 가공 → LSTM 기반 모델 생성 → 학습 → 예측 결과

2.4.1 Environmental data Processing

자율주행 공기청정기가 수집한 환경 데이터는 CSV 포맷으로 가공되고 AI 서버에 1 차적으로 받아온다.

이후, 인공지능 모델이 학습하기에 적합한 다차원 배열로 재가공 한다.

No	year	month	day	hour	local	temp	humidity	metter
1	2022	8	2	9	1	22.5	38.3	3
2	2022	8	2	9	2	22.8	37.2	3
3	2022	8	2	9	3	23.5	35.8	3

<사진 1> 데이터의 가공 전 사진

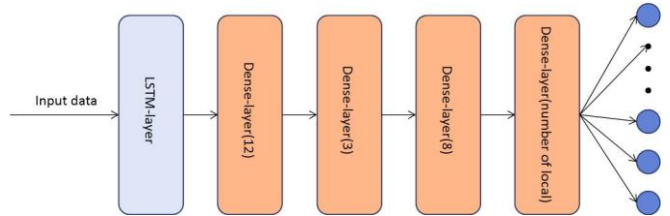
```
(182, 6, 3) (182, 6) (91, 6, 3) (91, 6)
[[[0.00988002 0.94809675 0.33333334]
 [0.01058574 0.94809675 0.33333334]
 [0.0091743 0.93079567 0.66666667 ]
 [0.01270288 0.8581314 0.33333334]
 [0.00776288 0.87889266 0. ]
 [0.00635144 0.9411764 0.33333334]]]
```

<사진 2> 데이터의 가공 후 사진

(182, 6, 3) (182, 6)와 (91, 6, 3) (91, 6) 각각 훈련 데이터와 테스트 데이터의 크기이다.

2.4.2 LSTM neural network-based prediction model

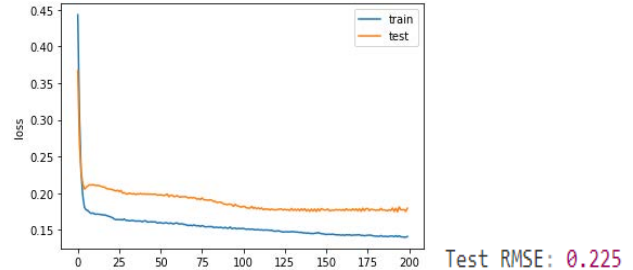
모델은 Keras 의 LSTM 과 Dense 를 이용하여 다음과 같이 구성하였다.



<그림 4> 모델 구성[3]

2.4.3 Training

아래는 시계열 환경데이터를 학습한 결과의 평균제곱근오차(RMSE, Root Mean Square Error)이다. 목표한 정확도인 70% 이상의 성능을 보여준다.



<사진 3> 학습 결과 및 성능(x: epoch, y: MAE - mean absolute error)

2.4.4 Result

모델의 예측 결과에서 Softmax 를 통해 자율주행 공기청정기의 주행 스케줄로 변환한다. 'local'은 실내 구역별 번호이고 'stay\_time' 구역별 정화하는 시간이다.

A	B
local	stay_time
1	5
2	9
3	15
4	15
5	8
6	8
1	8
2	9

<사진 4>예측한 주행스케줄

2.5 서버

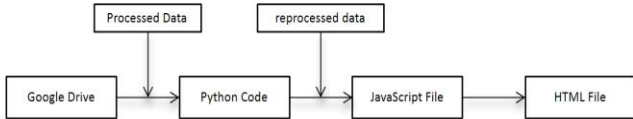
Google Server 를 활용한 Cloud Service 인 Google Drive 를 자율주행 공기청정기와 연동된 마스터 PC 에 연결한다. 그리고 연결되지 않은 남은 PC 인 AI & Web PC 를 마스터 PC 와 연결된 Google Drive 에 연결한다.

이때 Google Drive 로 연결된 각 PC 에는 연결과 동시에 자동으로 Local Disk 형태로 Google Drive 가 생

성된다. 생성된 Disk 에는 주로 ROS2 를 통해 실시간으로 수집되는 Data 와 AI Server 를 통해 가공된 Data 가 저장되며, 후에 Data 들은 각 팀원들의 Google Drive 에 실시간으로 자동 Upload 되어 공유된다.

2.6 웹

Google Drive 에 Upload 된 Processed Data 를 Python 을 활용하여 전처리한다. 후에 해당 Data 를 HTML, CSS , Javascript 를 기반으로 DashBoard 로 재구성한다. 이때 제작된 DashBoard 에서는 실시간으로 수집된 Data 의 통계자료를 다양한 다이어그램 형태로 시각화 해준다.



<그림 5> 작업 구성도

3. 주요 기능 구현

3.1 데이터 수집과 상호작용

수집한 데이터는 각종 장치와 상호작용하여 구동한다. 온습도 센서로 수집한 데이터는 가습기 작동에 관여한다. 실내 온도에 따른 적정 습도 범위 미만이면, 릴레이 스위치가 작동하여 가습기를 가동한다. 적정 습도 범위를 벗어나면 가습기 작동을 중단한다. 미세먼지 센서는 2.5pm, 10pm 입자를 데이터로 반환한다. 환경부에서 발표한 미세먼지 기준을 1~10 단계로 등분하고 2.5pm, 10pm 입자 수치를 적절한 비율로 조합하여 오염도 수치를 계산한다. 이 오염도 수치는 팬의 속도를 조절하는데 사용된다. 만약 유해가스가 감지되면 그 즉시 팬의 가동 속도를 높여 정화한다.

3.3 SLAM 과 자율주행

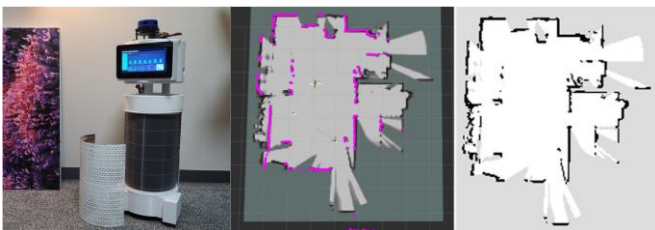
자율주행은 ROS2 패키지인 Cartographer 와 NAV2 로 구현하였다.

SLAM 은 Cartographer 를 이용하여 진행한다. SLAM 으로 나온 지도에 약 2 평 마다 구역좌표를 설정한다. 이 좌표는 CSV 포맷으로 저장된다.

자율주행은 NAV2 를 이용한다. NAV2 의 Nav2SimpleCommander 패키지를 활용하였다. 좌표 CSV 파일을 읽어 구역 순서대로 주행을 진행한다.

4. 구현결과

자율주행 공기청정기는 새로운 공간에 대해서 SLAM 이 필요하다.



<사진 5>자율주행 공기청정기(좌) Rviz Cartographer SLAM(중) SLAM 결과물 Map.pgm(우)

RVIZ 상에서 보이는 격자의 넓이는 1 제곱미터이다. 테스트에서는 2 제곱미터를 하나의 공간으로 가정하였다.

초기위치, 경로점의 Orientation 과 Position, 정화 체류시간 예측데이터의 CSV 파일을 읽는다. 위치 좌표의 Orientation 은 Quaternion 방식이 사용된다.

Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard	Standard
1 X	y	yaw	1 X	y	w	1	Local	stay_time	hour
2 1.0	0.0	180.0	2 0.0	0.0	1.0	2 1	12	9	
3 -1.0	0.0	1.0	3 2	13	9				
4 -2.0	0.0	1.0	4 3	11	9				
5 -2.0	-1.0	1.0	5 4	9	9				
6 -1.0	-1.0	1.0	6 5	7	9				
7 -1.0	0.0	1.0	7 6	8	9				

<사진 6> 초기 좌표(좌) 경로 좌표(중) 체류시간 예측 데이터(우)

각 구역에 도착하면 예측한 체류시간 만큼 대기한다. 오염도가 높을 것으로 예측되는 지역에 대해서 더 오래 체류하고 정화함과 동시에 환경 데이터를 수집한다.



<사진 7> 1번 공간 주행 완료(좌) 1번공간 정화중(우) 서버에 누적된 데이터는 일주일 단위로 관제대시보드에 업데이트 되어 화면에 표시된다.



<사진 8> 일주일간의 각 일별 데이터

5. 향후 연구 방향

향후 NAV2 를 보완하여 자율주행 상태에서 SLAM 을 진행하도록 하는 연구가 필요하다. 또한 비전을 이용하여 낮은 장애물을 인지하고 정지 혹은 회피하여 사각지대를 해소할 필요성이 있다.

※ 본 프로젝트는 과학기술정보통신부 정보통신창의 인재양성사업의 지원을 통해 수행한 ICT 멘토링 프로젝트 결과물입니다.

참고문헌

[1] 관계부처합동 발행, “실내공기질 관리 기본계획 [2020~2024]”, 환경부, 2020  
 [2]표윤석, 임태훈 지음, “ROS2 로 시작하는 로봇 프로그래밍”, 경기도 부천시, 루비페이퍼, 2021  
 [3] Jason Brownlee "Multivariate Time Series Forecasting with LSTMs in Keras"