두산 Rokey Boot Camp

스터디 주간 활동 보고서

팀명	**벌꿀오소리**	제출자 성명	**정민섭**
참여 명단	**정찬원, 송주훈, 강인우, 정민섭**		
모임 일시	**2025 년 4 월 24 일 21 시 ~ 22 시(총 1 시간)**		
장소	**Google meet 화상회의**	출석 인원	**4 / 4**
학습목표	로봇 패키지를 받아서 시뮬레이션 및 TF 확인해보기		
학습내용	● **정찬원: Gazebo 와 SLAM 을 이용한 turtlebot3 시뮬레이션** TurtleBot3 와 ROS 2 Humble 을 기반으로, gazebo 를 이용해 로봇을 가상 환경에서 시뮬레이션 가능하다. Cartographer → 구글(Google)에서 오픈소스로 개발한 **2D/3D 실시간 SLAM 라이브러리.** 센서 데이터를 실시간으로 처리해서, 로봇이 모르는		

공간에서도 스스로 지도를 만들고 위치를 추정할 수 있게 해주는 SLAM 엔진.

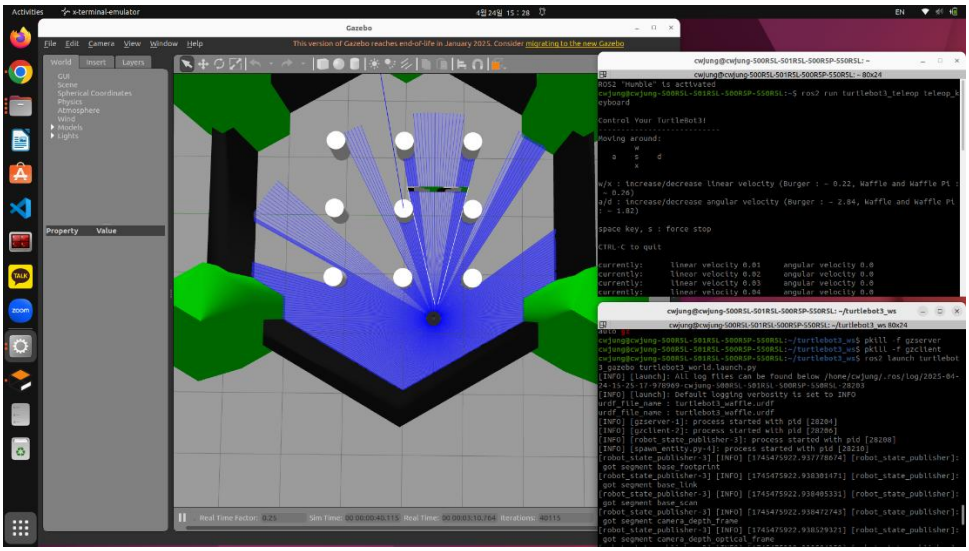
1. 시뮬레이션 월드 launch

다음 명령어로 gazebo 에서 시뮬레이션 열기

```
$ export TURTLEBOT3_MODEL=waffle
$ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

teleop_keyboard 로 turtlebot 키보드로 제어하기

```
$ ros2 run turtlebot3_teleop teleop_keyboard
```



<Fig 1. Turtlebot3 를 gazebo 상에서 제어하는 모습>

2. SLAM Simulation

- Simulation World launch

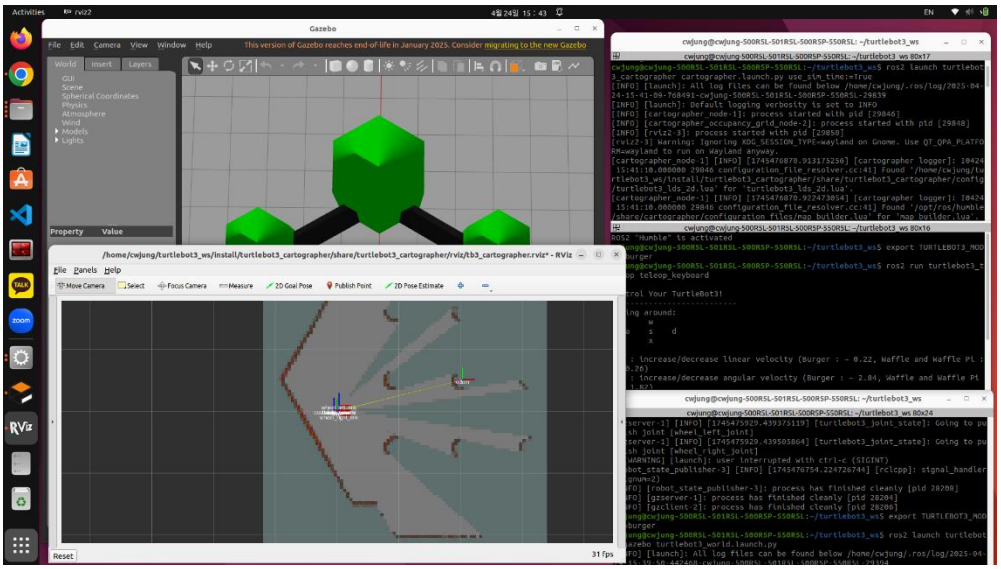
```
$ export TURTLEBOT3_MODEL=burger
$ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

- SLAM 노드 실행(다른 터미널 창 열고)

```
$ export TURTLEBOT3_MODEL=burger
$ ros2 launch turtlebot3_cartographer cartographer.launch.py use_sim_time:=True
```

- teleoperation node run (다른 터미널 창 열고)

```
$ export TURTLEBOT3_MODEL=burger
$ ros2 run turtlebot3_teleop teleop_keyboard
```



<Fig 2. SLAM 으로 Map 을 인식하는 모습>

\

- **송주훈: Gazebo 와 Rviz 를 이용한 turtlebot3 시뮬레이션**

1. gazebo, slam, navigation, turtlebot 과 시뮬레이션 패키지 설치

3. 1. 3. Install Dependent ROS 2 Packages

1. Open the terminal with Ctrl+Alt+T on the **Remote PC**.
2. Install Gazebo

[Remote PC]

```
$ sudo apt install ros-humble-gazebo-*
```

3. Install Cartographer

[Remote PC]

```
$ sudo apt install ros-humble-cartographer
$ sudo apt install ros-humble-cartographer-ros
```

4. Install Navigation2

[Remote PC]

```
$ sudo apt install ros-humble-navigation2
$ sudo apt install ros-humble-nav2-bringup
```

3. 1. 4. Install TurtleBot3 Packages

Install the required TurtleBot3 Packages.

[Remote PC]

```
$ source /opt/ros/humble/setup.bash
$ mkdir -p ~/turtlebot3_ws/src
$ cd ~/turtlebot3_ws/src/
$ git clone -b humble https://github.com/ROBOTIS-GIT/DynamixelSDK.git
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3.git
$ sudo apt install python3-colcon-common-extensions
$ cd ~/turtlebot3_ws
$ colcon build --symlink-install
$ echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc
$ source ~/.bashrc
```

6. 1. 1. Install Simulation Package

The **TurtleBot3 Simulation Package** requires `turtlebot3` and `turtlebot3_msgs` packages. Without these prerequisite packages, the Simulation cannot be launched.
Please follow the PC Setup instructions if you did not install required packages and dependent packages.

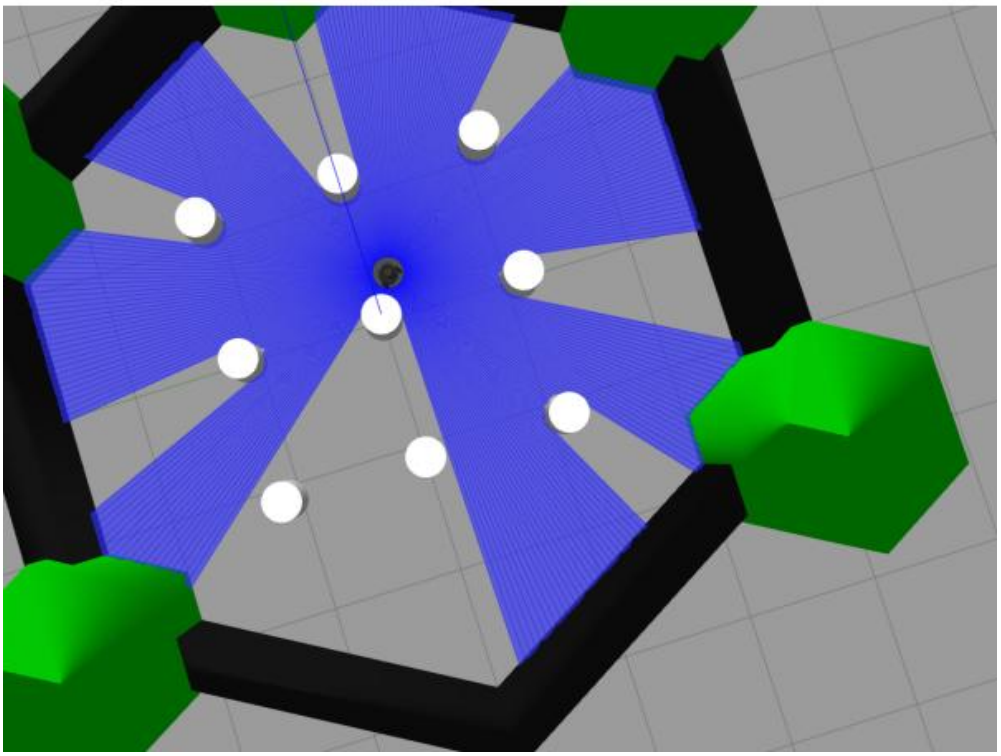
```
$ cd ~/turtlebot3_ws/src/
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3_simulations.g
$ cd ~/turtlebot3_ws && colcon build --symlink-install
```

시뮬레이션 생성 시, TURTLEBOT3_MODEL 을 waffle 또는 buger 로 입력하여 모델을 선택할 수 있다.

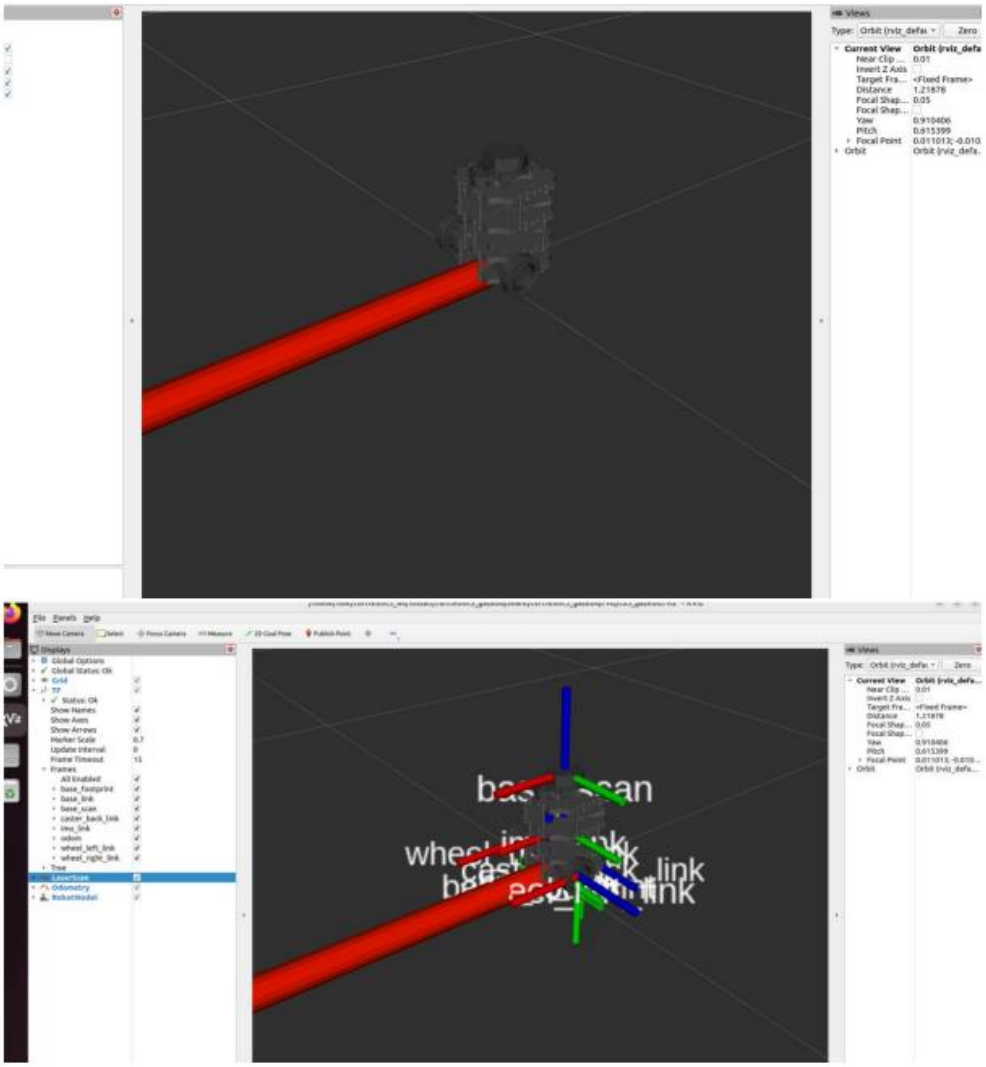
```
$ export TURTLEBOT3_MODEL=waffle
$ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

```
hbk@hbk-IdeaPad-Slim-3-16ABR8:~/turtlebot3_ws$ export TURTLEBOT3_MODEL=burger
hbk@hbk-IdeaPad-Slim-3-16ABR8:~/turtlebot3_ws$ ros2 run turtlebot3_teleop teleop_ke
yboard

Control Your TurtleBot3!
---------------------------
Moving around:
        w
   a    s    d
        x
```



<Fig 3. Turtlebot3 를 gazebo 상에서 제어하는 모습>



<Fig 4. Rviz 를 통해 Turtlebot3 의 TF 를 확인하는 모습>

- **강인우: Gazebo 및 Moveit 을 이용한 Turtlebot3 와 UR5e 시뮬레이션**

1. Turtlebot3 시뮬레이션

   - 위의 내용과 동일하기 때문에 생략

2. UR5e 시뮬레이션

UR5e: Universal Robots 협동 로봇

패키지 설치

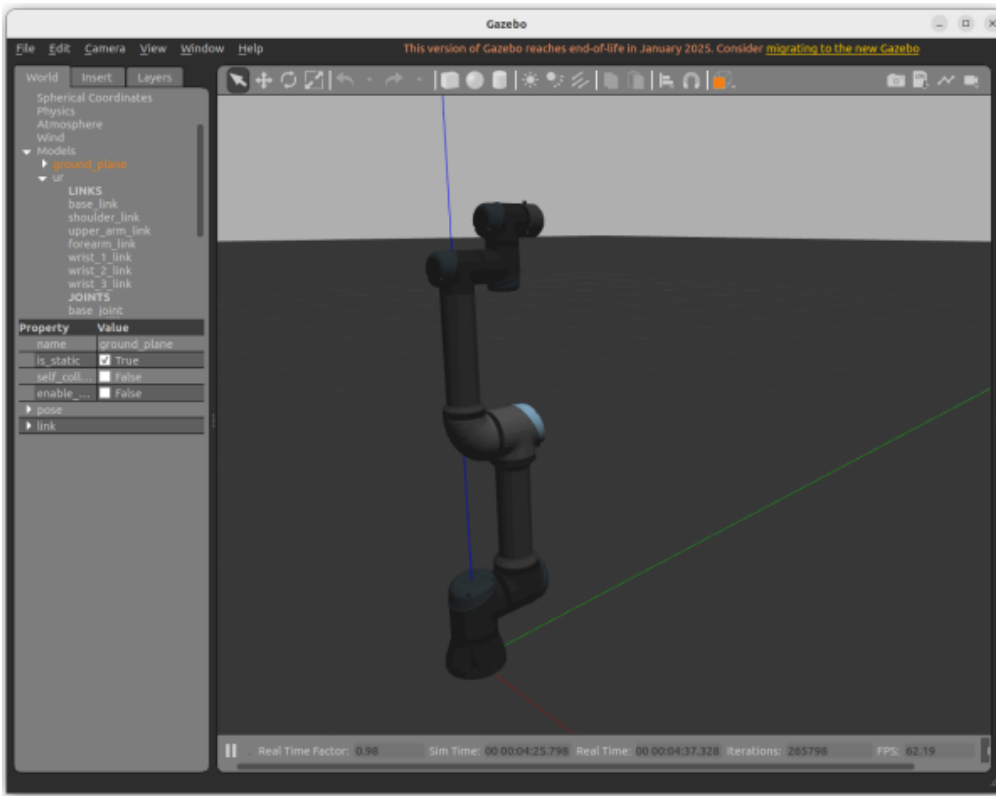
```
# Universal Robots Gazebo 시뮬레이션 패키지 클론
git clone https://github.com/UniversalRobots/Universal_Robots_ROS2_Gazebo

# ROS 2 의존성 설치
sudo apt update
rosdep update
rosdep install --from-paths src --ignore-src -r -y
```

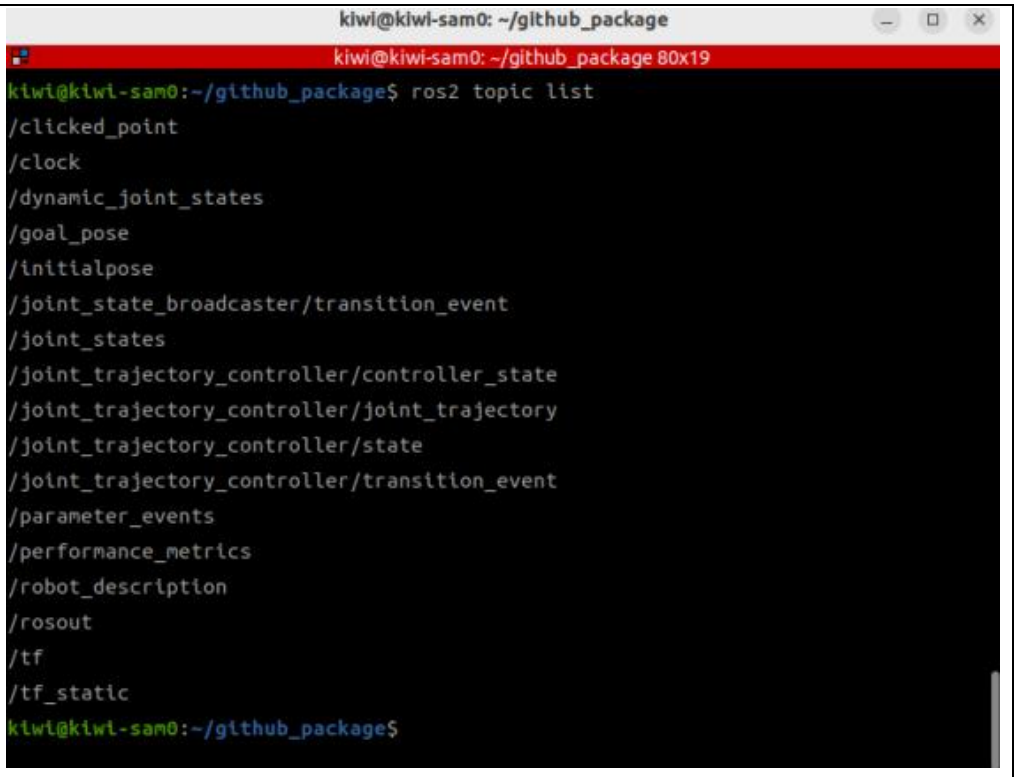
시뮬레이션 실행

#ros2 launch ur_simulation_gazebo ur_sim_control.launch.py
#ros2 launch ur_robot_driver test_joint_trajectory_controller.launch.py

ros2 launch ur_simulation_gazebo ur_sim_moveit.launch.py

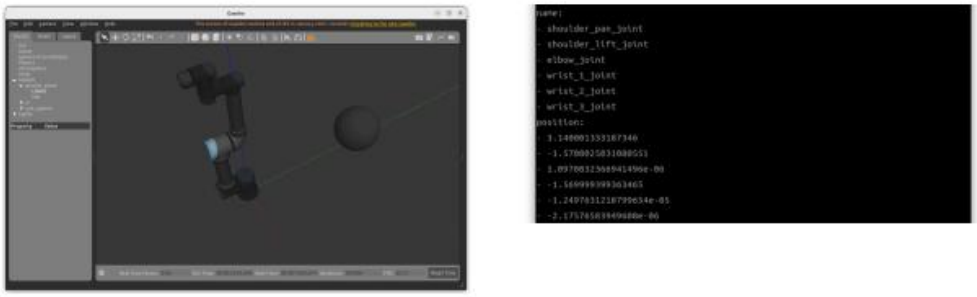


Topic list 확인



Topic 발행해서 shoulder_pan_joint 각도 수정하기

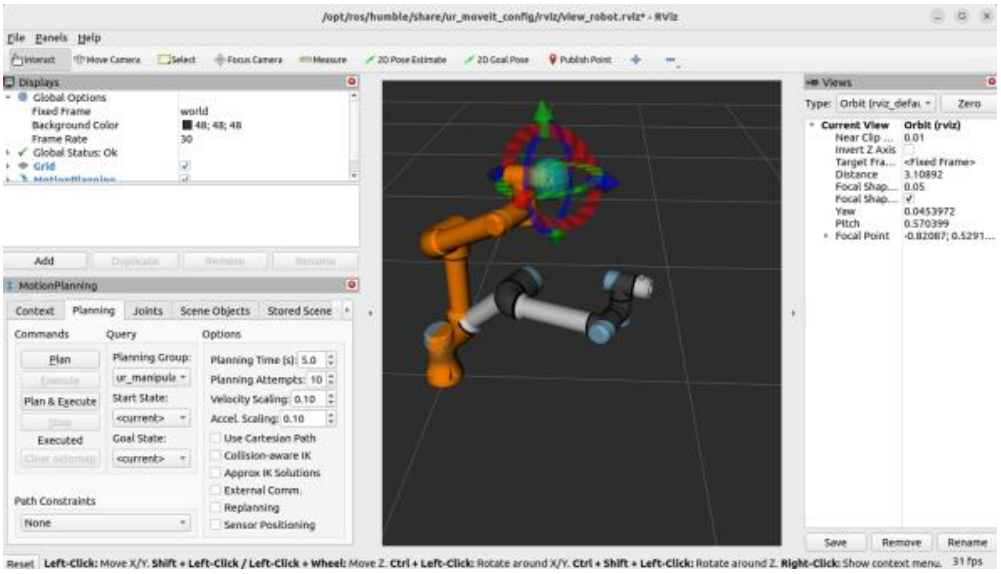
```
ros2 topic pub --once /joint_trajectory_controller/joint_trajectory trajectory_ms
  stamp:
    sec: 0
    nanosec: 0
  frame_id: ''
joint_names:
- shoulder_pan_joint
- shoulder_lift_joint
- elbow_joint
- wrist_1_joint
- wrist_2_joint
- wrist_3_joint
points:
- positions: [3.14, -1.57, 0.0, -1.57, 0.0, 0.0]
  time_from_start:
    sec: 3
    nanosec: 0"
```



<Fig 5. Topic 발행으로 매니퓰레이터를 제어하는 모습>

3. MoveIt2 로 제어

목표 위치 설정 -> Plan execute



<Fig 6. MoveIt2 로 매니퓰레이터를 제어하는 모습>

- **정민섭: Webots 패키지를 이용한 Spot Simulation**

Webots VS Gazebo

항목	Webots	Gazebo (Ignition 포함)
🛠️ 개발사	Cyberbotics	Open Source Robotics Foundation (OSRF)
🌐 오픈소스 여부	완전 오픈소스 (MIT)	오픈소스 (Apache 2.0)
🧩 ROS 통합	ROS2 지원 좋음 (웹 기반 UI 없이도 동작)	ROS와 깊이 통합되어 있음 (특히 ROS2는 Ignition과 통합)
✨ GUI	직관적이고 사용하기 쉬움 (드래그 앤 드롭 UI)	GUI는 기능 많지만 복잡하고 설정이 많음
📦 시뮬레이션 환경	다양한 로봇 & 환경 내장, 현실감은 약간 떨어질 수 있음	현실감 뛰어남 (렌더링, 센서 모델 정확도 우수)
🎓 학습 난이도	초보자에게 매우 적합	중급 이상 사용자에게 적합
🎥 센서 시뮬레이션	카메라, GPS, 라이다, 터치센서 등 직관적 구성	고정밀 센서 시뮬레이션 가능 (노이즈 포함 등)
🧠 사용 목적	교육용, 연구용, 프로토타이핑	연구용, 산업용, 자율주행 시뮬레이션
🚀 속도	비교적 빠르고 가볍다	무거운 시뮬레이션에서 성능 손해 있음
🧱 물리 엔진	ODE (내장, 간단한 설정)	ODE, Bullet, DART, Simbody 등 선택 가능

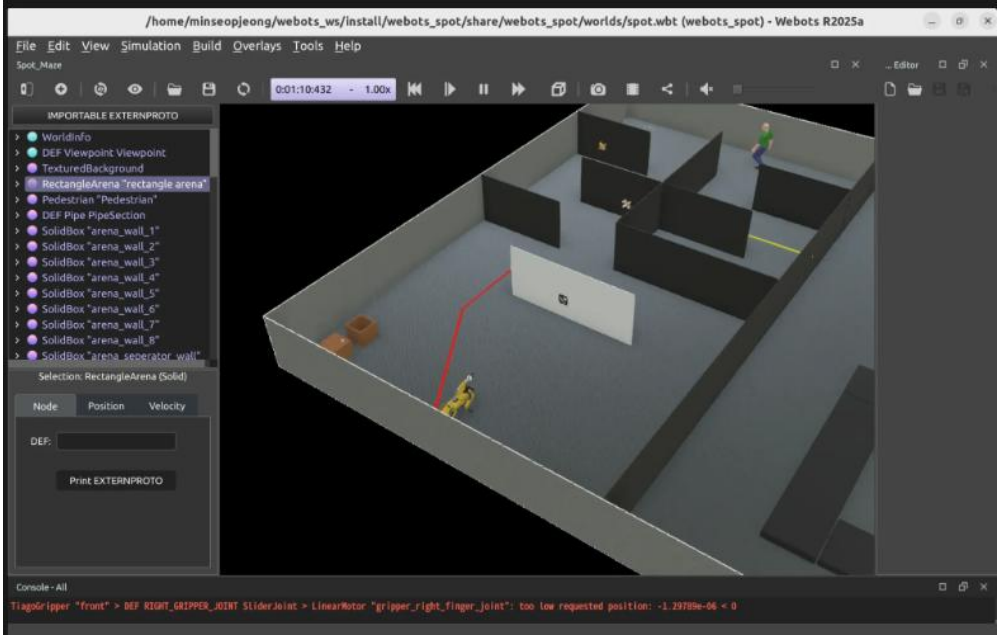
1. Webots 패키지 설치

관련 패키지를 가져오고, 종속성을 설치하고, 작업 공간을 컴파일하고 소싱

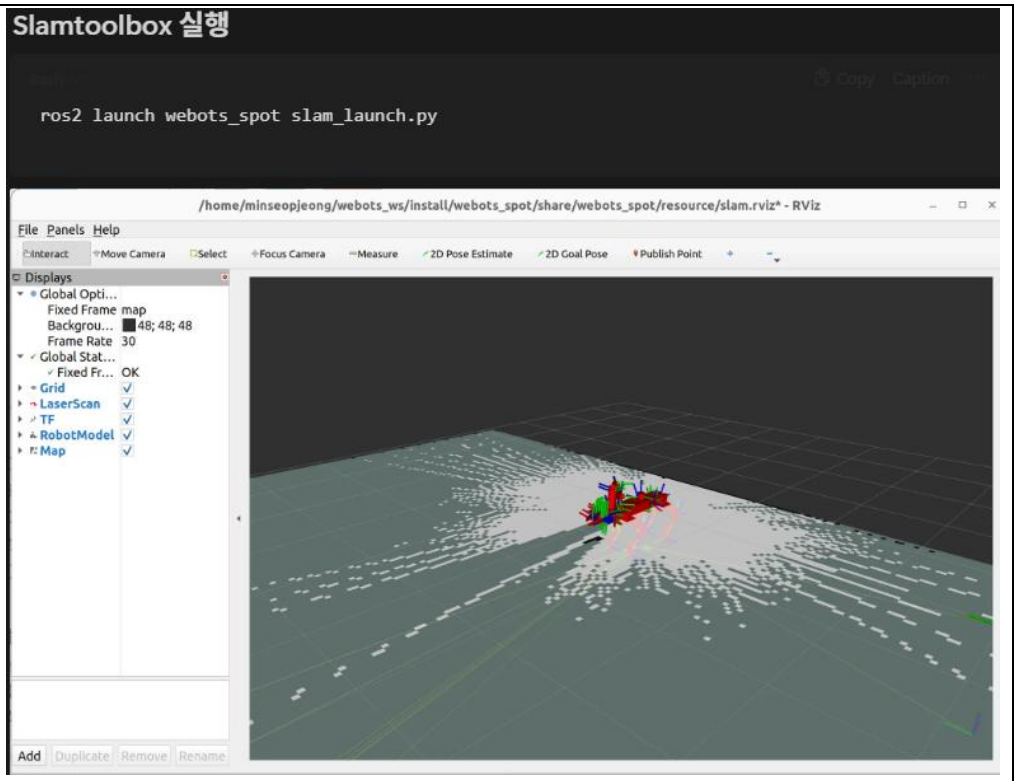
```
cd $COLCON_WS
git clone https://github.com/MASKOR/webots_ros2_spot src/webots_ros2_spot
rosdep install --ignore-src --from-paths src -y -r
vcs import --recursive src --skip-existing --input src/webots_ros2_spot/webots_ros2_spot.repos
chmod +x src/webots_ros2/webots_ros2_driver/webots_ros2_driver/ros2_supervisor.py
```

Webots 실행

```
ros2 launch webots_spot spot_launch.py
```



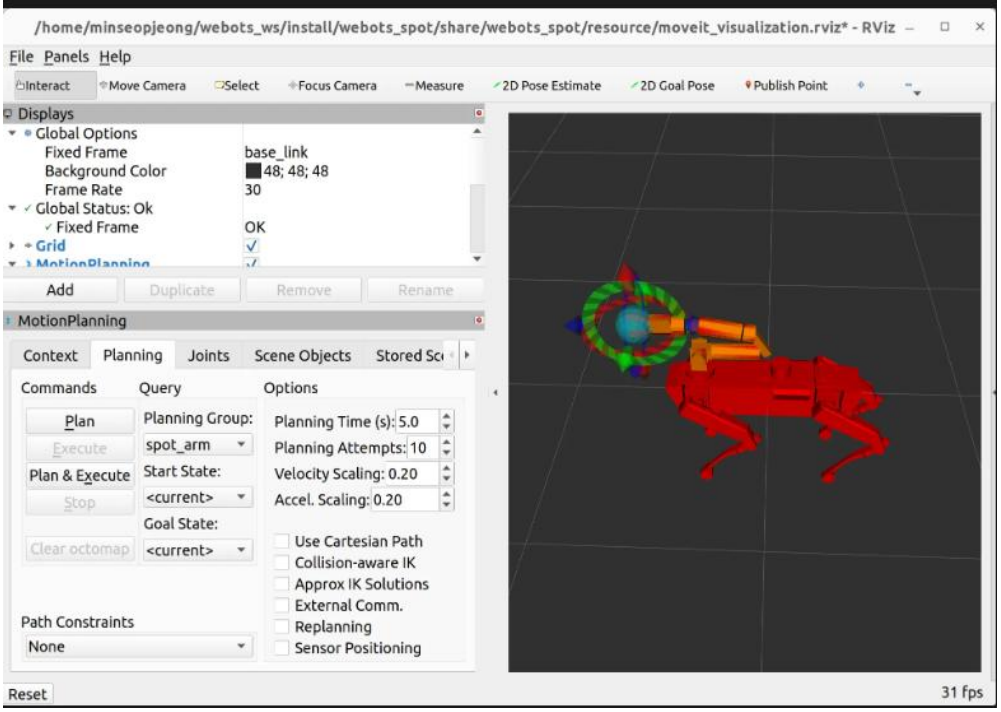
<Fig 7. Webots 상에 Spot 소환>



<Fig 8. Spot 의 센서 범위 확인>

MoveIt 실행

```
ros2 launch webots_spot moveit_launch.py
```

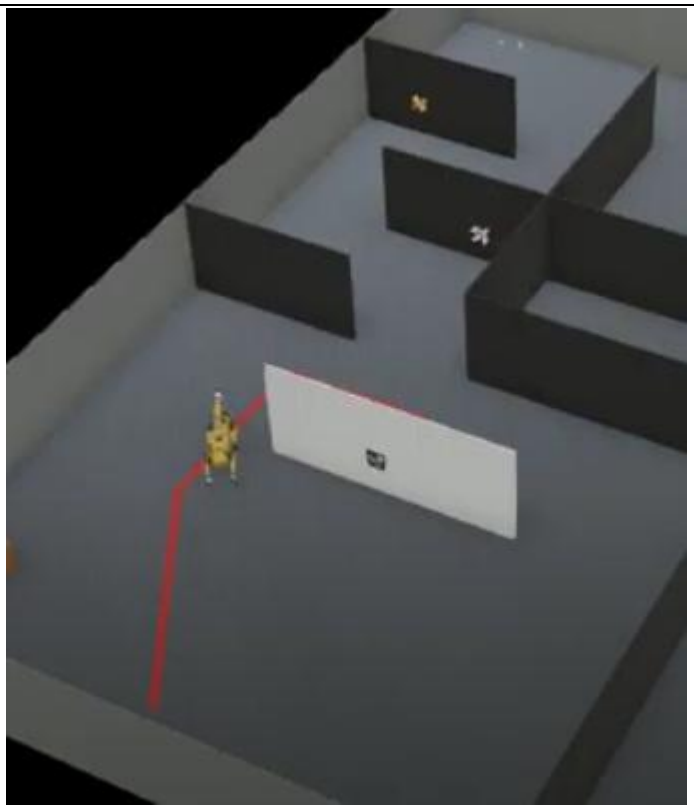


<Fig 9. MoveIt 을 통해 매니퓰레이터 확인>

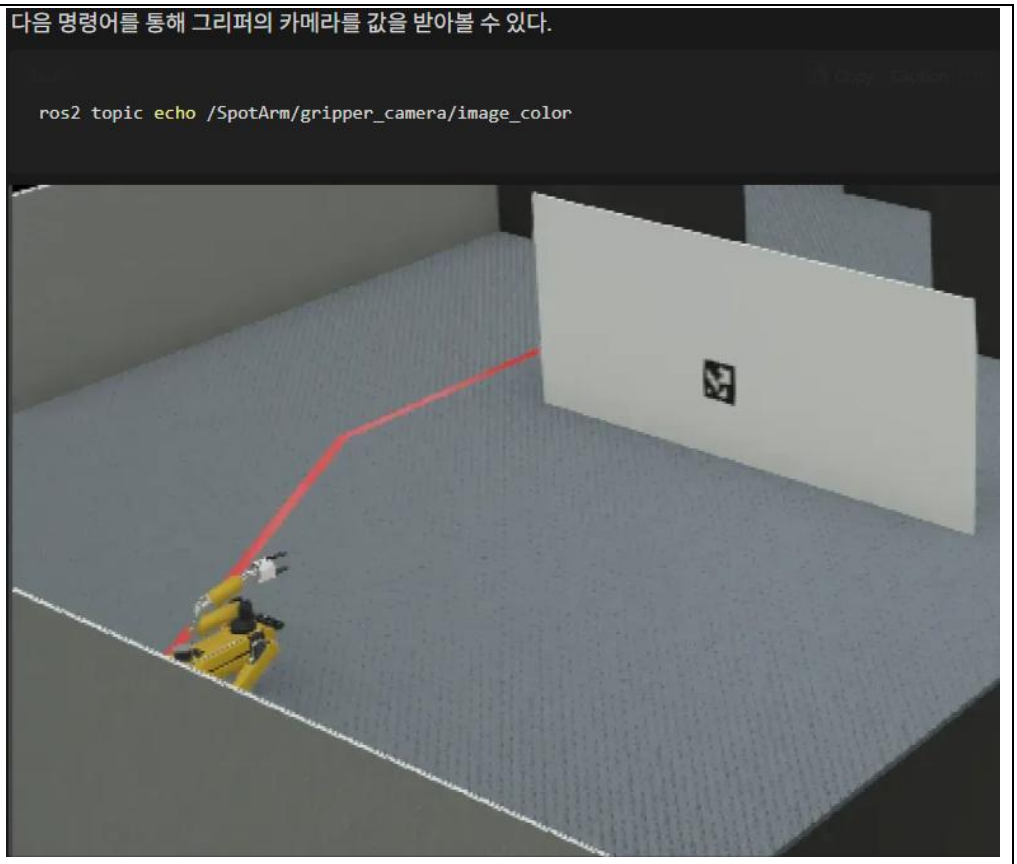
2. Webots 상에서 Teleop keyboard 를 사용하여 Spot 제어

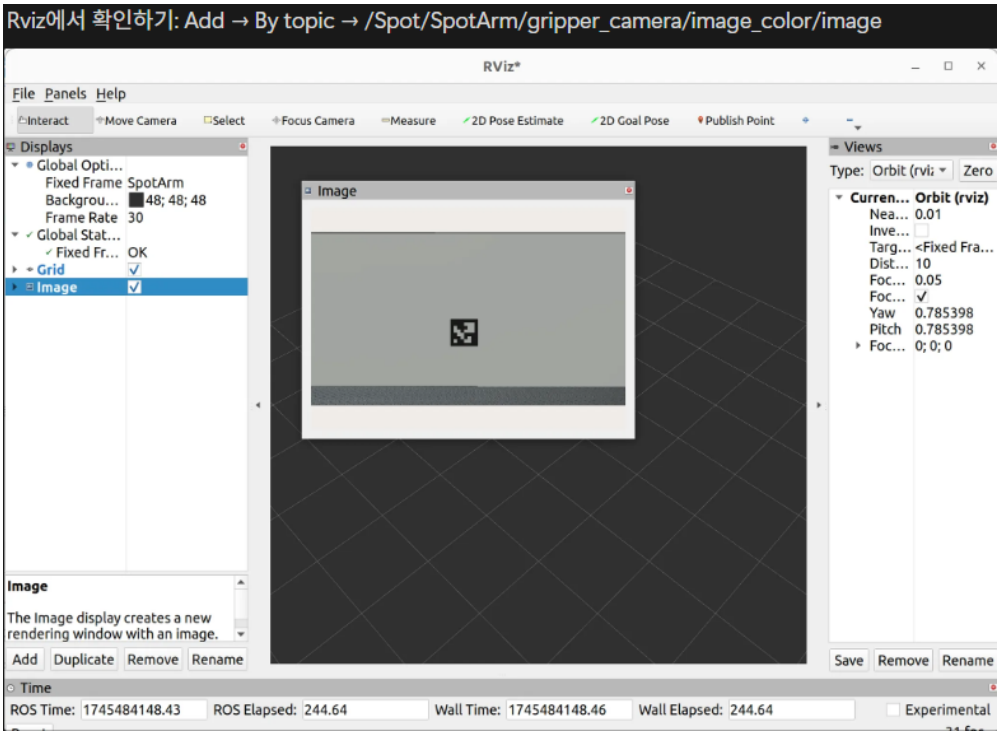
Teleop keyboard

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```



<Fig 10. Spot 을 teleop_key 로 제어하는 모습>



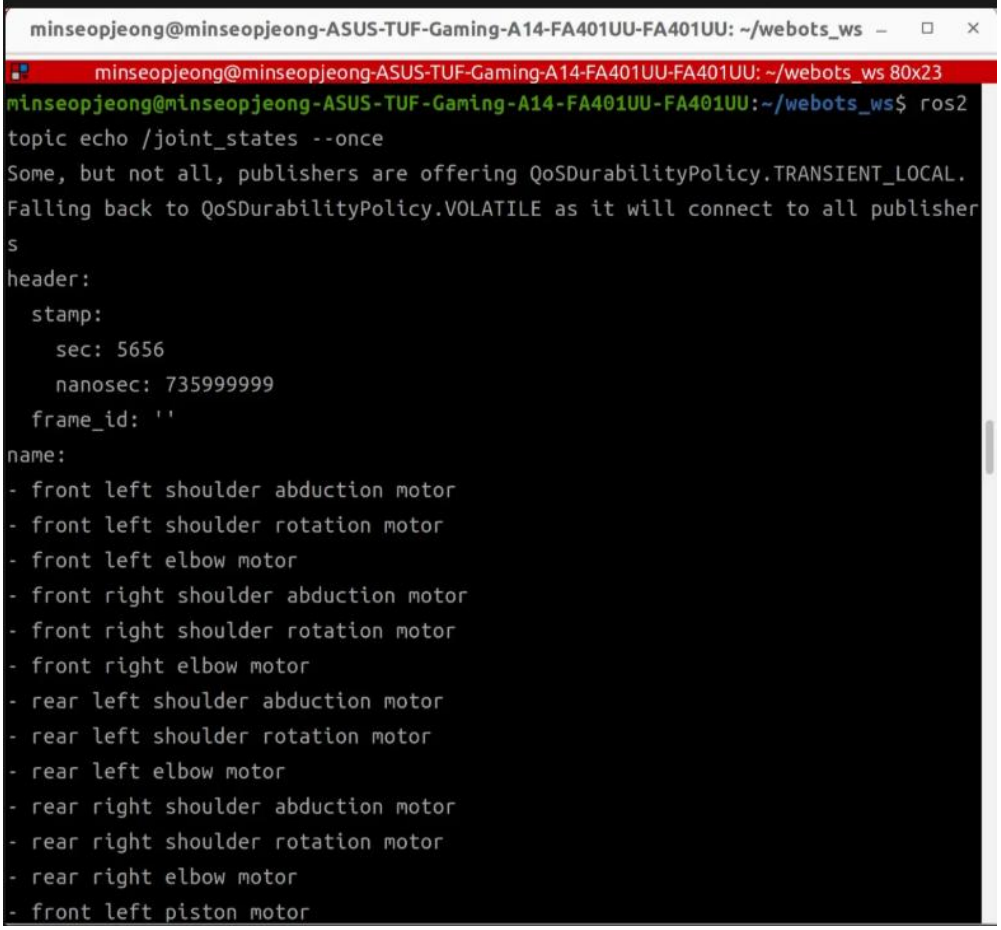


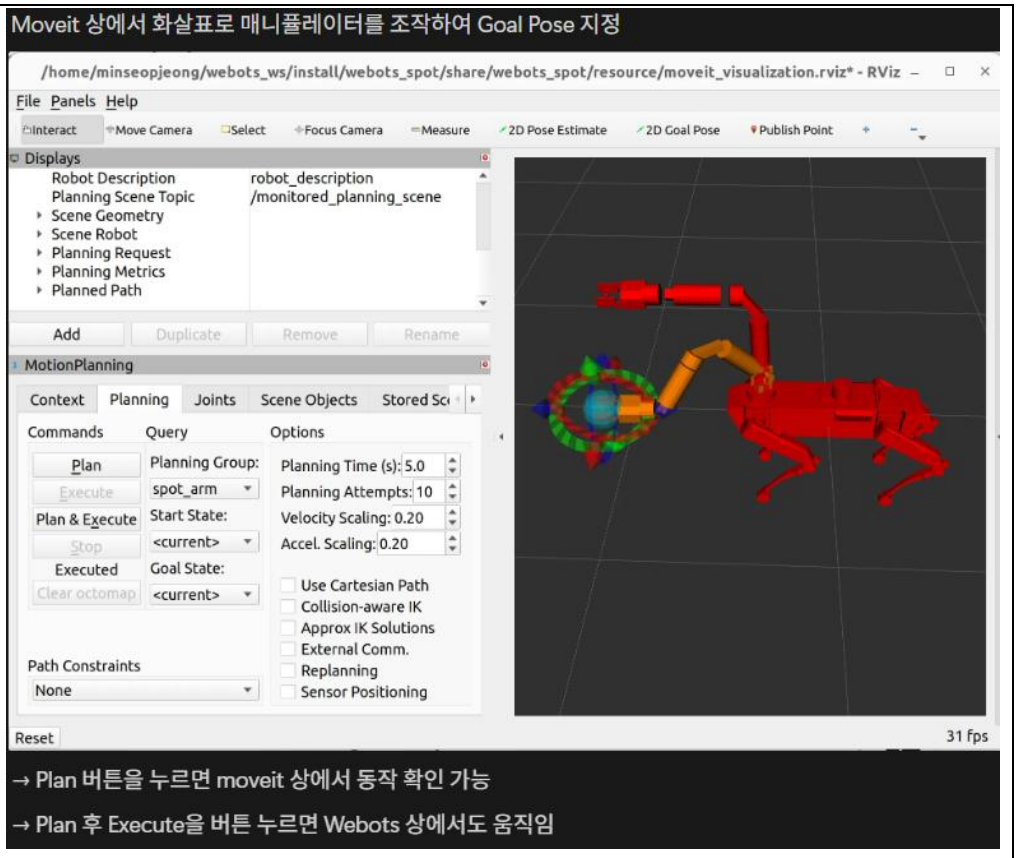
<Fig 11. Spot 의 그리퍼 캠의 image 토픽을 시각화한 모습>

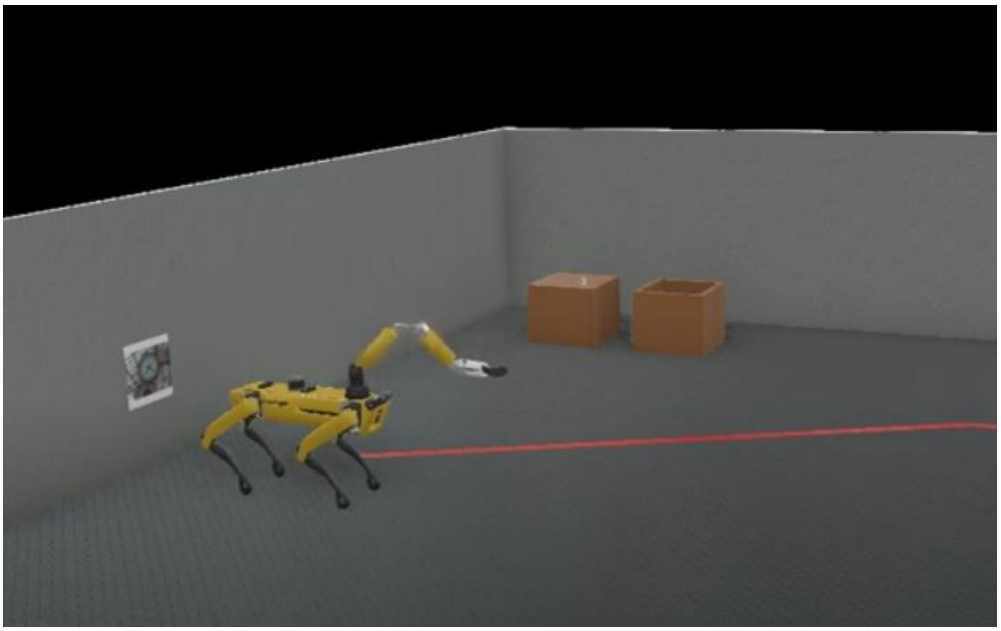
3. MoveIt 을 통한 매니퓰레이터 제어

다음 명령어를 통해 각 Joint state가 제대로 publish되고 있는지 확인할 수 있다.

```
ros2 topic echo /joint_states --once
```

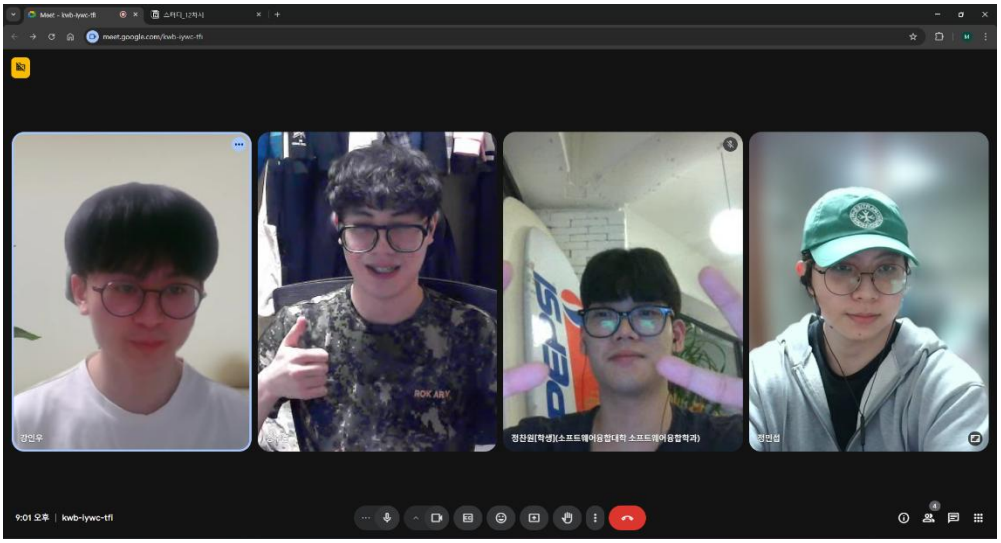
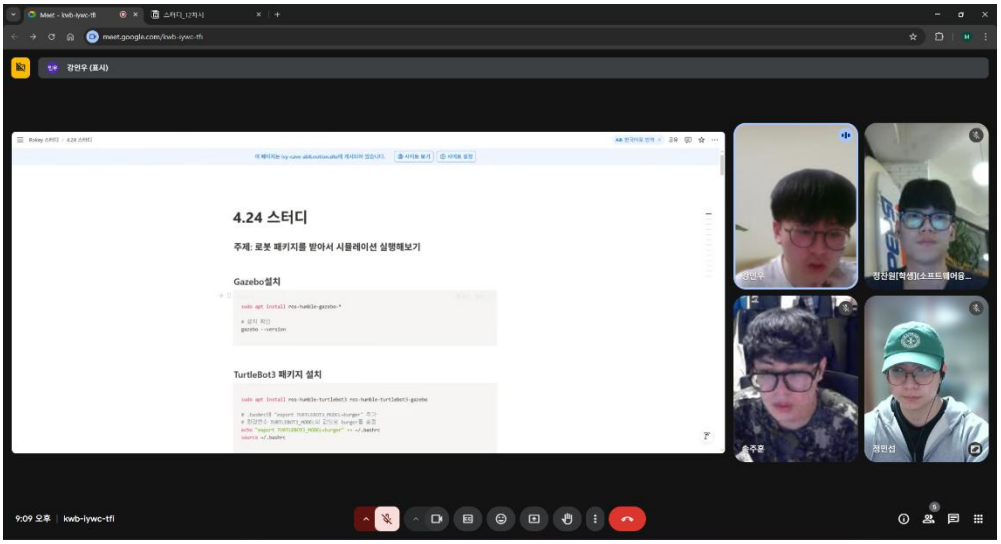






<Fig 12. MoveIt 으로 plan 한 궤적대로 Webots 상에서도 움직인 모습>

활동평가	로봇의 패키지를 직접 시뮬레이션 해보면서 로봇의 joint 와 발행되는 topic 확인하기 등, ROS2 의 이해도를 높일 수 있는 시간이었다.
과제	ROS 평가 대비 수업 내용 복기
향후 계획	ROS 평가 대비 교재 단원 하나 정해서 정리한 후 발표

첨부 자료	 강민우 정찬원[학생](소프트웨어융합대학 소프트웨어융합학과) 정민섭 9:01 오후 | kwb-iywc-tfi  강민우 (표시) 4.24 스터디 주제: 로봇 패키지를 받아서 시뮬레이션 실행해보기 Gazebo설치 TurtleBot3 패키지 설치 강민우 정찬원[학생](소프트웨어융... 손주훈 정민섭 9:09 오후 | kwb-iywc-tfi