

# 매니퓰레이션 툴 MoveIt!과 OpenManipulator

**ROBOTIS**

Open Source Team

Yoonseok Pyo



자~ 이번 목표는  
오픈 매니퓰레이터  
MoveIt! 사용법  
입니다!

# Index

---

I. 로봇 암

II. OpenManipulator

III. MoveIt!을 이용한 모션 플래닝

# Index

---

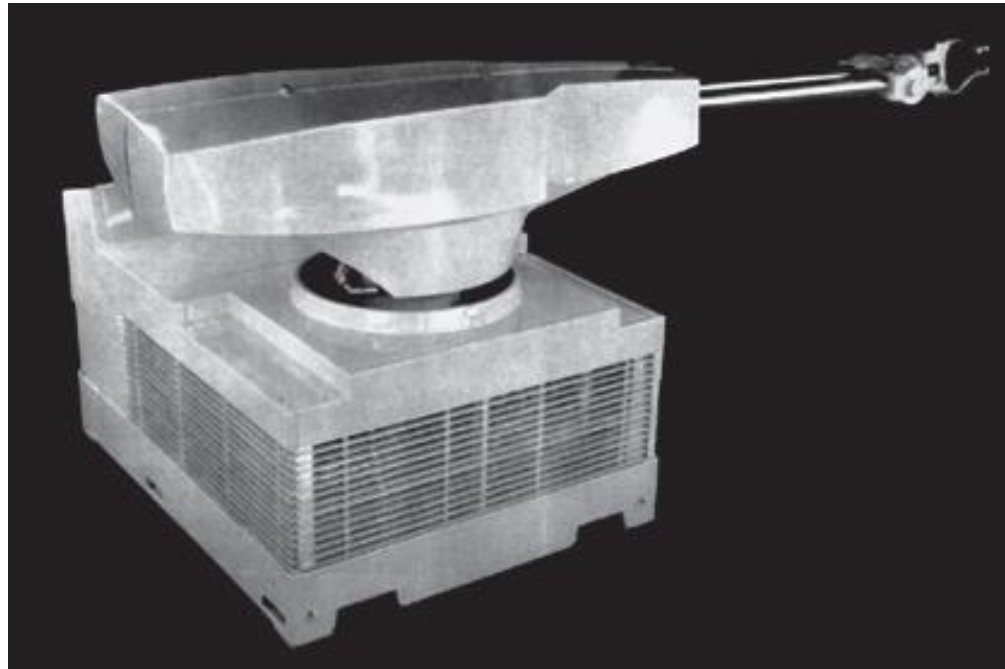
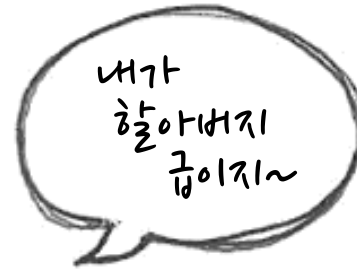
I. 로봇 암

II. OpenManipulator

III. MoveIt!을 이용한 모션 플래닝

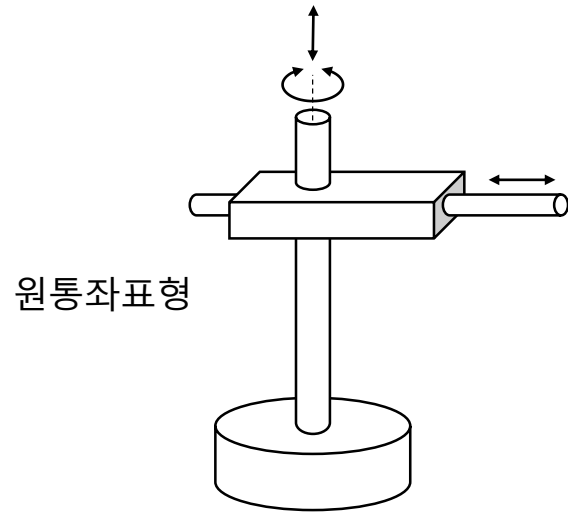
로봇 암?

# 로봇 암(Robot Arm)

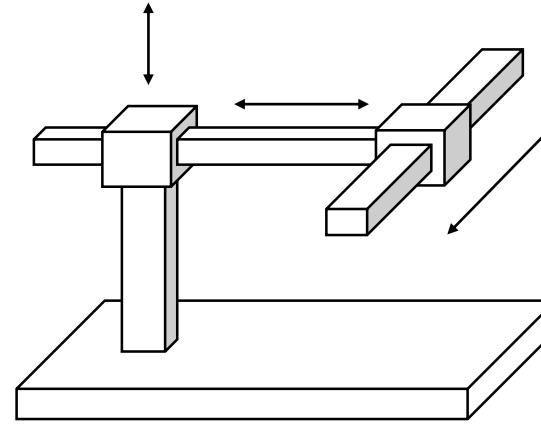


세계 최초 산업용 로봇 유니메이트(unimate), 1961

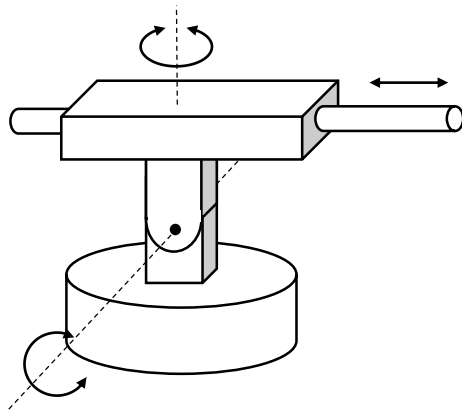
# 로봇 암의 형태별 종류



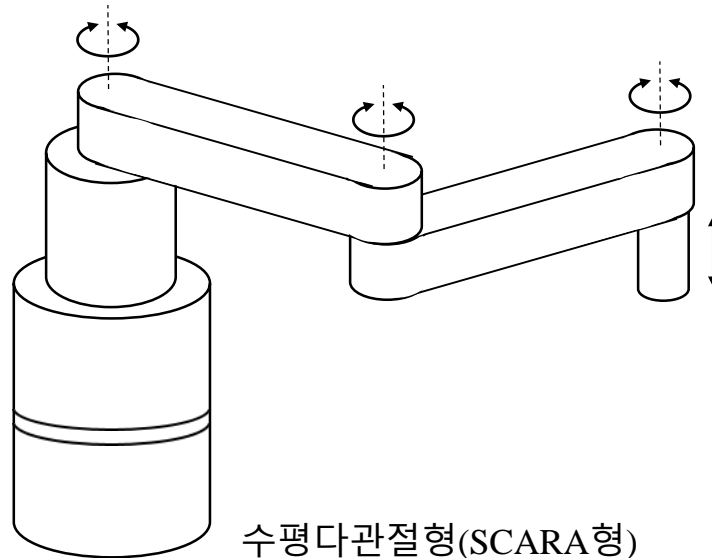
원통좌표형



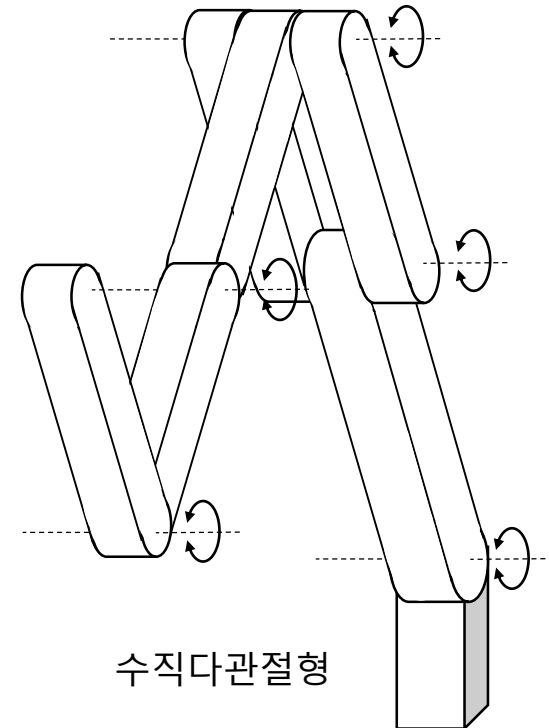
직교좌표형



극좌표형



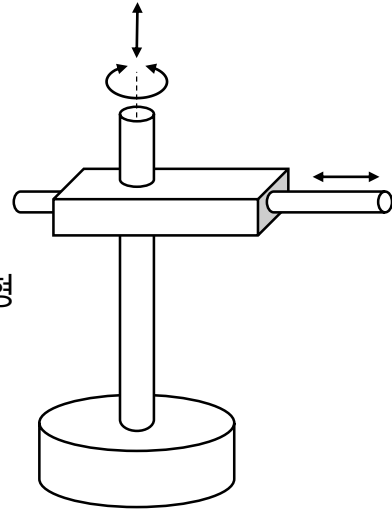
수평다관절형(SCARA형)



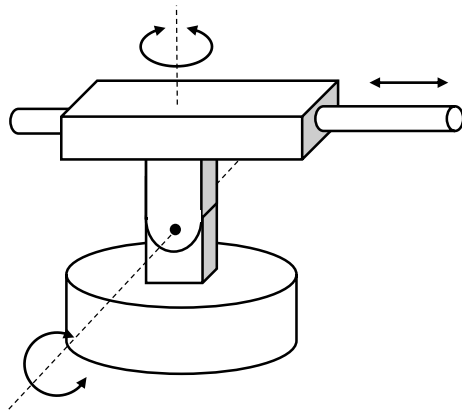
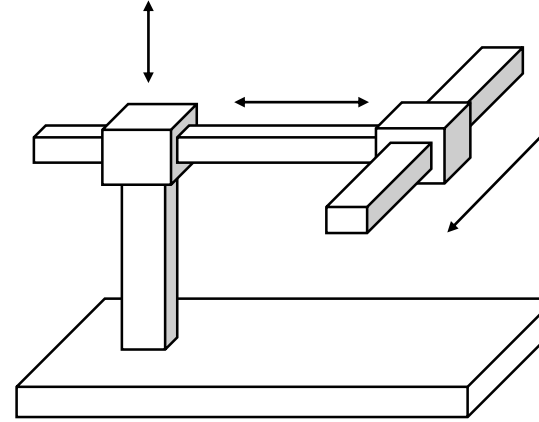
수직다관절형

# 로봇 암의 형태별 종류

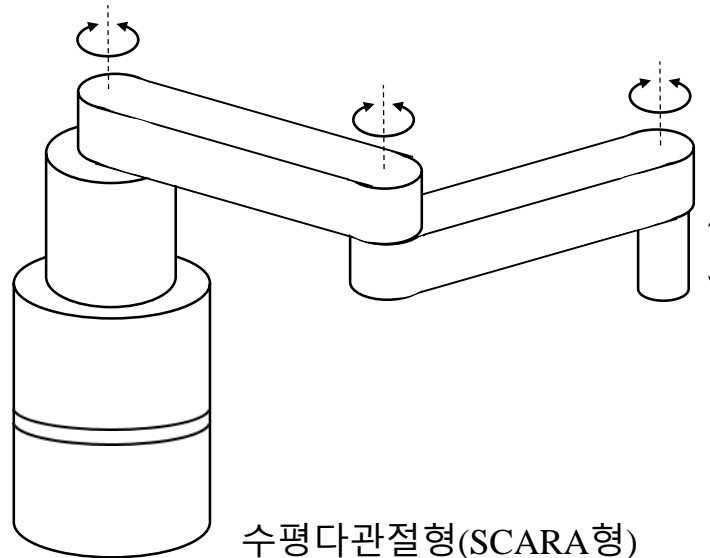
원통좌표형



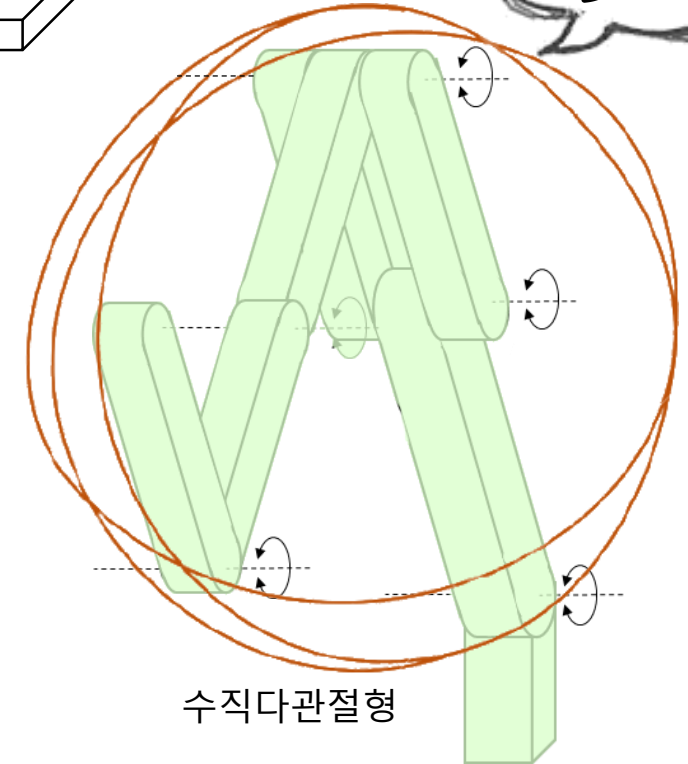
직교좌표형



극좌표형



수평다관절형(SCARA형)



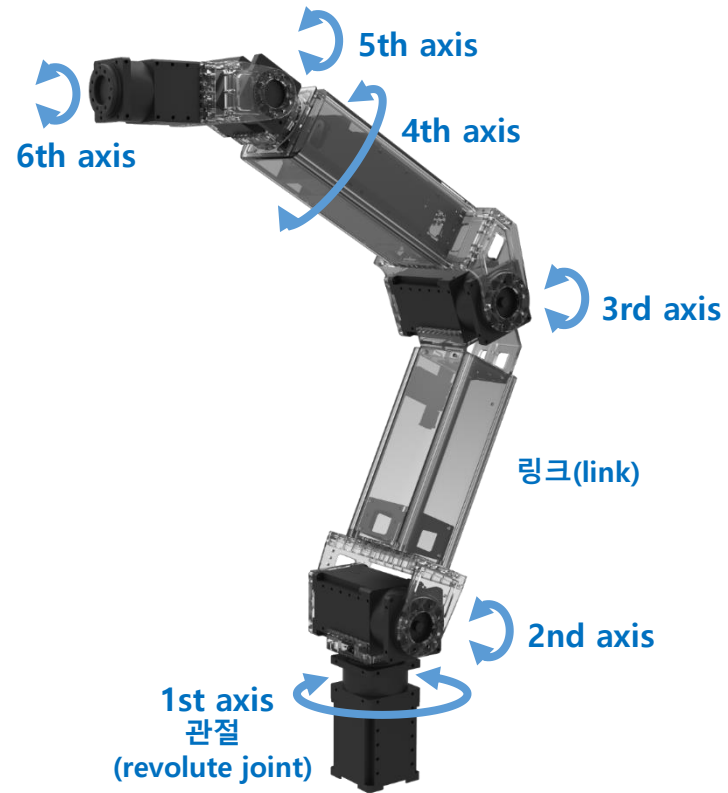
수직다관절형

오늘 세미나에서  
다들 너석



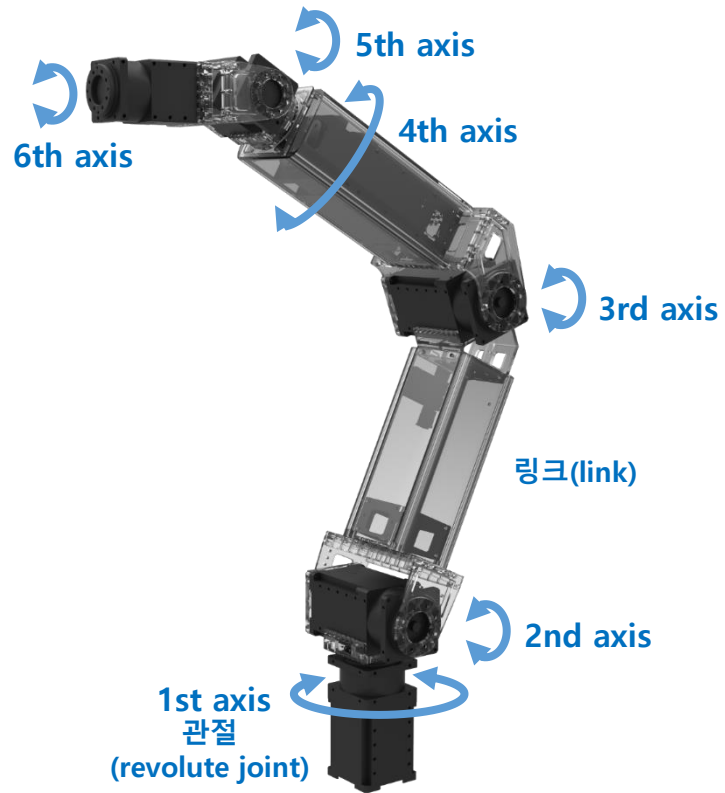
# 로봇 암(Robot Arm)

---

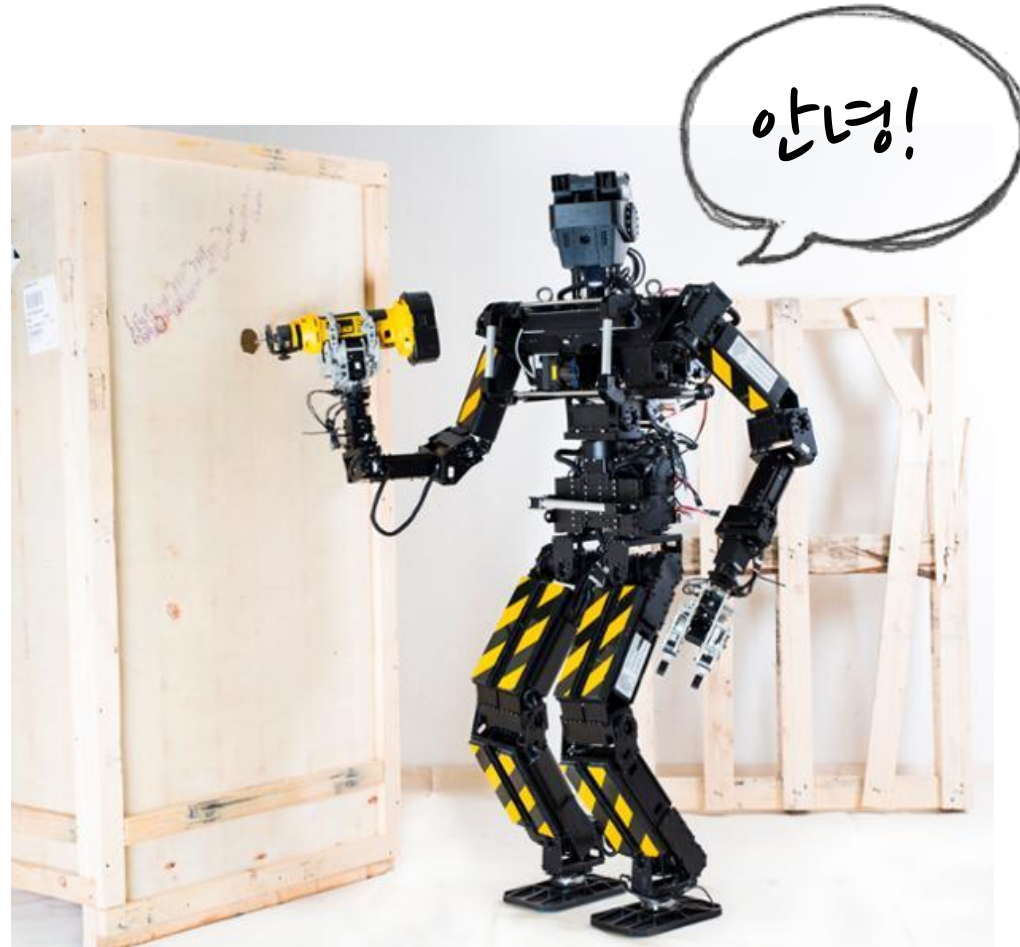


<http://www.robotis.com/>

# 로봇 암(Robot Arm)

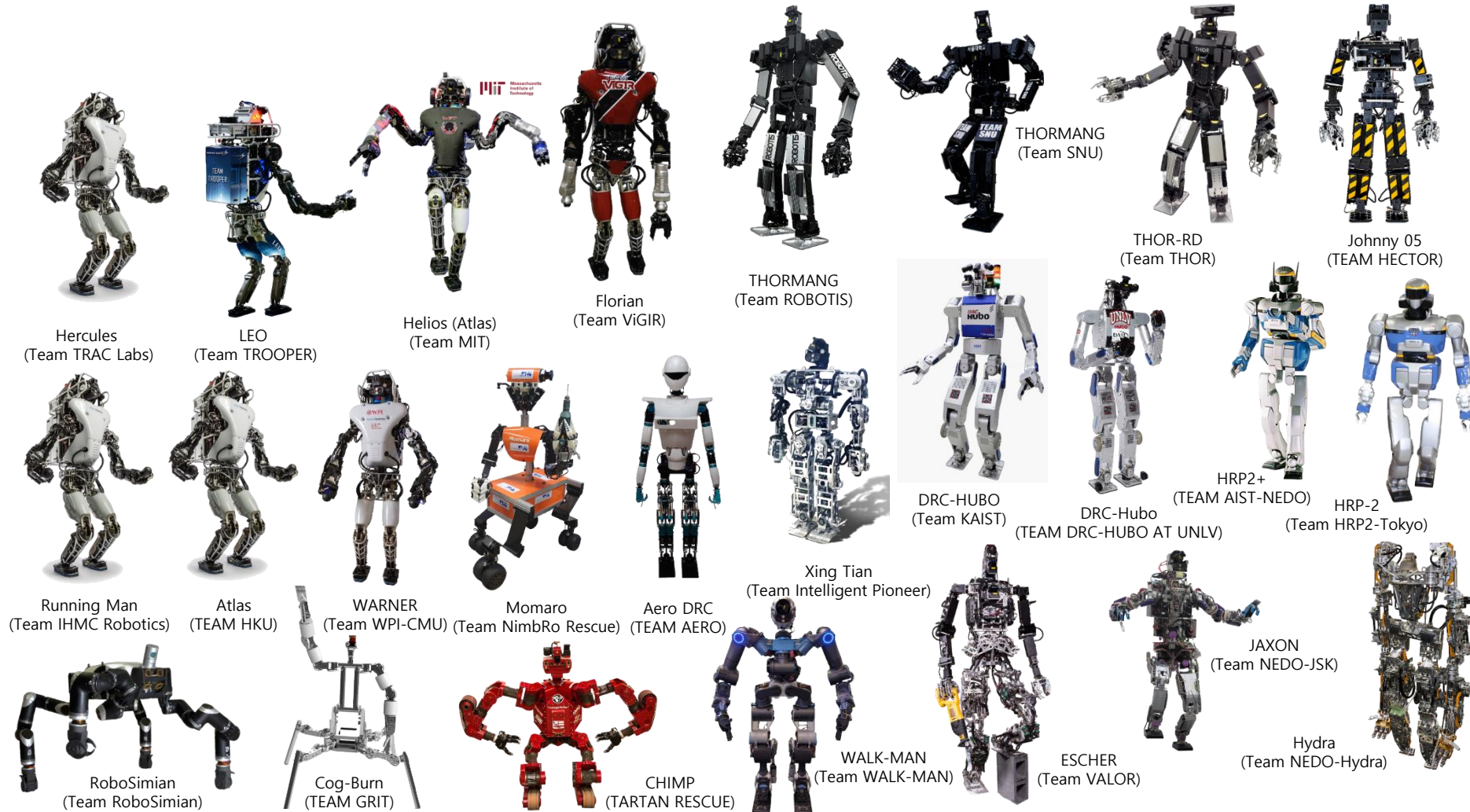


<http://www.robotis.com/>



<http://www.theroboticschallenge.org/>

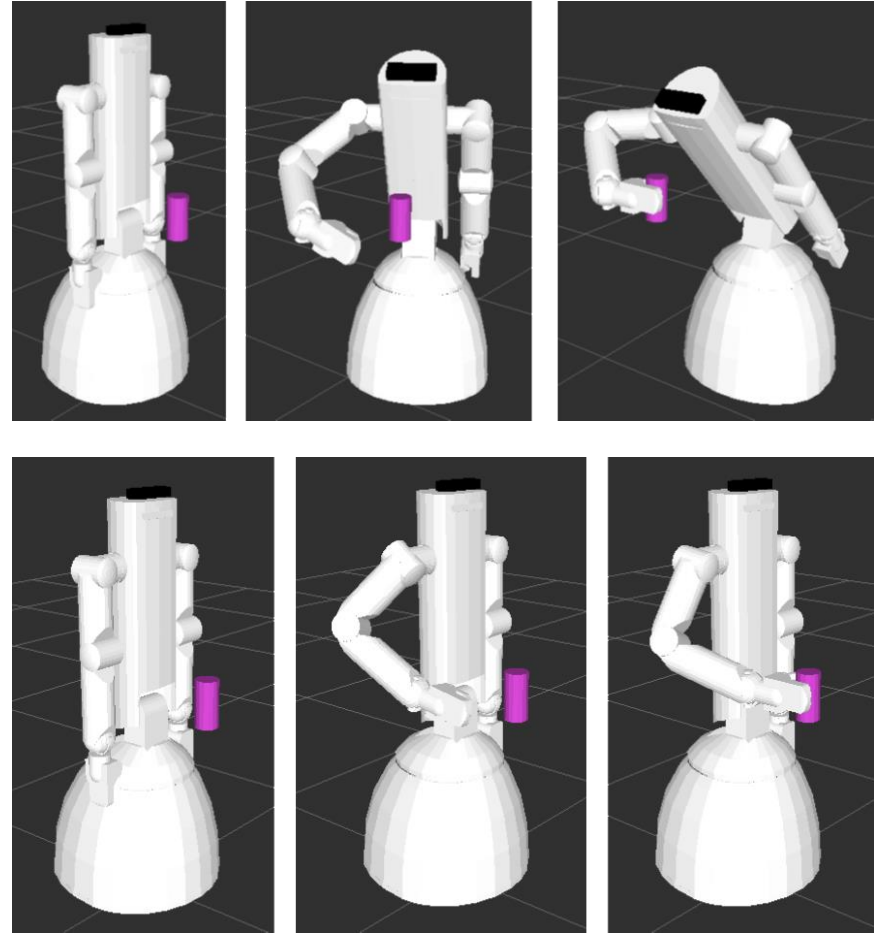
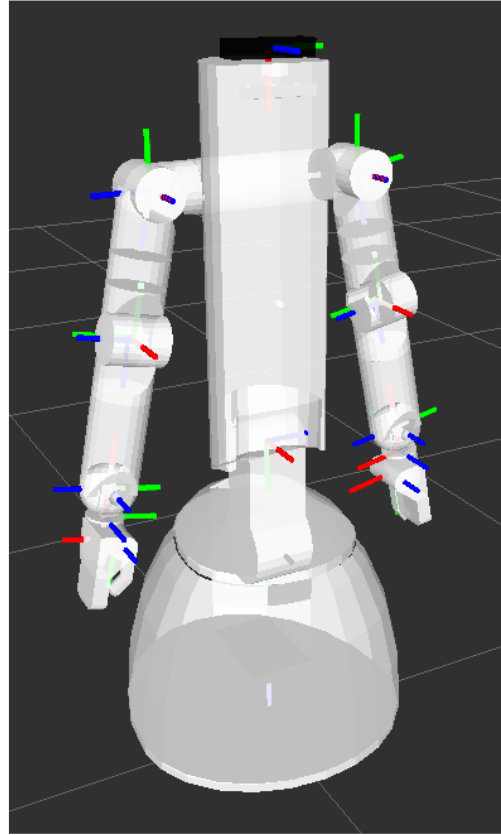
# Manipulator in DRC Finals 25



# 로봇 암 제어



Yaskawa SmartPal V



# 로봇 암 제어

## ■ 센싱(Sensing)

- 위치, 토크
- 주변 사물

## ■ 계획(Plan)

- 수동 / 자동
- 충돌 회피
- 파지(grasping)
- 궤적 생성

## ■ 실행(Action)

- 위치, 속도, 힘 제어

### **Motion Planing for Fetch-and-Give Task using the Manipulability of Robot and Person**

Yoonseok Pyo, Kouhei Nakashima, Tokuo Tsuji,  
Ryo Kurazume and Ken'ichi Morooka

Kyushu University

The 2015 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics

<https://youtu.be/4VtDHWiN0as>

로봇 암 제어?

그리 호락호락한 너석은 **절대** 아님!

로봇 암 제어?

그리 호락호락한 녀석은 **절대** 아님!

근데! 판이 바뀌었다!



등장! \ (^ ^ ) /

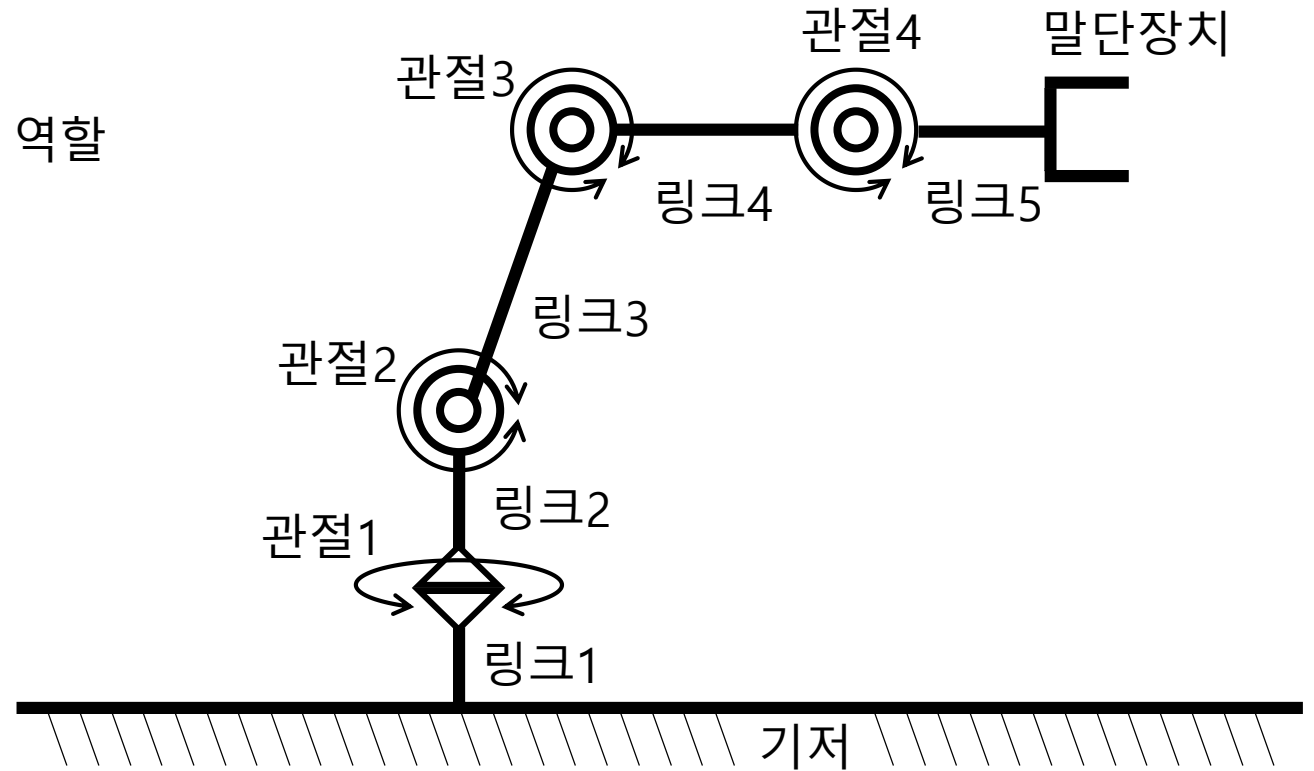


그래도, 기초는 알고 가자!



# 로봇 암의 구성 요소

- 기저(base)
  - 로봇 암이 고정되어 있는 부분
- 링크(link)
  - 기저, 관절, 말단 장치를 연결해주는 강체 역할
- 관절(joint)
  - 로봇의 움직임 을 만들어내는 부분
  - 회전 (revolute) 운동형
  - 병진 (prismatic) 운동형
- 말단 장치(end effector)
  - 사람의 손에 해당하는 부분 또는 장비

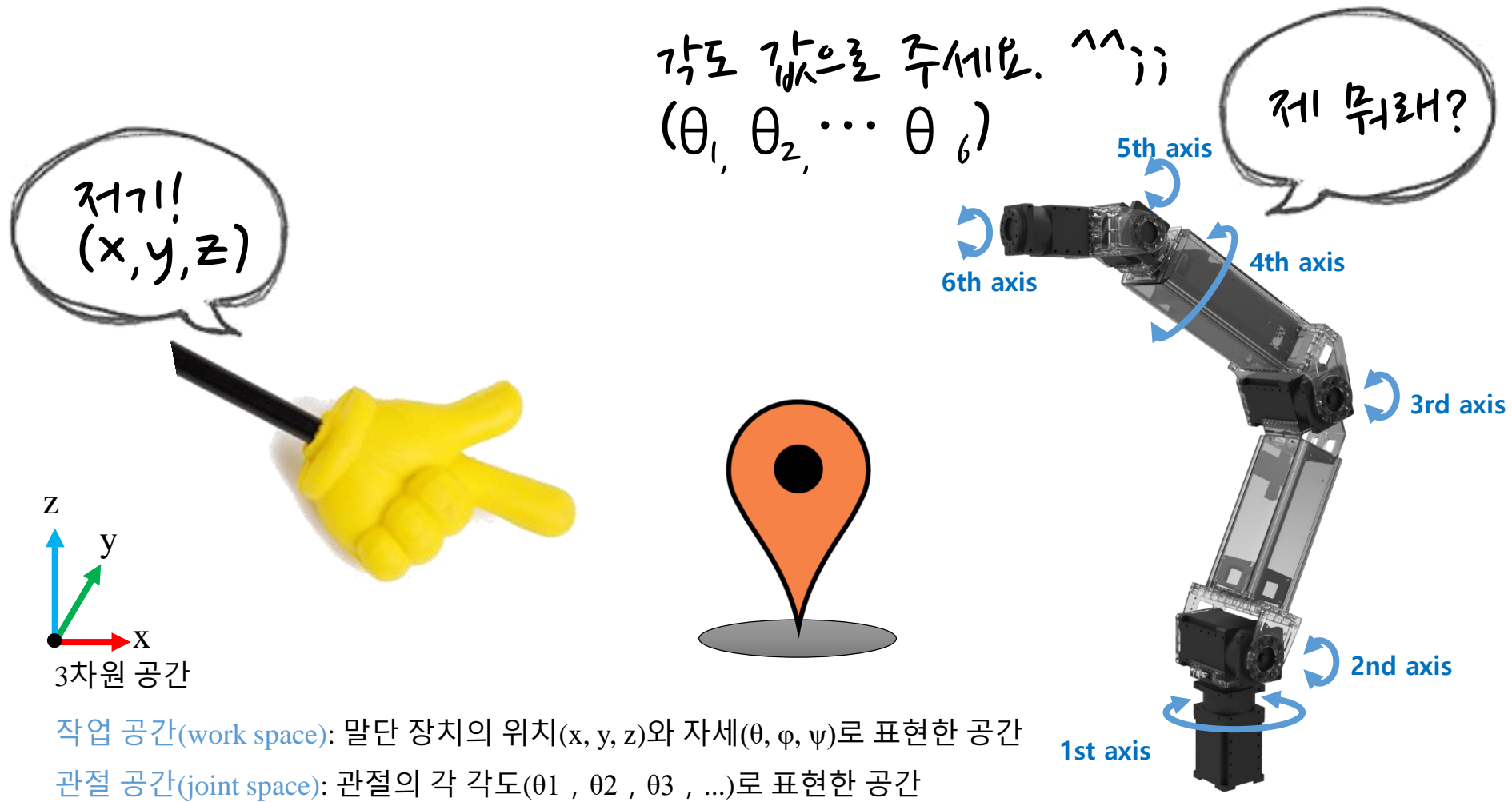


# 작업 공간과 관절 공간



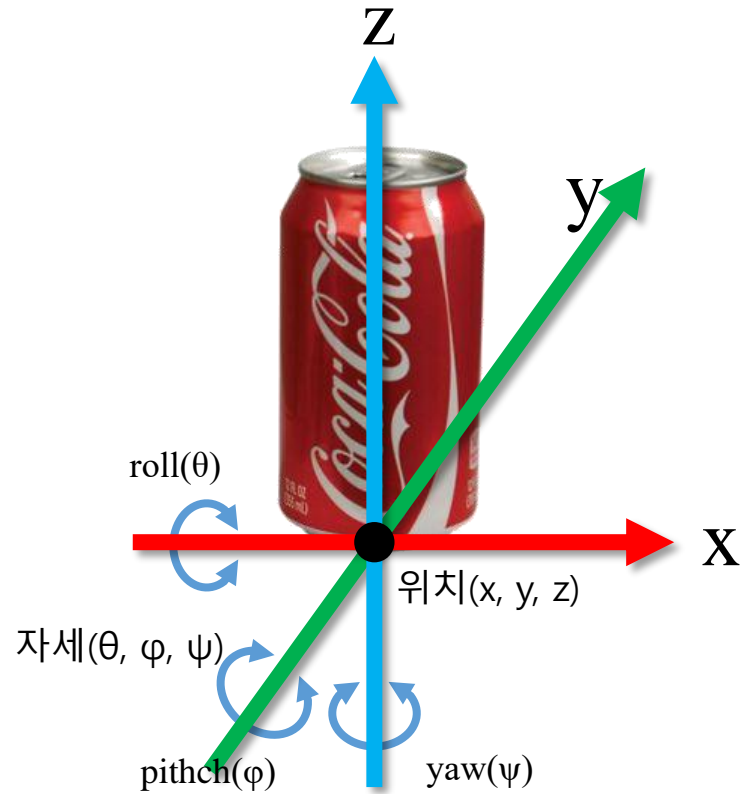
작업 공간(work space): 말단 장치의 위치(x, y, z)와 자세( $\theta$ ,  $\phi$ ,  $\psi$ )로 표현한 공간

# 작업 공간과 관절 공간



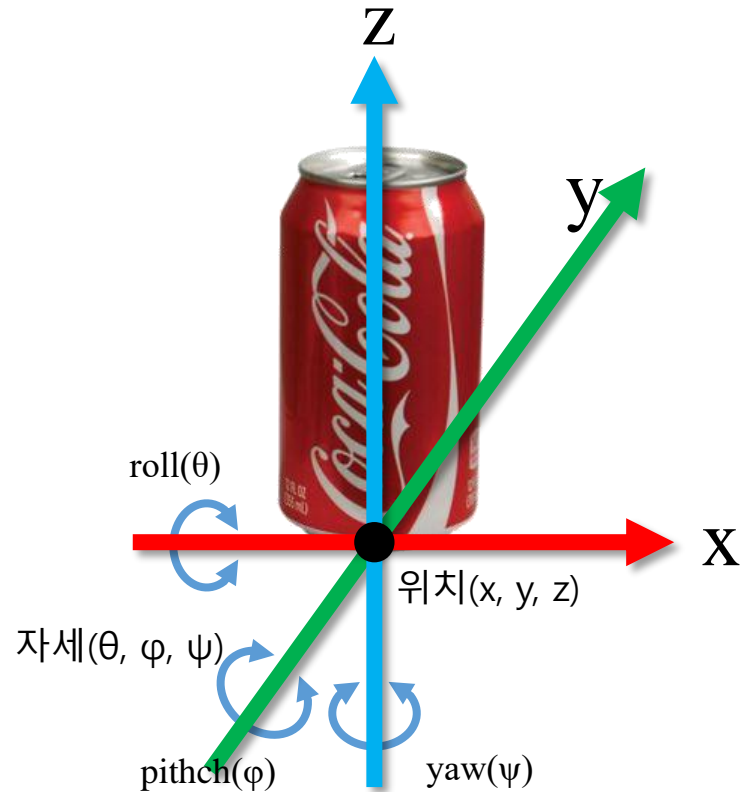
# 자유도(DOF, degrees of freedom)

6개의 변수로 물체의 위치/자세가 표현 가능  
= 6자유도로 표현가능

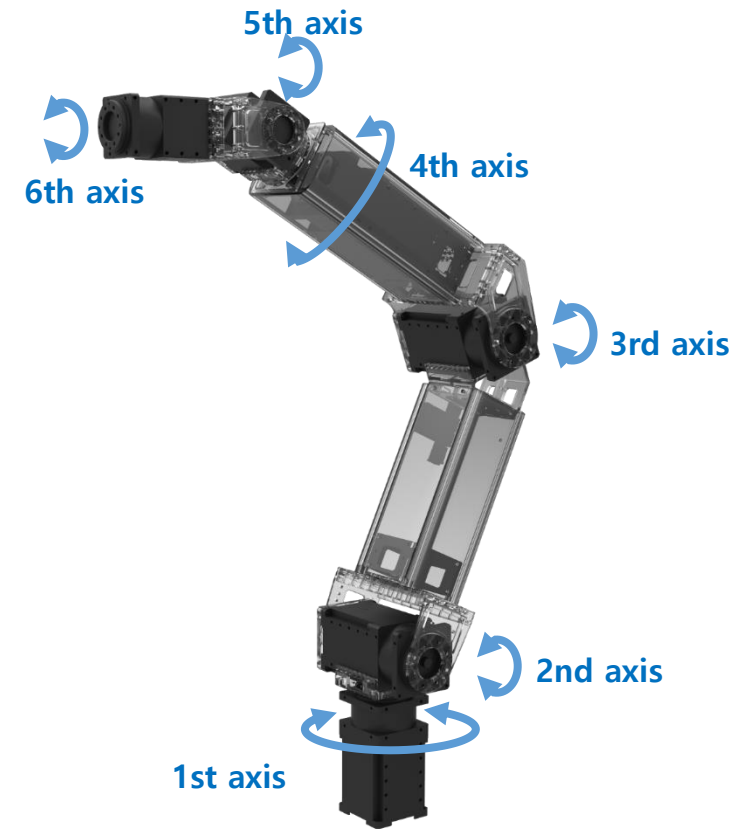


# 자유도(DOF, degrees of freedom)

6개의 변수로 물체의 위치/자세가 표현 가능  
= 6자유도로 표현가능



6개의 변수로 로봇의 말단 장치의 위치, 자세가 결정  
= 6자유도를 갖는다



# 순운동학과 역운동학

---

- 로봇을 목적대로 동작시키기 위해서 필요한 것은?
  - 기하학적인 운동계획을 세우고 (Kinematics, 운동학 또는 기구학이라고도 함) ,
  - 동력학적 효과를 계산하여(Dynamics, 동력학),
  - 각 관절에게 적당한 명령 값을 내리는 것

# 순운동학과 역운동학

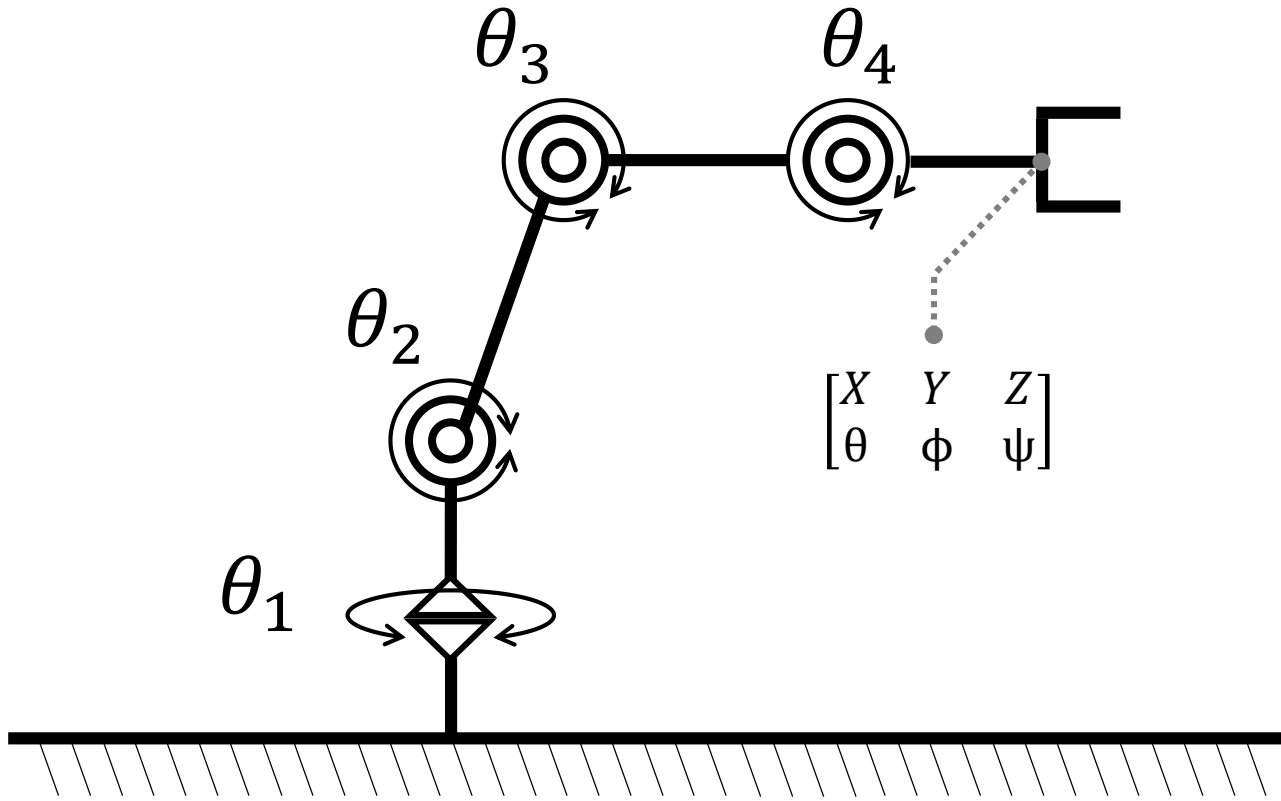
---

- 로봇을 목적대로 동작시키기 위해서 필요한 것은?
  - 기하학적인 운동계획을 세우고 (Kinematics, 운동학 또는 기구학이라고도 함) ,
  - 동력학적 효과를 계산하여(Dynamics, 동력학),
  - 각 관절에게 적당한 명령 값을 내리는 것
- 순운동학(FK, Forward Kinematics)
  - 로봇의 각 관절 각도가 주어졌을 때 말단 장치의 위치 및 자세를 구하는 것
  - $(\theta_1, \theta_2, \theta_3, \dots, \theta_n) \rightarrow (x, y, z), (\theta, \phi, \psi)$
- 역운동학(IK, Inverse Kinematics)
  - 말단 장치의 위치와 자세가 주어졌을 때 각 관절 각도를 구하는 것
  - $(x, y, z), (\theta, \phi, \psi) \rightarrow (\theta_1, \theta_2, \theta_3, \dots, \theta_n)$

# 순운동학과 역운동학

- 순운동학(FK, Forward Kinematics)

- 로봇의 각 관절 각도가 주어졌을 때 말단 장치의 위치 및 자세를 구하는 것
- $(\theta_1, \theta_2, \theta_3, \dots, \theta_n) \rightarrow (x, y, z), (\theta, \phi, \psi)$

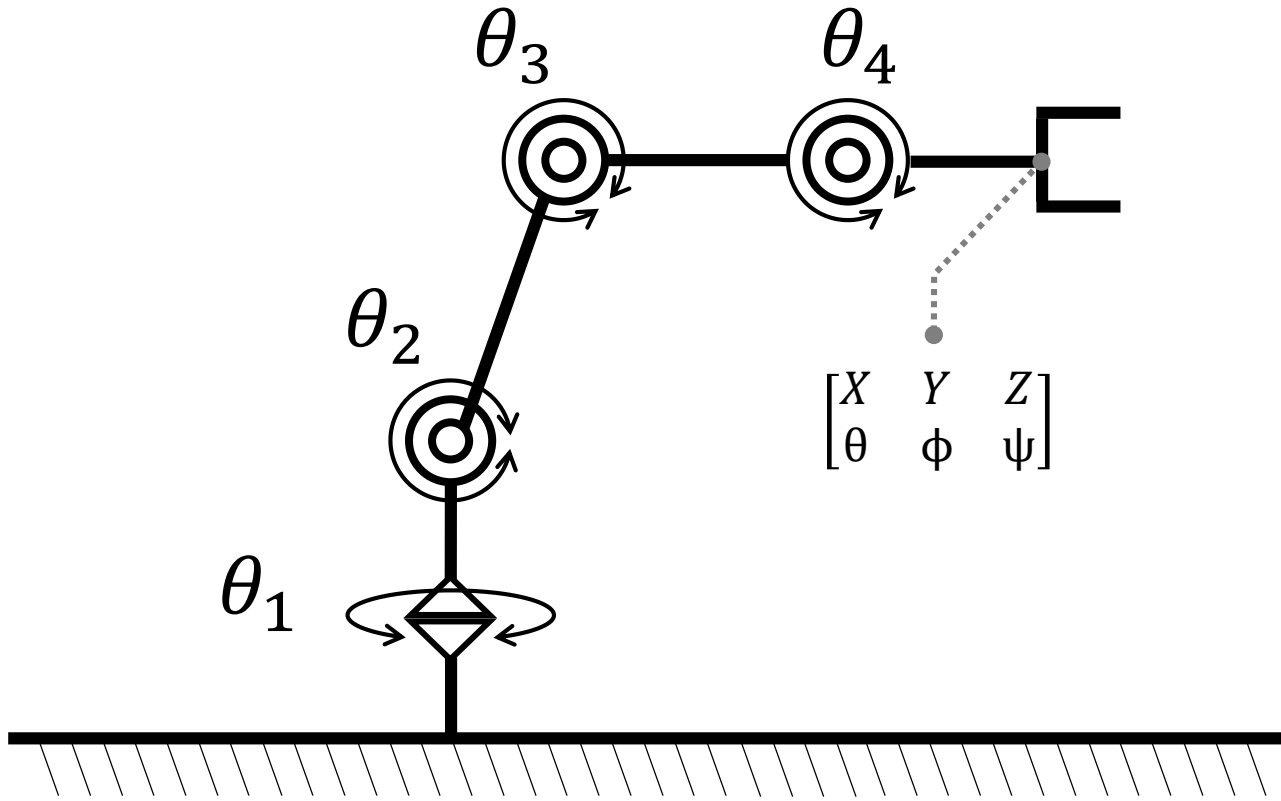


$$\begin{matrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{matrix} \rightarrow \begin{bmatrix} X & Y & Z \\ \theta & \phi & \psi \end{bmatrix}$$



# 순운동학과 역운동학

- 역운동학(IK, Inverse Kinematics)
  - 말단 장치의 위치와 자세가 주어졌을 때 각 관절 각도를 구하는 것
  - $(x, y, z), (\theta, \phi, \psi) \rightarrow (\theta_1, \theta_2, \theta_3, \dots, \theta_n)$



$$\begin{bmatrix} X & Y & Z \\ \theta & \phi & \psi \end{bmatrix} \rightarrow \begin{matrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{matrix}$$

# Index

---

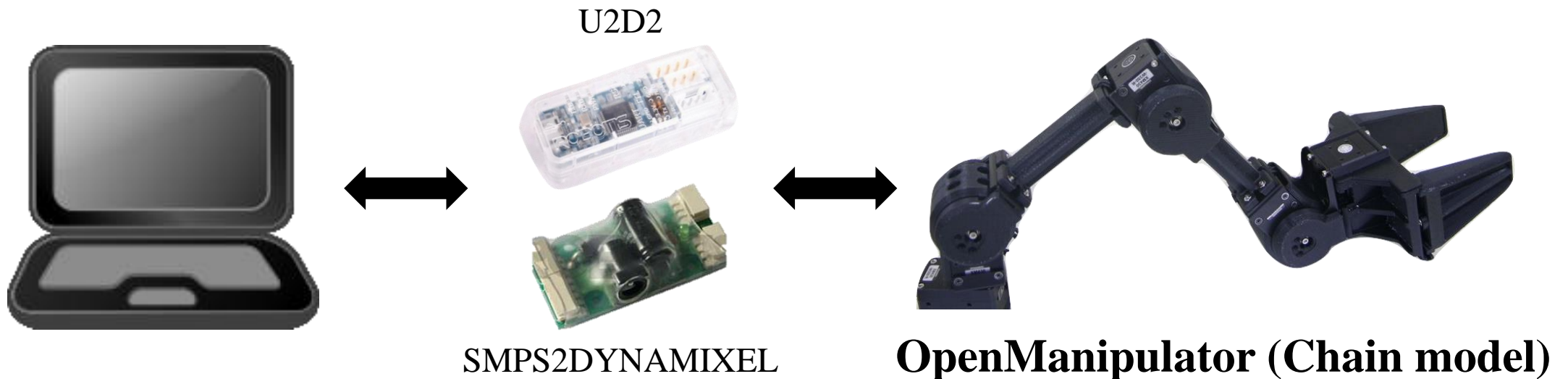
I. 로봇 암

II. OpenManipulator

III. MoveIt!을 이용한 모션 플래닝

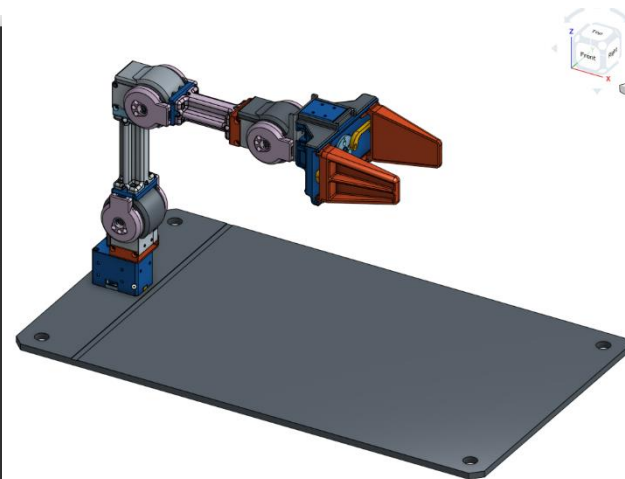
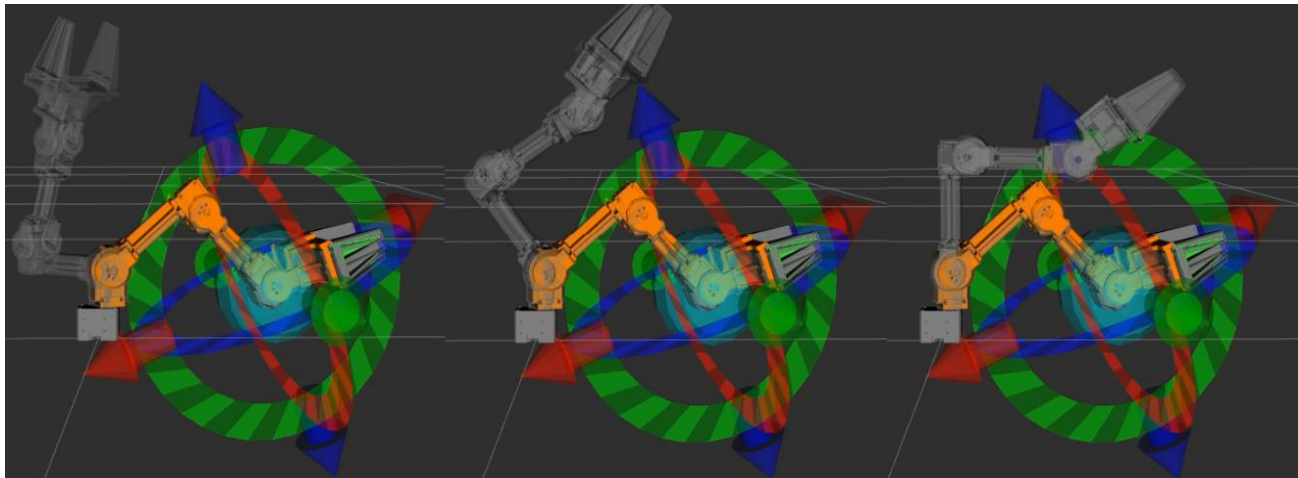
# OpenManipulator

- 로봇티즈에서 개발된 **오픈 소스 S/W, H/W 기반**의 매니퓰레이터
  - Dynamixel X 시리즈**를 지원, 다양한 스펙의 액추에이터를 골라 로봇 제작 가능
  - 수직다관절형, SCARA형, Link형, Delta 등의 **3D 디자인 파일 예제 제공**
  - ROS**를 지원함과 동시에 매니퓰레이션 툴 **MoveIt!** 예제 제공
  - OpenCR**과 **Arduino IDE, Processing**도 함께 지원
  - TurtleBot3**과 함께 사용 가능하여 **모바일 매니퓰레이터**로 사용 가능



# OpenManipulator

- 위키 페이지: [https://github.com/ROBOTIS-GIT/open\\_manipulator/wiki](https://github.com/ROBOTIS-GIT/open_manipulator/wiki)
- 공개 소프트웨어: [https://github.com/ROBOTIS-GIT/open\\_manipulator](https://github.com/ROBOTIS-GIT/open_manipulator)
- 공개 하드웨어
  - TurtleBot3 + OpenManipulator Chain: <https://goo.gl/xgxxh2G>
  - OpenManipulator Chain: <https://goo.gl/NsqJMu>
  - OpenManipulator SCARA: <https://goo.gl/oVsK8m>



# Index

---

I. 로봇 암

II. OpenManipulator

III. MoveIt!을 이용한 모션 플래닝

# 매니퓰레이터를 위한 ROS의 유용한 툴

---

- **로봇 모델링**을 위한 시각화 툴로서 **URDF** (Unified Robot Description Format)파일을 XML(Extensive Markup Language)을 통해 손쉽게 제작 가능
- 실제 동작 환경을 꾸밀 수 있는 **3D 시뮬레이터 Gazebo**를 지원하기 때문에 Gazebo의 여러 가지 센서와 로봇의 제어를 도와주는 plugin기능과 ROS-CONTROL을 사용 가능
- 매니퓰레이터를 위한 통합 라이브러리인 **MoveIt!** 사용 가능
  - Kinematics and Dynamics Library(KDL)와 The Open Motion Planning Library(OMPL) 등의 오픈 라이브러리를 제공
  - 충돌 계산, 모션 플래닝 그리고 Pick and Place 데모 등과 같이 매니퓰레이터의 여러 가지 기능 쉽게 사용 가능

고생들 하셨습니다!

고생들 하셨습니다!

좀 쉬게 갑시다.

마니퓰레이션 툴





# Movelt! (매니퓰레이터를 위한 통합 라이브러리)

---



<https://youtu.be/0og1SaZYtRc>

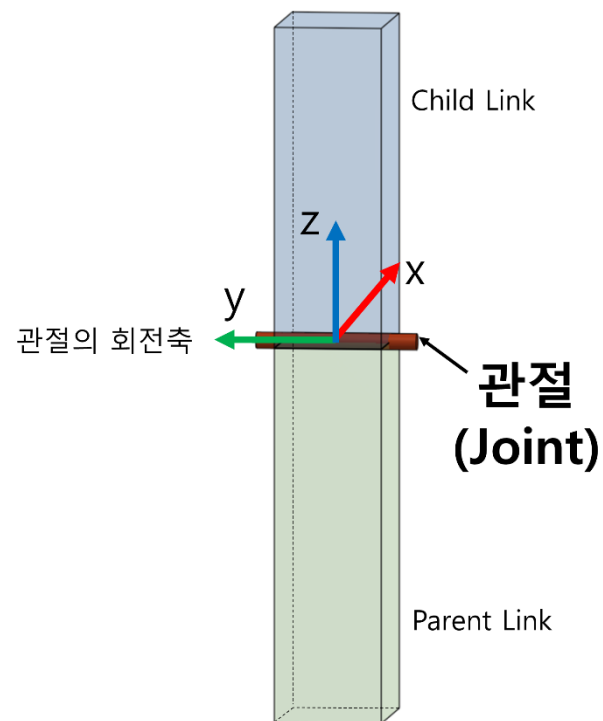
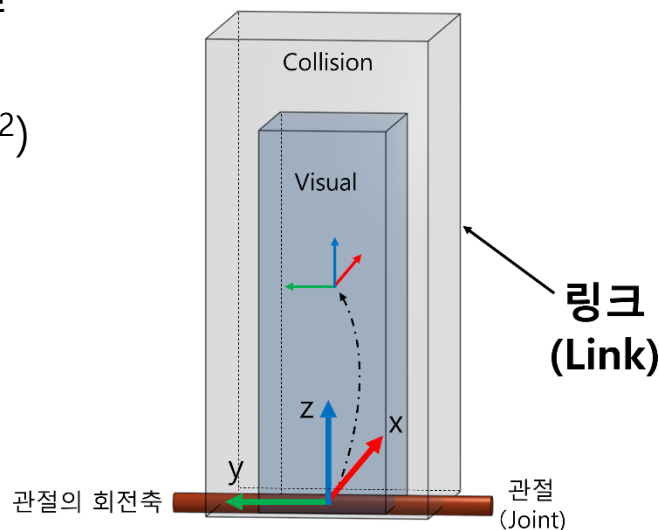
# Movelt! 에 넘겨줘야 할 정보

## ■ 링크(link) 정보

- 기하학적 정보
- STL 또는 DAE(COLLDA) 3차원 설계 파일
- 무게(mass, Kg)
- 관성 모멘트(moments of inertia,  $\text{Kg} \times \text{m}^2$ )

## ■ 관절(joint) 정보

- 전/후 링크와의 관계
- 운동 종류(회전 운동, 병진 운동)
- 회전 및 병진 운동의 기준 축
- 한계 값(관절에 부여되는 힘, 최대/최소 관절 값, 속도)



# MoveIt! 에 넘겨줘야 할 정보

## ■ 링크(link) 정보

- 기하학적 정보
- STL 또는 DAE(COLLDA) 3차원 설계 파일
- 무게(mass, Kg)
- 관성 모멘트(moments of inertia,  $\text{Kg} \times \text{m}^2$ )

## ■ 관절(joint) 정보

- 전/후 링크와의 관계
- 운동 종류(회전 운동, 병진 운동)
- 회전 및 병진 운동의 기준 축
- 한계 값(관절에 부여되는 힘, 최대/최소 관절 값, 속도)

즉,  
→ URDF  
하나면 된다!

URDF(Universal Robot Description Format)

→ RViz

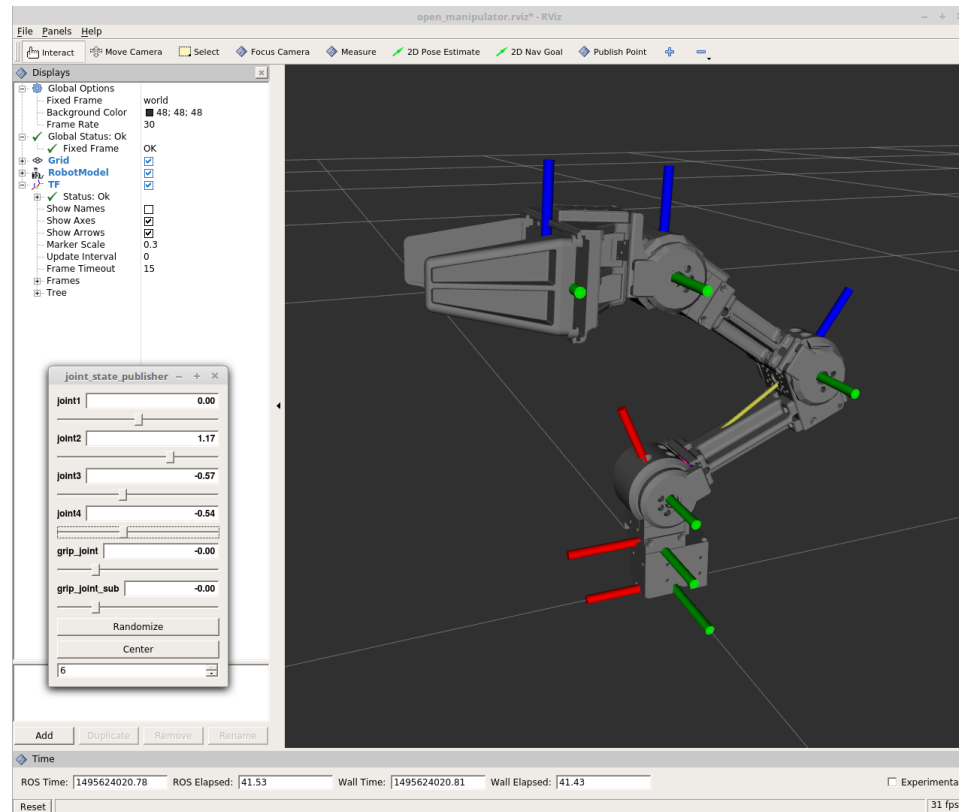
SRDF(Semantic Robot Description Format)

→ MoveIt!

SDF(Simulation Description Format)

→ Gazebo

URDF를 이용한 모델링 작성은  
시간 관계상 이번 세미나에서는 생략 ^^;



시간  
조급

“OpenManipulator  
with Gazebo and MoveIt!”

따라 해 보시다!

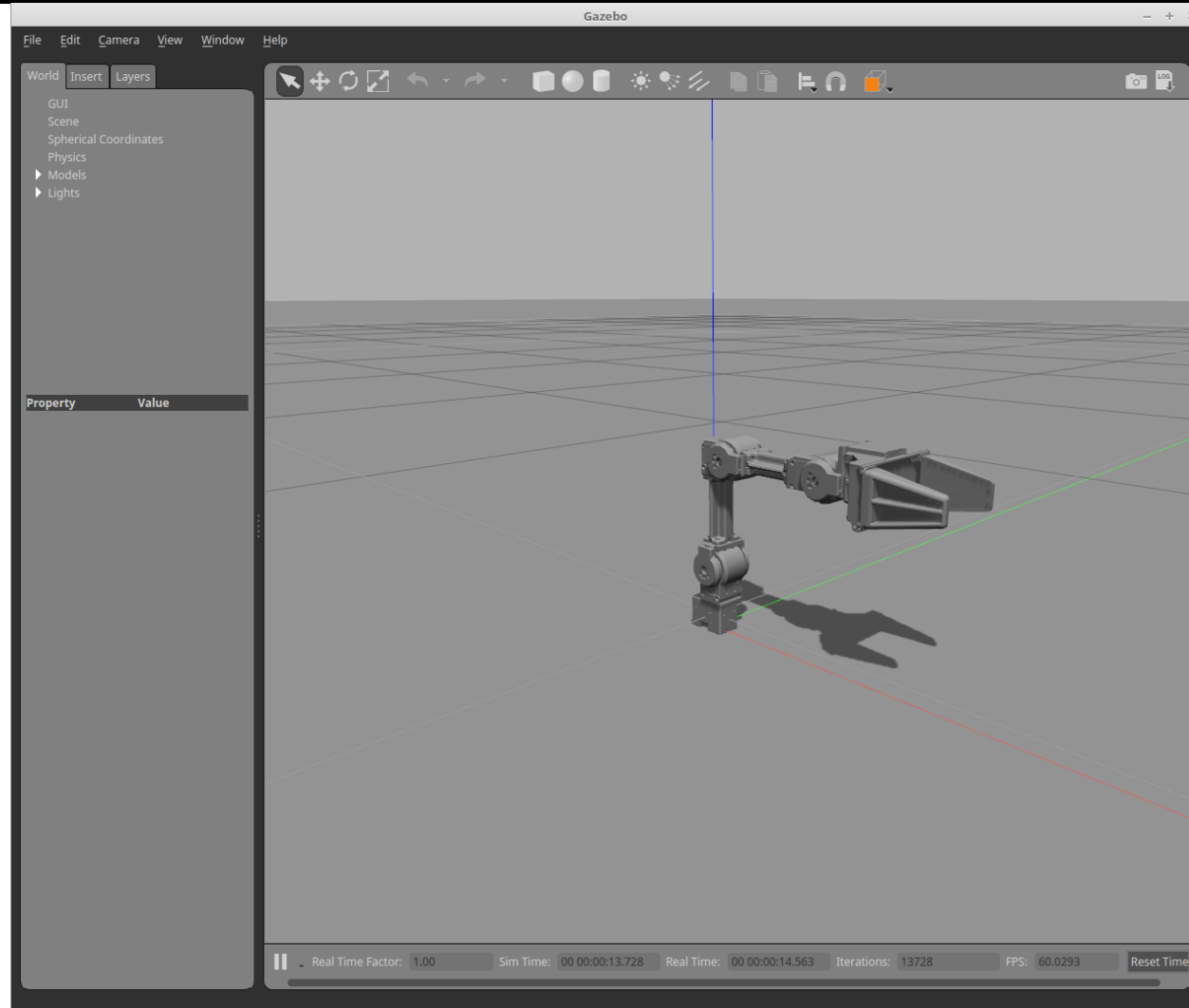
# OpenManipulator 및 MoveIt! 의존성 패키지 설치

```
$ sudo apt-get install ros-kinetic-desktop-full  
$ sudo apt-get install ros-kinetic-rqt*  
$ sudo apt-get install ros-kinetic-moveit*  
$ sudo apt-get install ros-kinetic-industrial-core  
$ sudo apt-get install ros-kinetic-dynixel-sdk  
$ sudo apt-get install ros-kinetic-dynixel-workbench-toolbox  
$ sudo apt-get install ros-kinetic-robotis-math
```

```
$ cd ~/catkin_ws/src  
$ git clone https://github.com/ROBOTIS-GIT/open_manipulator.git  
$ cd ~/catkin_ws && catkin_make
```

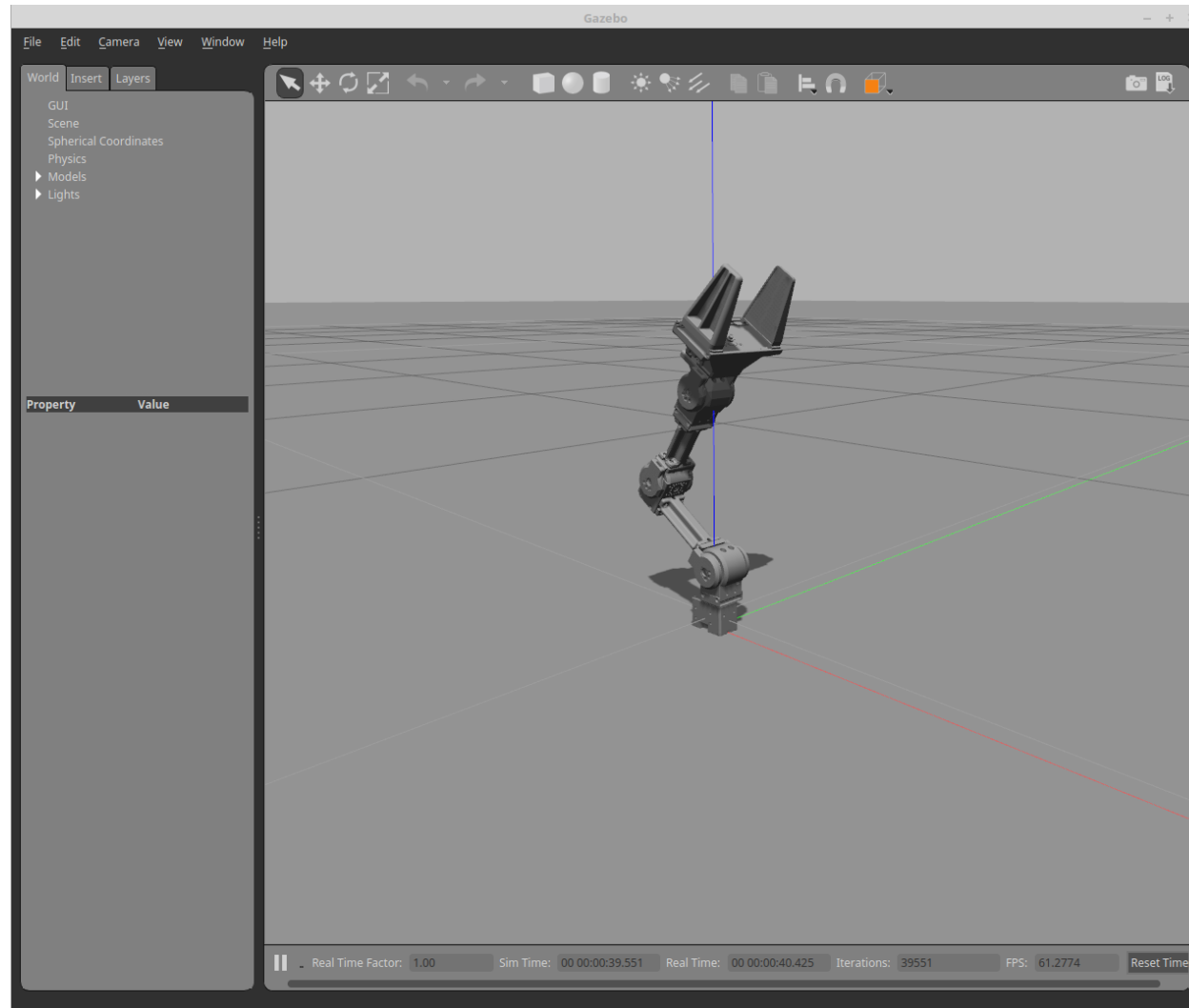
# Gazebo에서 OpenManipulator 실행하기

```
$ roslaunch open_manipulator_gazebo open_manipulator_gazebo.launch
```



# Gazebo에서 OpenManipulator 실행하기

```
$ rostopic pub /open_manipulator_chain/joint2_position/command std_msgs/Float64 "data: 1.0" --once
```





# Movelt!

- 센싱(Sensing)

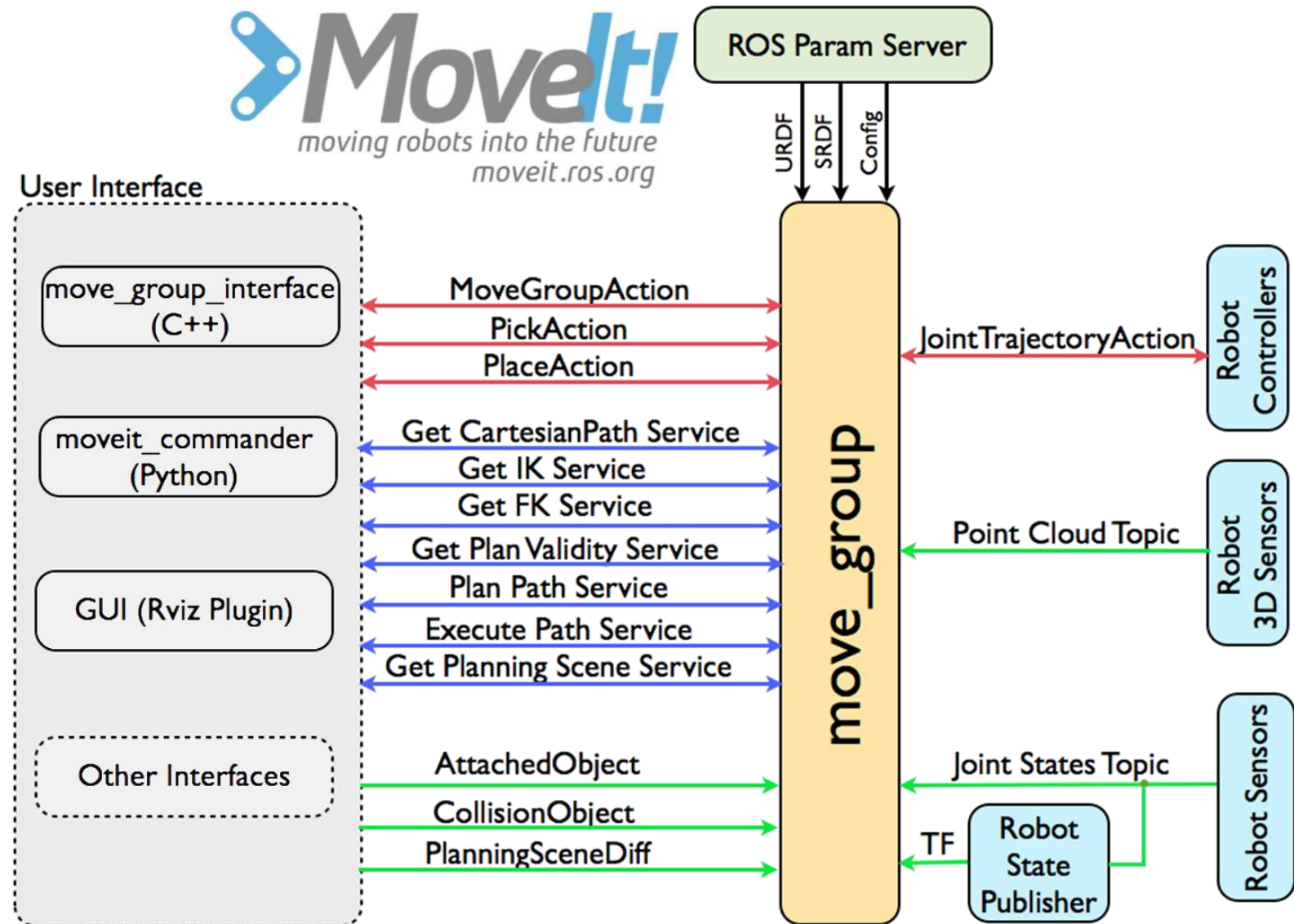
- 관절 위치
- TF
- 3D sensing

- 계획(Plan)

- FK, IK
- Grasping
- 충돌 회피
- 모션 계획

- 실행(Action)

- 위치/속도 제어
- Pick and Place



# Movelt!

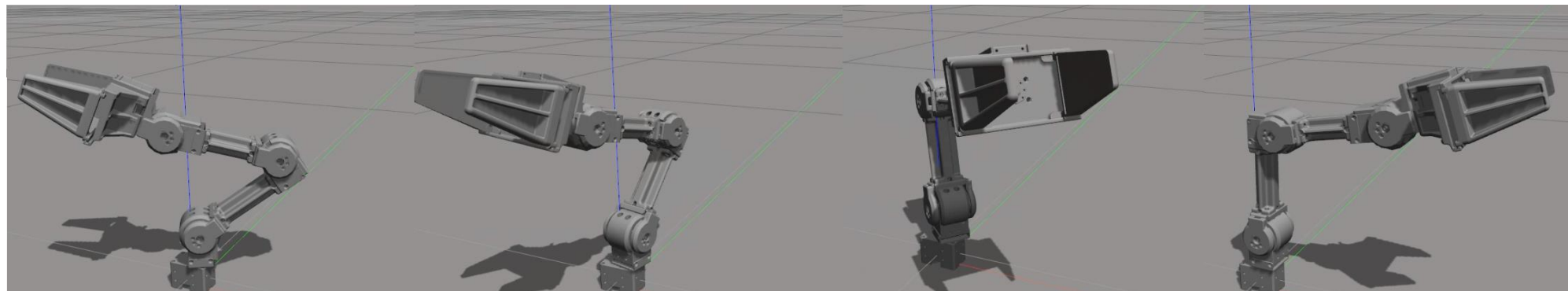
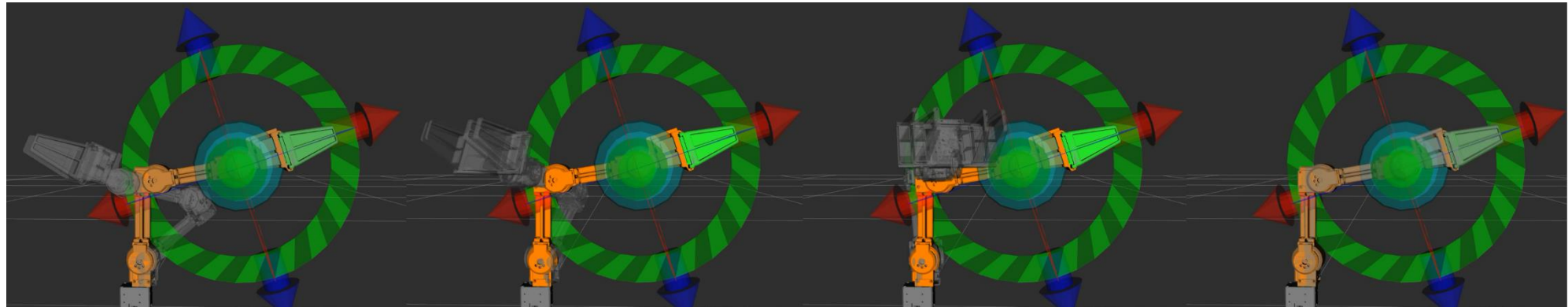
---

- 매니퓰레이터를 위한 통합 라이브러리: 매니퓰레이션 툴
- 모션 플래닝을 위한 빠른 역기구학 해석, 매니퓰레이션을 위한 고급 알고리즘, 로봇 핸드 제어, 동역학, 제어기 그리고 모션 플래닝 등 다양한 기능 제공
- 로봇 정보는 URDF 및 SRDF를 통해 입력 받음
- 초보자도 사용하기 쉬운 GUI 툴 제공 (Rviz의 플러그인 형태)
- C++ 언어 move\_group interface 제공
- Python 언어 move\_commander 제공
- Motion Planning plugin to RViz 제공
- OMPL, KDL, FCL, RRT 등 다양한 오픈 소스 기반의 라이브러리 제공

# Movelt! 실전

```
$ roslaunch open_manipulator_gazebo open_manipulator_gazebo.launch
```

```
$ roslaunch open_manipulator_moveit open_manipulator_demo.launch use_gazebo:=true
```



마지막 과제!

모바일 로봇과 매니퓰레이터의 결합

# 합체! OpenManipulator + TurtleBot3 Waffle

```
$ roscd open_manipulator_with_tb3/urdf  
$ gedit open_manipulator_chain_with_tb3.xacro
```

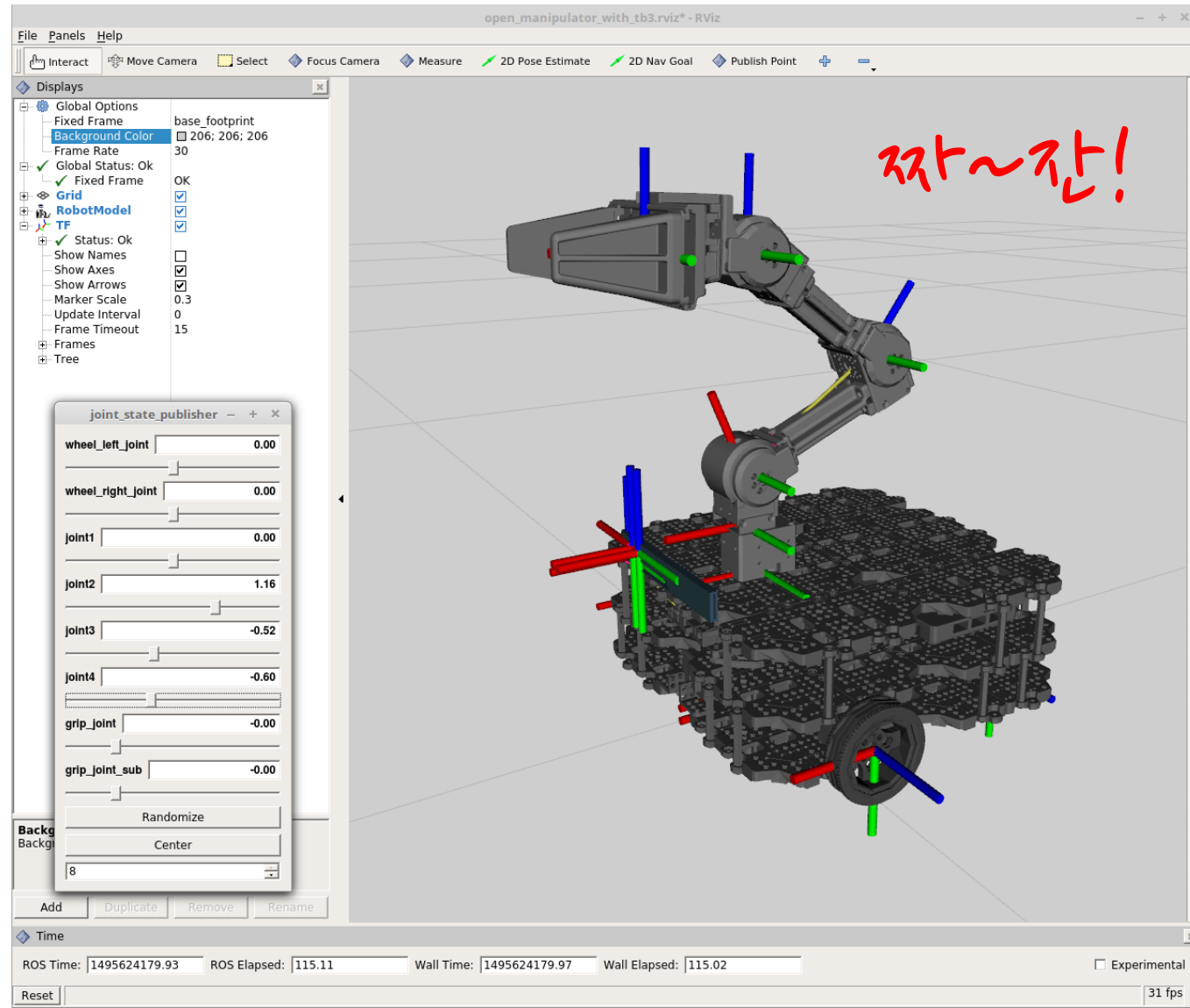
```
<!-- Include TurtleBot3 Waffle URDF -->  
<xacro:include filename="$(find turtlebot3_description)/urdf/turtlebot3_waffle_naked.urdf.xacro" />  
  
<!-- Base fixed joint -->  
<joint name="base_fixed" type="fixed">  
  <origin xyz="-0.005 0.0 0.091" rpy="0 0 0"/>  
  <parent link="base_link"/>  
  <child link="link1"/>  
</joint>
```



터틀봇3에 OpenManipulator 추가

# 합체! OpenManipulator + TurtleBot3 Waffle

```
$ roslaunch open_manipulator_with_tb3 open_manipulator_chain_with_tb3_rviz.launch
```



Key Point?

OpenManipulator 와 MoveIt

알고 보니 쉬웠죠?

솔직히 좀 복잡합니다. T.T

첫 술에 어찌 배 부르겠습니까?

함께 공부합니다. 😊

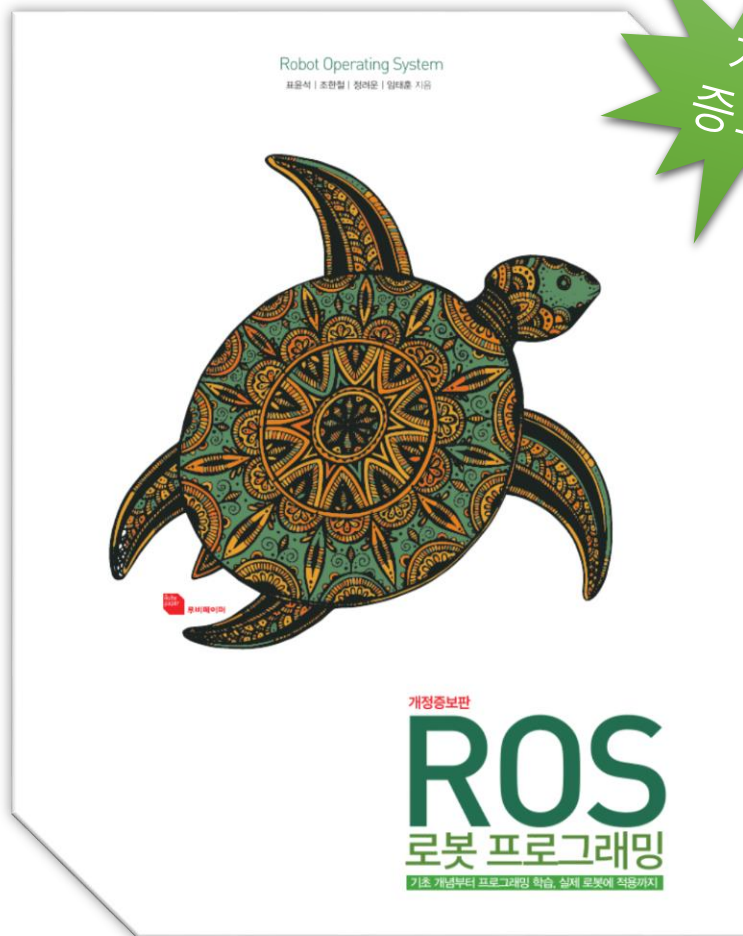


---

**질문  
대환영!**

---

여기서! 광고 하나 나가요~



✓ Direct Link

국내 유일! 최초! ROS 참고서!  
ROS 공식 플랫폼 **TurtleBot3** 개발팀이  
직접 저술한 바이블급 ROS 책

# 여기서! 광고 둘 나가요~

## TURTLEBOT3

### 인공지능(AI) 연구의 시작, ROS 교육용 공식 로봇 플랫폼

터틀봇3는 ROS기반의 저가형 모바일 로봇으로  
교육, 연구, 제품개발, 취미 등 다양한 분야에서 활용 할 수 있습니다.

✓ Direct Link



• Collaboration with  open robotics  intel

여기서! 광고 셋 나가요~



- 오로카
- [www.oroqa.org](http://www.oroqa.org)
- 오픈 로보틱스 지향
- 풀뿌리 로봇공학의 저변 활성화
- 공개 강좌, 세미나, 프로젝트 진행

- 로봇공학을 위한 열린 모임 (KOS-ROBOT)
- [www.facebook.com/groups/KoreanRobotics](https://www.facebook.com/groups/KoreanRobotics)
- 로봇공학 통합 커뮤니티 지향
- 일반인과 전문가가 어울러지는 한마당
- 로봇공학 소식 공유
- 연구자 간의 협력

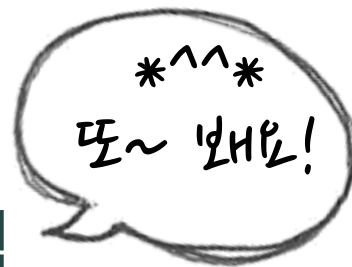
혼자 하기에 답답하시다고요?  
커뮤니티에서 함께 해요~

# 끝.

---

표윤석

Yoonseok Pyo  
pyo@robotis.com  
www.robotpilot.net



[www.facebook.com/yonseok.pyo](https://www.facebook.com/yonseok.pyo)