

ROS 2 기반 모바일 로봇 디지털 트윈 환경 구축

최성식*¹, 최승환*², 김창현*³, 권효은*⁴, 이수웅*⁵

Development of a ROS 2-based Digital Twin Environment for Mobile Robots

Seong-Sig Choi*¹, Seung-Hwan Choi*², Chang-Hyun Kim*³, Hyoeun Kwon*⁴, and Suwoong Lee*⁵

이 연구는 2025년도 산업통상자원부 및 한국산업기술평가관리원(KEIT) 연구비 지원에 의한 연구(‘RS-2024-00422721’)입니다. 또한, 이 연구는 한국생산기술연구원 기관주요사업 “제품 제조현장 작업파트너 로봇 기술 개발 (kitech EO-25-0005)”의 지원으로 수행한 연구입니다.

요 약

최근 디지털 트윈은 로봇 분야에서 실제와 가상 환경의 동기화를 통해 안전성과 효율성을 높이는 방안으로 주목받고 있다. 본 논문에서는 ROS 2를 기반으로 모바일 로봇과 가상 환경을 연동하여 디지털 트윈을 구현하였다. 제어 및 센서 데이터를 ROS 2 토픽으로 공유하도록 시스템을 구성하였으며, 실험에서는 직선 주행과 회전 동작을 비교하여 실제와 가상 로봇의 유사성을 검증하였다. 또한 각 로봇에 장착된 LiDAR 데이터를 동일한 메시지 구조로 발행하고 시각화하여, 센서 데이터가 네트워크상에서 호환 가능함을 확인하였다. 본 연구는 실제와 가상 로봇 간 양방향 제어 및 센서 연동이 가능한 디지털 트윈 구현 사례로, 향후 자율주행 및 복합 센서 융합 연구의 기반으로 활용될 수 있을 것으로 기대된다.

Abstract

Recently, digital twins have gained attention in robotics as a way to enhance safety and efficiency through synchronization between real and virtual environments. In this study, a digital twin environment was implemented by integrating a mobile robot with a virtual platform using ROS 2. Control and sensor data were shared through ROS 2 topics, and experiments comparing straight and rotational motions demonstrated the similarity of real and virtual robot behaviors. LiDAR data were published with a unified message structure and visualized in ROS 2, confirming their interoperability within the network. This study demonstrates a bidirectional digital twin framework that enables synchronized control and sensor interaction between real and virtual robots, which can serve as a foundation for future research on autonomous navigation and multi-sensor fusion.

Keywords

ROS 2, mobile robot, isaac sim, digital twin, lidar

* 한국생산기술연구원 연구원(*⁵ 교신저자)
- ORCID¹: <https://orcid.org/0009-0000-4153-7806>
- ORCID²: <https://orcid.org/0000-0003-1898-9884>
- ORCID³: <https://orcid.org/0000-0001-5805-7120>
- ORCID⁴: <https://orcid.org/0009-0006-6226-146X>
- ORCID⁵: <https://orcid.org/0000-0001-6547-9225>

• Received: Sep. 24, 2025, Revised: Oct. 30, 2025, Accepted: Nov. 02, 2025
• Corresponding Author: Suwoong Lee
Advanced Mobility System Group, Daegyeong Technology Application
Division, Korea Institute of Industrial Technology, 320 Techno sunhwan-ro,
Yuga-eup, Dalseong-gun, Daegu 42994 Republic of Korea
Tel.: +82-53-580-0161, Email: lee@kitech.re.kr

I. 서 론

최근 인공지능(AI), 센서 기술, 네트워크 통신의 발달과 함께 실제 물리적 시스템을 가상 공간에 그대로 구현하는 디지털 트윈(Digital twin) 기술이 다양한 분야에서 활용되고 있다. 디지털 트윈은 실제 시스템의 동작 상태를 가상 모델과 실시간으로 동기화하여 분석·예측·제어에 활용할 수 있다는 점에서 주목받고 있으며, 스마트 제조, 물류 자동화, 자율주행, 에너지 관리, 농업 등 다양한 산업 분야에서 도입이 확대되고 있다[1].

이동로봇 분야에서도 디지털 트윈의 활용은 점차 증가하고 있다. 실제 로봇을 활용한 실험은 환경 구축에 많은 비용과 시간이 소요될 뿐 아니라 반복 실험 과정에서 장비 손상이나 안전사고의 위험이 존재한다. 반면, 시뮬레이션만을 이용할 경우 실제 환경의 불확실성과 제약 조건을 충분히 반영하지 못해 현실 적용 가능성에 한계가 있다. 따라서 실제 로봇과 시뮬레이터를 연동하여 동일한 제어 및 센서 데이터 구조를 공유하는 디지털 트윈 구현이 필요하다. 이러한 접근은 실험의 안전성과 효율성을 동시에 확보하면서도 현실 적용 가능성을 높일 수 있는 장점을 갖는다[2].

ROS(Robot Operating System)는 로봇 분야에서 가장 널리 사용되는 미들웨어로, 특히 ROS 2는 분산 시스템 지원, 실시간성 강화, 보안성 개선 등을 통해 다양한 로봇 응용에 활용되고 있다[3]. 한편, NVIDIA Isaac Sim은 Omniverse 기반의 고성능 시뮬레이터로, ROS 2와의 연동을 지원하여 실제 로봇과 가상 로봇을 통합할 수 있는 환경을 제공한다.

선행 연구로 ROS와 NVIDIA Omniverse를 연동하여 TurtleBot3의 실환경 맵을 가상환경에 동일하게 적용하고, ROS Bridge를 통해 실환경 로봇과 가상 로봇의 동시 제어를 구현하였다[4]. 해당 연구는 ROS 1 기반의 단일 센서(SLAM 중심) 실험에 초점을 두었으나, ROS 2 환경에서의 분산 통신, 다중 센서 데이터 연동, 정합성 평가 등은 다루지 않았다.

이에 본 연구에서는 ROS 2 기반 모바일 로봇과 NVIDIA Isaac Sim을 연동하여 실제와 가상 로봇 간 제어 명령 및 센서 데이터를 실시간으로 동기화하고, LiDAR 데이터를 포함한 센서 레벨의 일관성 검

증이 가능한 디지털 트윈 환경을 구현하였다. 또한 ROS 2의 DDS 통신과 Chrony 시간 동기화를 활용하여 제어 명령과 센서 데이터의 정합성을 실험적으로 검증하였다. 이를 통해 실제-가상 연동 구조의 확장성과 정밀도를 평가하였다[5][6].

본 논문은 다음과 같이 구성된다. 2장에서는 전체 시스템 구성을 설명하고, 3장에서는 구현 방법을 구체적으로 기술한다. 4장에서는 실험을 통해 제안한 시스템의 동작과 센서 데이터 발행 결과를 검증하며, 5장에서는 결론과 향후 연구 방향을 제시한다.

II. 시스템 구성

2.1 모바일 로봇 시스템

본 연구에서 사용된 모바일 로봇 시스템은 Clearpath Robotics에서 개발한 Jackal 로봇이다. Jackal은 소형 차륜형 무인 이동체로, ROS 2 환경에서 제어 및 센서 데이터를 발행할 수 있는 온보드 PC를 탑재하고 있으며, 다양한 센서를 장착할 수 있어 연구 및 실험 환경에서 활용하기에 적합하다. 그림 1은 연구에 사용한 모바일 로봇의 외관이고, 세부 사양은 표 1에 정리하였다[7].

구동 방식은 좌·우 바퀴의 속도 차이를 이용하여 구동하는 차동(Differential drive) 구조로, ROS 2에서 제공하는 `diff_drive_controller`를 통해 수행된다. 해당 컨트롤러는 ROS 2의 Twist 메시지로 입력되는 선속도(Linear.x)와 각속도(Angular.z)를 좌·우 바퀴 속도로 변환하여 제어 신호를 생성한다. 좌·우 바퀴의 속도 w_L, w_R 는 식 (1)과 같이 계산된다[8].

$$w_L = \frac{V - w \cdot \frac{b}{2}}{r}, \quad w_R = \frac{V + w \cdot \frac{b}{2}}{r} \quad (1)$$

여기서 b 는 바퀴 간 거리(Wheel separation), r 는 바퀴 반지름(Wheel radius)을 의미한다. 이렇게 산출된 각 바퀴의 회전 속도는 모터 드라이버를 통해 각 바퀴에 전달되며, 이를 통해 직진 및 회전과 같은 기본 주행 동작을 수행한다.

또한 모바일 로봇에는 Ouster OS1(REV.7) LiDAR

(최대 탐지거리 170m, 128채널, 360° 시야각)가 장착되어 3차원 포인트 클라우드 데이터를 ROS 2 토픽(/pointcloud)으로 발행하도록 구성되어있다.



그림 1. 모바일 로봇 외관(Ouster LiDAR 장착)
Fig. 1. Mobile robot with mounted ouster LiDAR

표 1. 모바일 로봇 사양

Table 1. Specifications of the mobile robot

Item	Specification
Size	508 x 430 x 500 mm
Weight	19 kg
Drive type	Differential drive
Maximum speed	2.0 m/s
Payload capacity	Up to 20 kg

2.2 가상 환경 플랫폼

가상 환경 플랫폼으로는 NVIDIA Omniverse 기반의 시뮬레이터 Isaac Sim을 사용하였다. Isaac Sim은 실제 환경과 유사한 물리 엔진과 다양한 센서 시뮬레이션 기능을 제공하며, ROS 2와의 연동을 공식적으로 지원한다[9]. 본 연구에서는 이러한 특성을 활용하여 가상 로봇을 실제 로봇과 동일한 제어 및 센서 구조로 동작하도록 구성하였다. 특히 Action Graph기능을 이용해 ROS 2 토픽 발행·구독을 구현함으로써, 가상 환경에서 생성된 제어 입력과 센서 데이터가 실제 로봇과 호환 가능한 형태로 처리되도록 하였다. 세부적인 노드 구성과 동작 절차는 3장에서 설명한다.

2.3 ROS 2 연동 구조

실제 로봇과 가상 로봇 간 데이터 교환은 ROS 2 토픽을 기반으로 이루어진다. 제어 명령은 /cmd_vel 토픽으로 발행되며, 동일한 메시지가 모바일 로봇과 Isaac Sim 내 모바일 로봇에 동시에 전달된다. 모바일 로봇은 /odom 토픽, 가상 모바일 로봇은 /odom_sim 토픽을 통해 오도메트리 데이터를 발행하고, /tf 토픽으로 좌표 변환 정보를 제공한다.

센서 데이터의 경우, 모바일 로봇에 장착된 Ouster LiDAR는 /pointcloud 토픽으로 현실 환경을 스캔한 데이터를 발행한다. Isaac Sim에서 동일한 Ouster LiDAR를 사용하여 가상 환경을 스캔한 데이터를 /point_cloud_sim 토픽으로 발행한다. 두 데이터는 센서 모델은 동일하나, 실제와 가상 환경에서 획득된 데이터라는 차이가 있으므로 분리된 토픽으로 관리하였다[10]. 그림 2는 본 연구에서 구현한 디지털 트윈 시스템의 전체 구조를 나타낸 것으로, 실제 로봇과 가상 로봇 사이의 데이터 흐름을 시각화한 것이다.

네트워크 연결은 무선(Wi-Fi) LAN 환경에서 공유기를 통해 구성하였으며, 모든 장치는 동일 네트워크에 접속하여 ROS 2의 DDS 미들웨어를 통해 토픽을 교환하였다. 또한 무선 환경에서 발생할 수 있는 시간 차이를 최소화하기 위해, 리눅스 기반 시간 동기화 서비스인 Chrony를 사용하여 로봇과 시뮬레이션 PC의 시간을 동기화하였다.

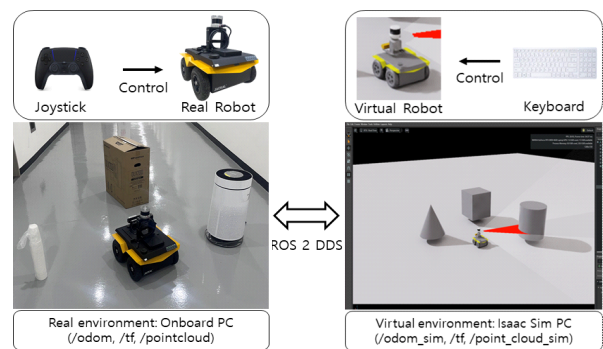


그림 2. 디지털트윈 구현 흐름도
Fig. 2. Flowchart of digital twin implementation

III. 구현 방법

3.1 가상 로봇 구현

본 연구에서는 NVIDIA Isaac Sim을 활용하여 모바일 로봇의 가상환경 모델을 구현하였다. Isaac Sim에서 제공하는 Jackal 로봇 Asset을 사용하였고, 기본 차체(base_link), 휠 조인트, 센서 장착부 등이 계층 구조(Tree)로 구성되어 있고, 구현된 가상 모바일 로봇은 그림 3과 같이 Isaac Sim의 Stage창에서 계층 구조로 모델링되어 있음을 확인할 수 있다. 이러한 구조는 실제 모바일 로봇의 물리적 구성을 반영하여 가상 환경에서도 동일한 구동 방식과 센서 배치를 재현할 수 있도록 한다. 그림 3은

가상 로봇은 ROS 2를 통해 오도메트리, 좌표계, 센서 데이터를 발행할 수 있도록 설정되었으며, 구체적인 제어 명령 처리와 센서 연동은 Isaac Sim의 Action Graph로 구현하였다. 이에 대한 세부 내용은 3.2와 3.3절에서 다룬다.

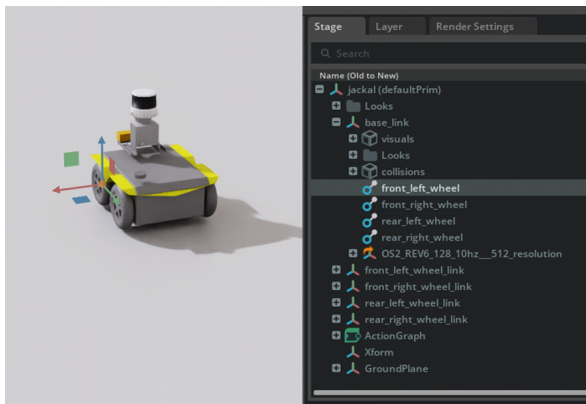


그림 3. Isaac Sim Stage 상의 Jackal 로봇과 계층 구조
Fig. 3. Jackal robot and Stage hierarchy in Isaac Sim

3.2 제어 연동

본 연구에서는 ROS 2의 Twist 메시지를 기반으로 실제와 가상 로봇의 제어를 연동하였다. 모바일 로봇은 조이스틱 입력을 받아 /cmd_vel 토픽을 발행하고, Isaac Sim에서는 키보드 입력을 통해 동일한 /cmd_vel 토픽을 발행하였다. 두 로봇은 공통적으로 해당 토픽을 구독하여 동일한 제어 명령을 수신하므로, 실제와 가상 환경에서 일관된 동작이 이루어질 수 있도록 구성하였다.

가상 로봇의 경우, /cmd_vel 토픽으로부터 전달된 Twist 메시지는 Isaac Sim의 Action Graph에서 처리

된다. Action Graph 내에서는 ROS2 Subscribe Twist 노드를 통해 메시지를 수신한 뒤, 이를 실제 로봇의 diff_drive_controller와 동일한 역할을 하는 Differential Controller 노드로 전달하여 좌·우 바퀴의 속도 신호로 변환한다. 변환된 속도 신호는 Articulation Controller 노드를 통해 휠 조인트에 적용되어, 모바일 로봇과 동일한 차동 구동 방식이 시뮬레이션에서도 구현되도록 하였다. 또한 가상 로봇과 실제 로봇의 동작 오차를 최소화하기 위해 Differential Controller 노드의 주요 파라미터(b , r)를 실험을 통해 결정하였다.

또한 본 구조는 양방향 제어가 가능하다. 즉, 실제 로봇에서 발행된 /cmd_vel 입력은 시뮬레이션에 적용될 수 있으며, 반대로 Action Graph에서 발행된 제어 입력 역시 동일한 /cmd_vel 토픽을 통해 실제 로봇에 전달된다. 이를 통해 실제와 가상 로봇 간의 제어 구조가 상호 검증 가능한 디지털 트윈 환경이 구현되었다.

3.3 센서 연동 구현

센서 연동은 모바일 로봇에 장착된 Ouster LiDAR를 중심으로 구현하였다. 모바일 로봇에서는 LiDAR가 현실 환경을 스캔한 데이터를 /pointcloud 토픽으로 발행하고, Isaac Sim에서는 동일한 Ouster 모델을 기반으로 Action Graph의 RTX Lidar Helper노드를 사용하여 가상 환경을 스캔한 데이터를 /point_cloud_sim 토픽으로 발행하였다. 두 데이터는 모두 ROS 2의 PointCloud2 형식을 따르지만, 획득된 환경이 다르므로 별도의 토픽으로 분리하여 관리하였다.

발행된 데이터는 ROS 2에서 제공하는 시각화 도구인 RViz2에서 정상적으로 시각화되었으며, 이를 통해 실제와 가상 환경에서 생성된 센서 데이터가 동일한 ROS 2 네트워크상에서 올바르게 처리될 수 있음을 확인하였다. 이러한 구조는 LiDAR뿐만 아니라 IMU, 카메라 등 다른 센서 데이터 처리에도 동일하게 확장 가능하다.

그림 4는 실제 모바일 로봇과 연동 구조도를 나타낸 것으로, ROS 2토픽을 통해 제어명령과 센서 데이터가 양방향으로 전달되는 전체 흐름을 보여준다.

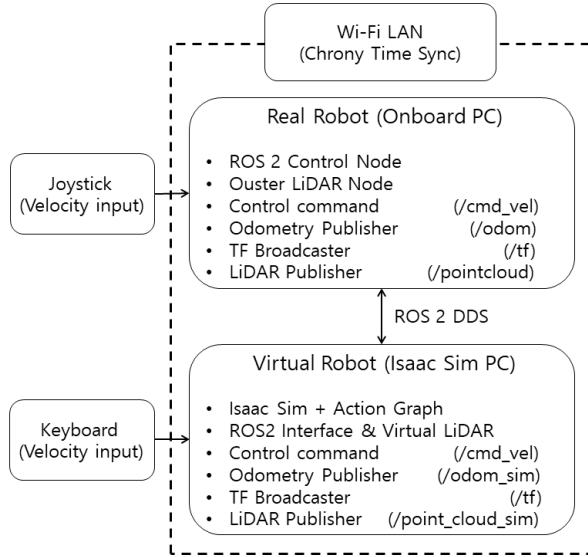


그림 4. 실제·가상 모바일 로봇 연동 구조도

Fig. 4. Integration architecture of real and virtual mobile robot

IV. 실험 및 결과

4.1 동작 검증

본 연구에서는 구현된 디지털 트윈 환경의 타당성을 검증하기 위해 모바일 로봇과 가상 로봇의 동작을 비교하였다. 본 실험에서는 키보드를 통해 ROS 2의 /cmd_vel 토픽으로 제어 입력을 발행하였다.

실험 환경은 그림 5와 같이 실내의 평탄한 바닥 (약 $1\text{m} \times 2\text{m}$ 공간)에서 수행되었으며, 두 가지 시나리오로 구성되었다. 첫째, 5초 동안 0.4m/s 의 선속도 입력을 주어 전진하는 주행 실험, 둘째, 8초 동안 0.6rad/s 의 각속도 입력을 주어 제자리 회전하는 실험이다. 각 시나리오는 10회씩 반복 수행하였으며, 실제 실험 환경은 그림 5에 제시하였다. 전진 시나리오의 5초는 로봇이 가속 구간을 지나 안정적인 속도를 유지할 수 있는 시간으로 설정하였으며, 회전 시나리오의 8초는 로봇이 약 180° 이상 회전하면서 충분한 각속도 응답 데이터를 확보할 수 있는 시간으로 설정하였다. 이러한 시간 기준은 실제 로봇과 가상 로봇의 동작 특성이 동일한 제어 구간에서 비교될 수 있도록 하기 위한 공통 기준으로 사용되었다.



그림 5. 실험 환경

Fig. 5. Experimental environment

전진 시나리오에서는 /odom에 기록된 위치 좌표를 누적하여 총 이동 거리를 계산하였으며, 그 계산식은 식 (2)과 같고, L :이동거리, N :데이터 개수, x_k : k 번째의 x 좌표, y_k : k 번째의 y 좌표이다.

$$L = \sum_{k=1}^{N-1} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \quad (2)$$

회전 시나리오에서는 /odom 메시지에 포함된 orientation 값을 실시간으로 yaw θ 로 변환하여 누적 회전 각도를 계산하였다. yaw θ 변환식은 식 (3)과 같고, w, x, y, z 는 쿼터니언(Quaternion)의 네 요소로 로봇의 orientation을 표현하기 위한 회전 성분이다. w : 스칼라, x, y, z :벡터 성분으로 구성된다.

$$\theta = \arctan2(2(wz + xy), 1 - 2(y^2 + z^2)) \quad (3)$$

이동 거리와 회전 각도의 차이는 평균 절대 오차 (Mean Absolute Error, MAE)로 산출하였으며, 그 정의는 각각 식 (4), (5)와 같다.

$$MAE_{distance} = \frac{1}{N} \sum_{i=1}^N |L_{real,i} - L_{sim,i}| \quad (4)$$

$$MAE_{rotate} = \frac{1}{N} \sum_{i=1}^N |\theta_{real,i} - \theta_{sim,i}| \quad (5)$$

실험 결과, 전진 시나리오에서 실제 로봇의 평균 이동 거리는 1.84 ± 0.03 m, 가상 로봇의 평균 이동 거리는 1.81 ± 0.04 m로 나타났다. 두 결과 간 평균 절대 오차(MAE)는 0.03 m로, 실제와 가상 환경에서 매우 유사한 주행 성능을 보였다.

회전 시나리오에서는 실제 로봇의 평균 회전 각도가 $194.12 \pm 2.35^\circ$, 가상 로봇의 평균 회전 각도가 $187.85 \pm 6.07^\circ$ 로 측정되었으며, MAE는 6.88° 로 확인되었다. 이는 실제와 가상 로봇이 동일한 제어 입력에 대해 대체로 일관된 회전 동작을 수행함을 의미하나, 가상 환경의 물리 파라미터 차이와 실제 주행 시 발생할 수 있는 바퀴 슬립, 무선 네트워크 지연 등의 요인으로 오차가 발생한 것으로 판단된다 [11]. 정량적인 비교 결과는 표 2에 정리하였다. 여기서 표 2의 Forward motion (5 s)과 Rotational motion (8 s)은 각 시나리오에서 인가된 제어 명령의 지속 시간을 의미한다.

표 2. 실제 및 가상 모바일 로봇 동작 결과 비교

Table 2. Comparison of motion results between real and simulated mobile robot

Scenario	Real value (Mean $\pm$ SD)	Simulated value (Mean $\pm$ SD)	MAE
Forward motion(5s)	1.84 ± 0.03 m	1.81 ± 0.04 m	0.03m
Rotational motion(8s)	$194.12 \pm 2.35^\circ$	$187.85 \pm 6.07^\circ$	6.88°

본 연구에서는 실내 주행 및 자세 제어 과업에서 일반적으로 사용되는 목표 허용오차 관행을 근거로 회전 오차의 실용 허용범위를 $\pm 10^\circ$ 내외로 설정하였다. ROS 2 Navigation2의 기본 목표 판정 기준에서 yaw 허용오차 기본값은 약 0.25 rad ($\approx 14.3^\circ$)로 설정되어 있으며[12], ROS1 및 로컬 플래너 튜닝 가이드에서도 $0.05 - 0.10$ rad ($\approx 2.9 - 5.7^\circ$) 수준이 자주 활용된다[13]. 따라서 본 연구에서 측정된 평균 절대 오차 6.88° 는 이러한 허용범위 내에 해당하며, 실제와 가상 로봇의 회전 거동이 정합성을 보인다고 판단된다.

4.2 센서 데이터 검증

센서 데이터 검증은 모바일 로봇에 장착된 Ouster LiDAR를 대상으로 수행하였다. 실제 로봇은 /pointcloud 토픽을 통해 데이터를 발행하였으며, 가상 로봇은 /point_cloud_sim 토픽을 통해 데이터를 발행하였다. 두 데이터는 모두 ROS 2의 PointCloud2 형식을 사용하였으며, RViz2에서는 토픽을 설정하여 실제 데이터 또는 가상 데이터를 각각 선택적으로 시각화됨을 확인하였다. 이를 통해 실제와 가상 환경에서 생성된 LiDAR 데이터가 ROS 2 네트워크상에서 올바르게 발행·구독됨을 확인하였다. 대표적인 결과는 그림 6에 제시하였다. 또한 실제와 가상 환경에서 수집된 센서 데이터가 형상적으로 유사하며, 메시지 구조 측면에서도 상호 호환됨을 검증하였다.

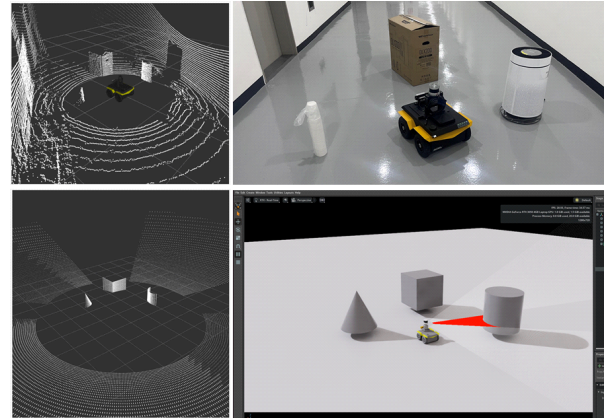


그림 6. RViz2 시각화

(상: 실제 환경, 하: 가상 환경)

Fig. 6. RViz2 visualization

(Top: Real environment, Bottom: Virtual environment)

V. 결론 및 향후 과제

본 연구에서는 ROS 2 기반의 모바일 로봇과 NVIDIA Isaac Sim을 연동하여 디지털 트윈 환경을 구현하였다. 실제 로봇과 가상 로봇은 ROS 2 토픽 (/cmd_vel, /odom, /tf, /pointcloud, /point_cloud_sim)을 통해 제어 명령과 센서 데이터를 주고받도록 구성되었으며, 가상 환경에서는 Isaac Sim의 Action Graph 기능을 활용하여 동일한 제어 구조와 센서 데이터를 재현하였다.

실험을 통해 전진 및 회전 시나리오에서 실제와 가상 로봇이 유사한 동작을 수행함을 확인하였고, 주행거리와 회전각도 비교 결과 평균 절대 오차 (MAE)는 각각 약 0.03 m와 6.88°로 나타났다. 또한 실제와 가상 LiDAR 데이터가 ROS 2 네트워크상에서 독립적으로 발행·구독되고, RViz2에서 정상적으로 시각화됨을 확인하였다. 이러한 결과는 제안한 시스템이 실제와 가상 로봇 간 상호 검증이 가능한 디지털 트윈 환경임을 보여준다.

본 연구에서는 실제와 가상 로봇의 제어 및 센서 데이터의 일관성 검증에 중점을 두었으며, 네트워크 지연(Latency)과 패킷 손실에 대한 정량적 평가는 향후 연구에서 추가적으로 수행할 예정이다.

향후 연구에서는 다음과 같은 방향으로 확장하고자 한다. 첫째, 물리 모델에서 수집한 주행 데이터를 기반으로 딥러닝 기법을 적용하여 제어기 파라미터를 자동 산출하는 방법을 제안할 예정이다. 둘째, LiDAR뿐만 아니라 IMU, 카메라 등 복합 센서 데이터를 연동하여 보다 정밀한 디지털 트윈 환경을 구축할 필요가 있다. 셋째, 가상 환경에서 학습된 제어 명령을 실제 로봇에 적용하는 실험적 검증을 수행함으로써, 양방향 디지털 트윈의 유효성을 평가할 계획이다. 넷째, 다양한 장애물 인식 및 회피 시나리오와 같은 응용 연구로 확장하여 제안한 시스템의 활용성을 검증하고자 한다. 마지막으로, 무선 네트워크 환경에서 발생할 수 있는 지연 (Latency)과 패킷 손실 문제를 최소화하기 위해 QoS 설정과 실시간 동기화 기법을 연구할 계획이다.

References

- [1] M. B. A. Zami, S. Shaon, V. K. Quy, and D. C. Nguyen, "Digital Twin in Industries: A Comprehensive Survey", *IEEE Access*, Vol. 13, pp. 47291-47336, Mar. 2025. <https://doi.org/10.1109/ACCESS.2025.3551532>.
- [2] Rojas, M.; Hermosilla, G.; Yunge, D.; Farias, G. "An Easy to Use Deep Reinforcement Learning Library for AI Mobile Robots in Isaac Sim", *Appl Science*, Vol. 12, No 17, pp. 8429, Aug. 2022. <https://doi.org/10.3390/app12178429>.
- [3] S. Macenski, et al., "Robot operating system 2: design, architecture, and uses in the wild", *Science Robot*, Vol. 7, No 66, May 2022. <https://doi.org/10.1126/scirobotics.abm6074>.
- [4] T. W. Lim and S. H. Lee, "A Study on the Establishment of ROS-based Digital Twin Environment Using NVIDIA Omniverse", *Proc. of the KIIT Conference*, Jeju, Korea, pp. 356-358, Jun. 2022.
- [5] Setting Up the Jackal UGV on ROS 2 Humble, https://github.com/r-shima/jackal_ros2_humble. [accessed: Feb. 12, 2025]
- [6] S. Nambiar, et al., "Digital Twin-Enabled Adaptive Robotics: Leveraging Large Language Models in Isaac Sim for Unstructured Environments", *Machines* Vol. 13, No. 7, Jul. 2025. <https://doi.org/10.3390/machines13070620>.
- [7] JACKAL robot Dataset, <https://roas.co.kr/jackal-ugv/>. [accessed: Jun. 12, 2025]
- [8] ROS 2 DiffDrive Controller Documentation, https://wiki.ros.org/diff_drive_controller. [accessed: Jun. 12, 2025]
- [9] ROS 2 Documentation - Install ROS 2, <https://docs.ros.org/en/humble/Installation.html>. [accessed: Feb. 12, 2025]
- [10] C. Lee, Y. Kim, J. H. Shim, M. Cho, S. Kim, N. Choi, and T.-R. Lee, "Development of ROS-based Personal Mobility Remote Control System with Digital Twin Model", *Proc. of the Korean Institute of Information Scientists and Engineers (KIISE) Conference*, Jeju, Korea, pp. 1386-1388, Jun. 2019.
- [11] J. Albardaner, et al., "Sim-to-real Gap in Reinforcement Learning: Use Case with TIAGo and Isaac Sim/Gym", *Proc. of the European Robotics Forum (ERF)*, Cham: Springer Nature Switzerland, Rimini, Italy, pp. 344-348, Mar. 2024.
- [12] ROS 2 Navigation2 Documentation, https://docs.nav2.org/configuration/packages/nav2_controller-plugins/

simple_goal_checker.html#simplegoalchecker.

[accessed: Oct. 7, 2025]

[13] ROS Navigation Tuning Guide, <https://kaiyuzheng.me/documents/navguide.pdf>. [accessed: Oct. 7, 2025]

저자소개

최 성 식 (Seong-Sig Choi)



2018년 2월 : 전주대학교

전기전자공학과(공학사)

2025년 2월 : 경북대학교

기계공학과(공학석사)

2025년 6월 ~ 현재 :

한국생산기술연구원 연구원

관심분야 : 디지털 트윈, 로봇

제어, 착용형 재활로봇

최 승 환 (Seung-Hwan Choi)



2012년 2월 : 동아대학교

전자공학과(공학사)

2014년 2월 : 부산대학교

로봇융합전공(공학석사)

2013년 12월 ~ 현재 :

한국생산기술연구원 선임연구원

관심분야 : 로봇제어 및 신호처리,

이상 검출 및 상태진단, AI 알고리즘

김 창 현 (Chang-Hyun Kim)



2016년 2월 : 금오공과대학교

기전공학과(공학사)

2018년 2월 : 금오공과대학교

메디컬IT융합공학과(공학석사)

2018년 4월 ~ 현재 :

한국생산기술연구원 연구원

2019년 3월 ~ 현재 : 경북대학교

기계공학과(공학박사)

관심분야 : 디지털 트윈, 로봇 제어, AI 알고리즘,
신호처리, 센서 신호 기반 데이터 분류

권 효 은 (Hyoeyun Kwon)



2021년 2월 : 금오공과대학교

전자공학부(공학사)

2024년 12월 ~ 현재 :

한국생산기술연구원 연구원

2025년 3월 ~ 현재 : DGIST

로봇및기계전자공학과 석사과정

관심분야 : 디지털 트윈, 로봇 제어

이 수 웅 (Suwoong Lee)



1999년 2월 : 동아대학교

전기공학과(공학사)

2002년 3월 : University of

Tsukbua(공학석사)

2005년 3월 : University of

Tsukbua(공학박사)

2005년 4월 ~ 2009년 3월 : 일본

AIST(산업기술종합연구소) Intelligent System Research

Institutte 박사후연구원

2009년 4월 ~ 2012년 11월 : Yamagata University Dept.

of Bio-System Engineering 조교수

2012년 12월 ~ 현재 : 한국생산기술연구원

모빌리티시스템그룹 수석연구원(그룹장)

관심분야 : 물리적 인간-로봇상호작용, 로봇기술 응용
운동기구, 로봇안전, 영상 기반 촉각센서