

04. URDF

AI ROBOT

Exported on 06/03/2021

Table of Contents

1 URDF	4
1.1 URDF - Unified Robot Description Format	4
1.2 link와 joint의 개념	4
1.3 link와 joint의 개념 2	5
1.4 sudo apt-get install liburdfdom-tools	5
1.5 연습용 urdf_exam 패키지 생성	5
1.6 urdf 폴더를 패키지 내에서 생성	6
1.7 현시점 catkin_ws/src 폴더 내부 구조	6
1.8 01_myfirst.urdf 생성	7
1.9 실습코드	7
1.10 check_urdf	7
1.11 urdf_to_graphviz	8
1.12 sudo apt-get install evince	8
1.13 pdf 파일 open	8
1.14 패키지 내에 launch 폴더를 만들고 display.launch 파일 생성	9
1.15 display.launch	10
1.16 워크스페이스에서 build	10
1.17 bashrc를 다시 읽고	10
1.18 roslaunch 실행	11
1.19 만약 입력인자인 model에서 pkg 경로를 인식시키고 싶다면	11
1.20 rviz가 실행되면 add를 누른다	12
1.21 RobotModel을 추가한다	13
1.22 Fixed Frame 옵션을 선택해서 base_link로 변경한다	14
1.23 매번 이렇게 하기 싫다면	14
1.24 rviz가 실행될때	15
1.25 rviz 확장명의 환경이 저장된 것이 보인다	15
1.26 화면에 cylinder가 나타난다	16
2 Pan/Tilt	17

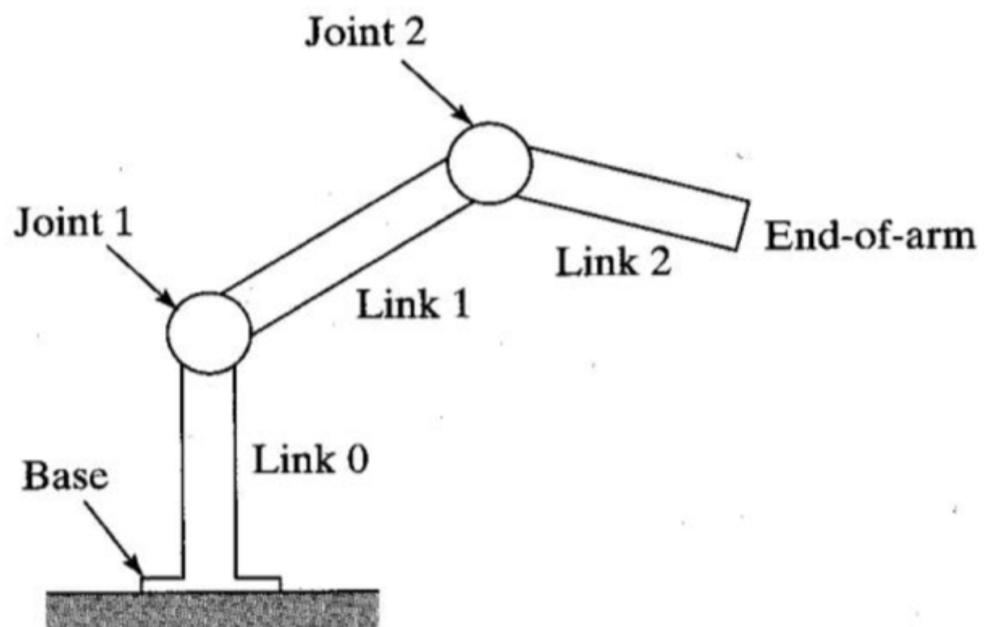
2.1 Robot Joint	18
2.2 Base_link	18
2.3 pan_joint	19
2.4 pan_link	19
2.5 tilt_joint	20
2.6 tilt_link	20
2.7 urdf 폴더에 pan_tilt.urdf 파일 생성	21
2.8 check_urdf	21
2.9 urdf_to_graphiz	21
2.10 그래프로 urdf 확인	22
2.11 최종	23
2.12 launch 폴더에 display.launch 변경	23
2.13 그리고 joint_state_publisher_gui를 설치	24
2.14 roslaunch urdf_exam display.launch model:='\${find urdf_exam}/urdf/pan_tilt.urdf'	24
2.15 gui 조작하기	25
2.16 TF 추가	25
2.17 rqt_graph	26

1 URDF

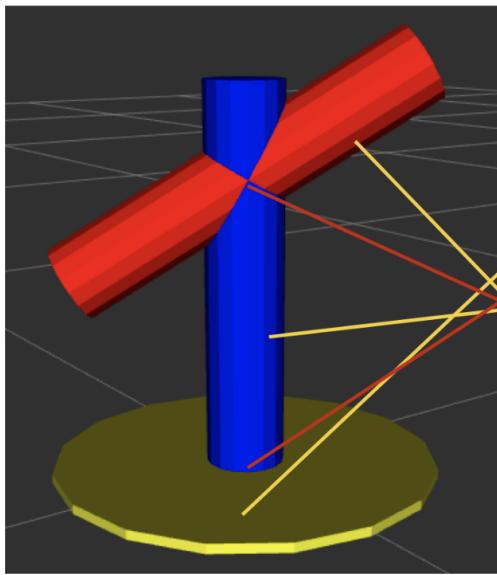
1.1 URDF - Unified Robot Description Format

- Unified Robot Description Format
- Kinematic and basic physics description of a robot
- XML format
- Kinematic tree structure

1.2 link와 joint의 개념



1.3 link와 joint의 개념 2



```
1  <?xml version="1.0"?>
2  <robot name="ex_urdf_pan_tilt">
3
4      <link name="base_link">■
5      </link>
6
7      <joint name="pan_joint" type="revolute">■
8      </joint>
9
10     <link name="pan_link">■
11     </link>
12
13     <joint name="tilt_joint" type="revolute">■
14     </joint>
15
16     <link name="tilt_link">■
17     </link>
18
19 </robot>
```

- 로봇, 부품, 관절 등을 설명하는 XML 포맷의 문서

1.4 sudo apt-get install liburdfdom-tools

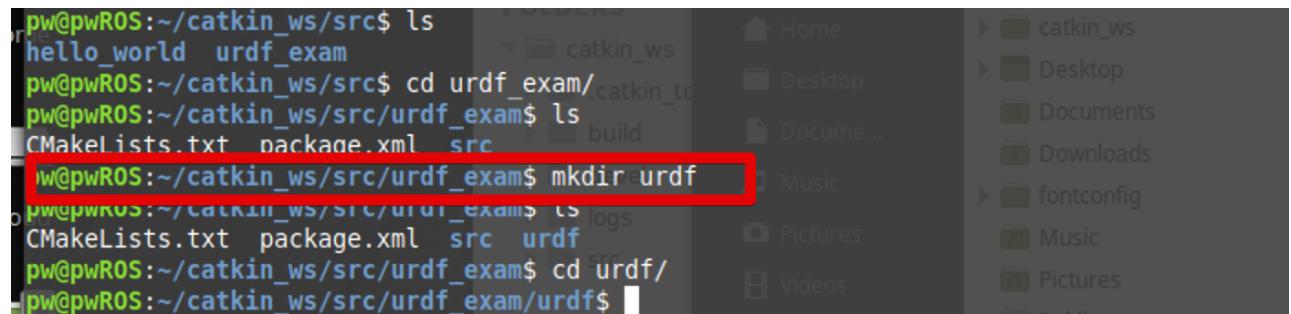
```
pw@melodic:~$ sudo apt install liburdfdom-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  liburdfdom-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.5 kB of archives.
After this operation, 54.3 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 liburdfdom-tools amd64 1.0.0-2u
buntu0.1 [12.5 kB]
Fetched 12.5 kB in 1s (11.4 kB/s)
Selecting previously unselected package liburdfdom-tools.
(Reading database ... 324053 files and directories currently installed.)
Preparing to unpack .../liburdfdom-tools_1.0.0-2ubuntu0.1_amd64.deb ...
Unpacking liburdfdom-tools (1.0.0-2ubuntu0.1) ...
Setting up liburdfdom-tools (1.0.0-2ubuntu0.1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

1.5 연습용 urdf_exam 패키지 생성

```
pw@melodic:~/ws/src$ catkin create pkg urdf_exam --catkin-deps std_msgs rospy urdf xacro  
Creating package "urdf_exam" in "/home/pw/ws/src"...
Created file urdf_exam/package.xml
Created file urdf_exam/CMakeLists.txt
Created folder urdf_exam/src
Successfully created package files in /home/pw/ws/src/urdf_exam.  
pw@melodic:~/ws/src$
```

- `catkin create pkg urdf_exam --catkin-deps std_msgs rospy urdf xacro`
 - 워크스페이스의 `src` 폴더에서 패키지 생성

1.6 urdf 폴더를 패키지 내에서 생성



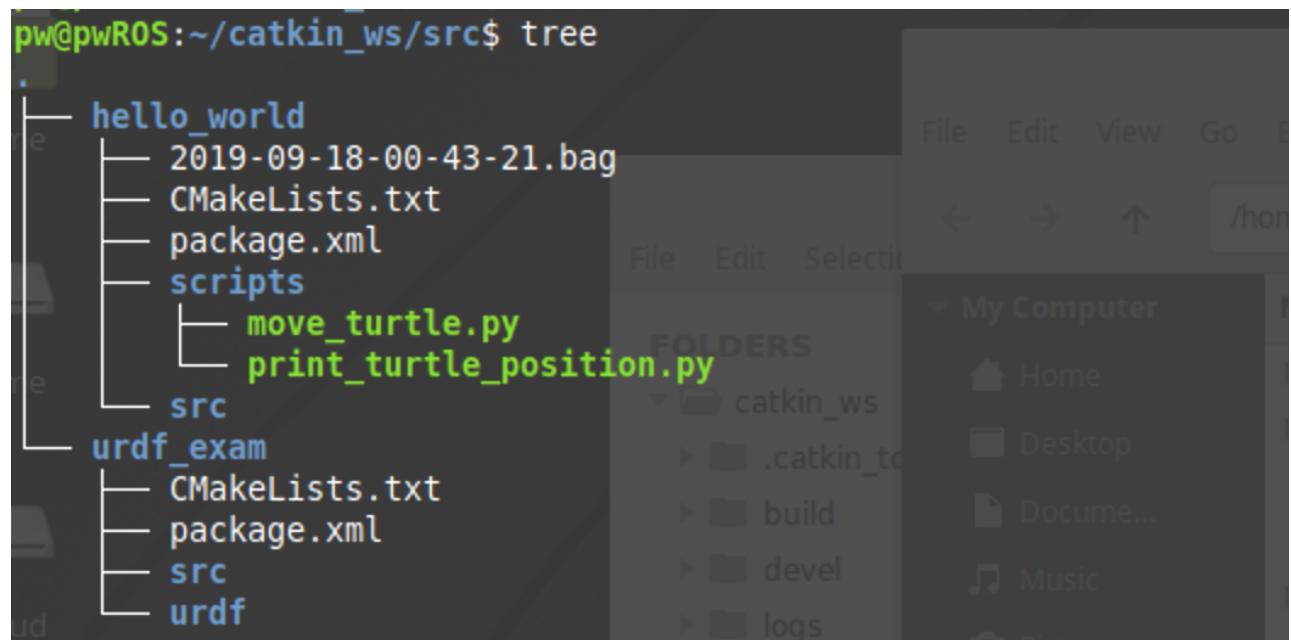
```

pw@pwROS:~/catkin_ws/src$ ls
hello_world urdf_exam
pw@pwROS:~/catkin_ws/src$ cd urdf_exam/
catkin_ws
pw@pwROS:~/catkin_ws/src/urdf_exam$ ls
CMakeLists.txt package.xml src
CMakeLists.txt package.xml src urdf
pw@pwROS:~/catkin_ws/src/urdf_exam$ cd src
urdf
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ 

```

The terminal shows the creation of a 'urdf' directory within the 'src' directory of the 'urdf_exam' package. The current working directory is 'urdf_exam/src'. The 'urdf' directory is highlighted with a red box.

1.7 현시점 catkin_ws/src 폴더 내부 구조



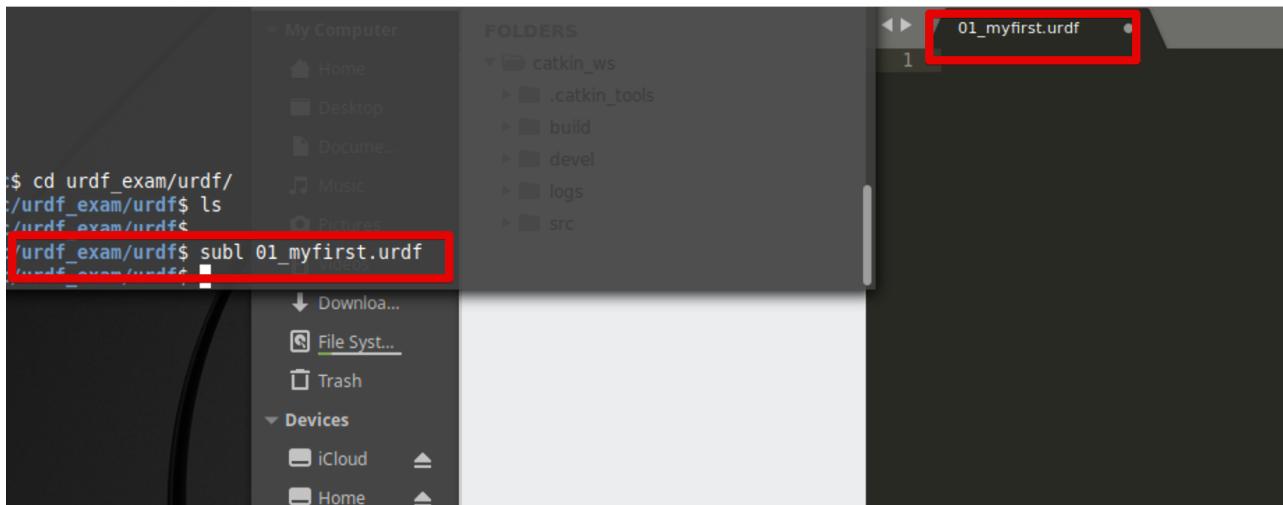
```

pw@pwROS:~/catkin_ws/src$ tree
.
├── hello_world
│   ├── 2019-09-18-00-43-21.bag
│   ├── CMakeLists.txt
│   ├── package.xml
│   └── scripts
│       └── move_turtle.py
│           └── print_turtle_position.py
└── urdf_exam
    ├── CMakeLists.txt
    ├── package.xml
    ├── src
    └── urdf

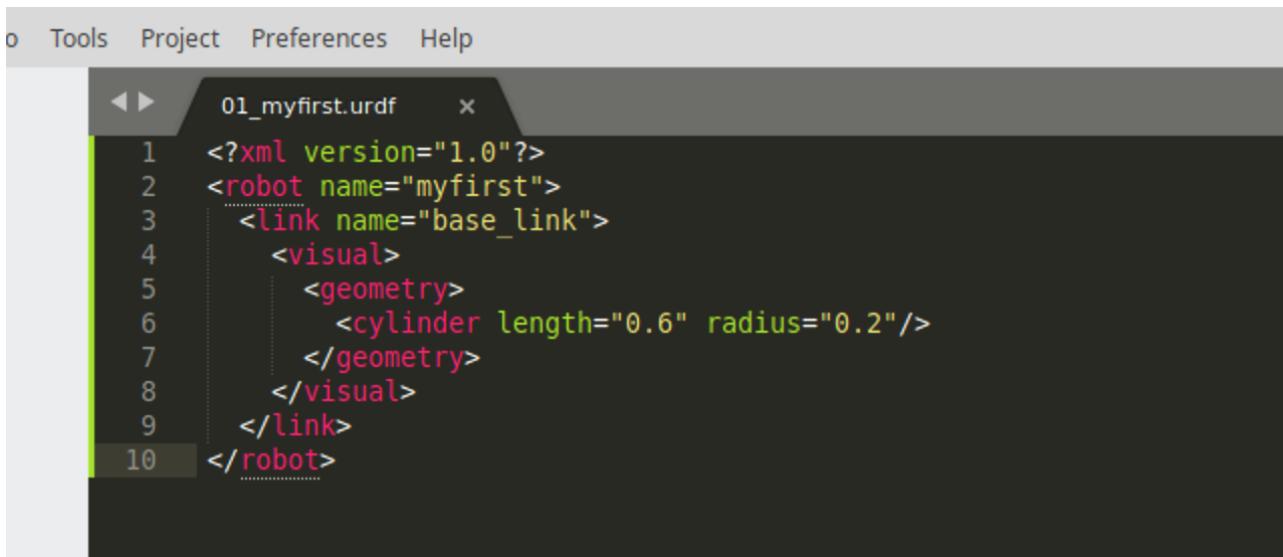
```

The terminal shows the tree structure of the 'catkin_ws/src' directory. It contains two packages: 'hello_world' and 'urdf_exam'. The 'urdf_exam' package has a 'src' directory which contains an 'urdf' directory. A file browser window is also visible, showing the same directory structure. The 'print_turtle_position.py' file is highlighted with a green box.

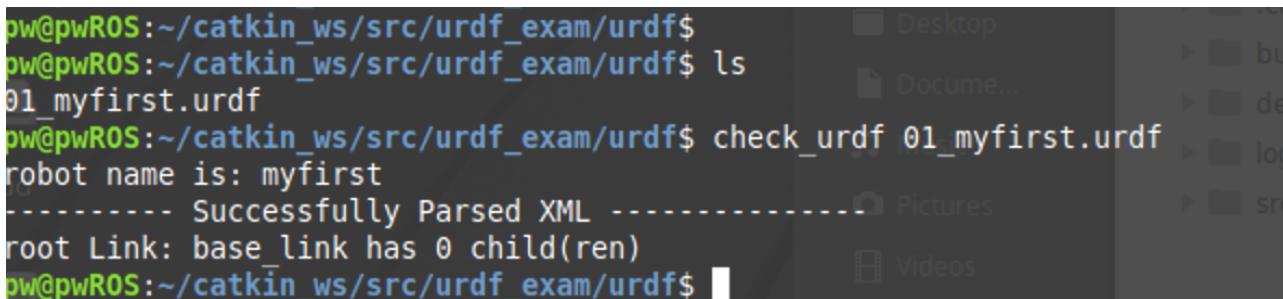
1.8 01_myfirst.urdf 생성



1.9 실습코드



1.10 check_urdf



1.11 urdf_to_graphiz

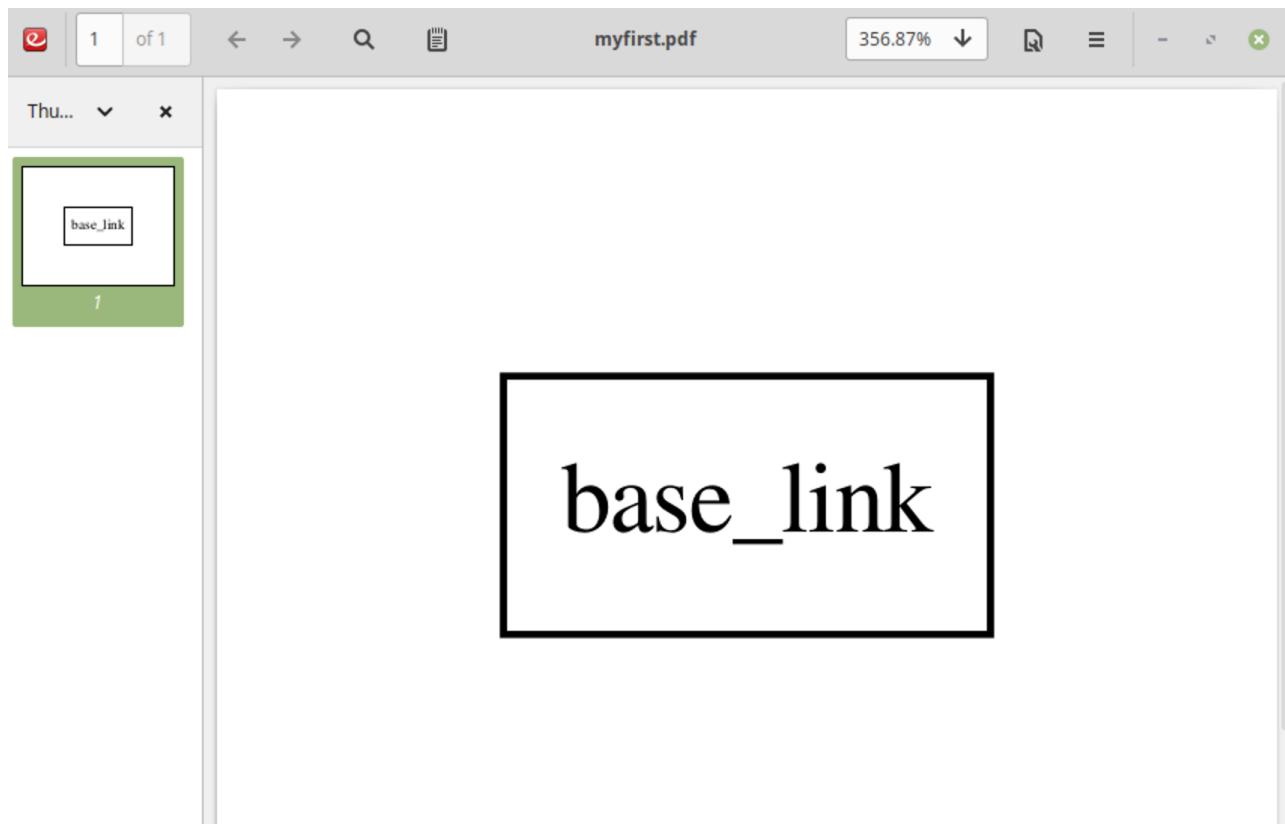
```
root LINK: base_link has 0 child(ren)
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ urdf_to_graphviz 01_myfirst.urdf
Created file myfirst.gv
Created file myfirst.pdf
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$
```

1.12 sudo apt-get install evince

```
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ sudo apt-get install evince
[sudo] password for pw:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  evince-common libevdocument3-4 libevview3-3
Suggested packages:
  nautilus-sendto
The following NEW packages will be installed:
  evince evince-common libevdocument3-4 libevview3-3
0 upgraded, 4 newly installed, 0 to remove and 527 not upgraded.
Need to get 713 kB of archives.
After this operation, 3,708 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

1.13 pdf 파일 open

```
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ ls
01_myfirst.urdf  myfirst.gv  myfirst.pdf
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ evince myfirst.pdf
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$
```



1.14 패키지 내에 launch 폴더를 만들고 display.launch 파일 생성

```
pw@pwROS:~/catkin_ws/src/urdf_exam$ tree
.
├── CMakeLists.txt
└── launch
    └── display.launch
├── package.xml
└── src
    └── urdf
        ├── 01_myfirst.urdf
        ├── myfirst.gv
        └── myfirst.pdf

3 directories, 6 files
```

1.15 display.launch

```

1 <?xml version="1.0"?>
2 <launch>
3   <arg name="model" />
4
5   <param name="robot_description" textfile="$(arg model)"/>
6   <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
7   <node name="robot_state_publisher" pkg="robot_state_publisher" type="state_publisher" />
8   <node name="rviz" pkg="rviz" type="rviz" args="-d $(find urdf_exam)/urdf.rviz" />
9 </launch>

```

- 아직 launch를 배우진 않았으니 가볍게 지나가요^^

1.16 워크스페이스에서 build

```

pw@melodic:~/ws$ catkin build
Profile:           default
Extending:        [cached] /opt/ros/melodic
Workspace:        /home/pw/ws
Build Space:      [exists] /home/pw/ws/build
Devel Space:      [exists] /home/pw/ws/devel
Install Space:    [unused] /home/pw/ws/install
Log Space:        [exists] /home/pw/ws/logs
Source Space:     [exists] /home/pw/ws/src
DESTDIR:          [unused] None
Devel Space Layout: linked

```

1.17 bashrc를 다시 읽고

```

pw@melodic:~/ws$ source ~/.bashrc
ROS1 is activated
pw@melodic:~/ws$ 
pw@melodic:~/ws$ 
pw@melodic:~/ws$ 

```

1.18 roslaunch 실행

```
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ 
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ roslaunch urdf_exam display.launch model:=01_myfirst.urdf
... logging to /home/pw/.ros/log/95bc3fd2-df58-11e9-a289-001c42f3af4f/roslaunch-pwROS-12531.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://pwROS:40119/
SUMMARY
=====
PARAMETERS
* /robot_description: <?xml version="1....
```

- roslaunch는 roscore가 실행중이 아니라면 자기가 실행한다

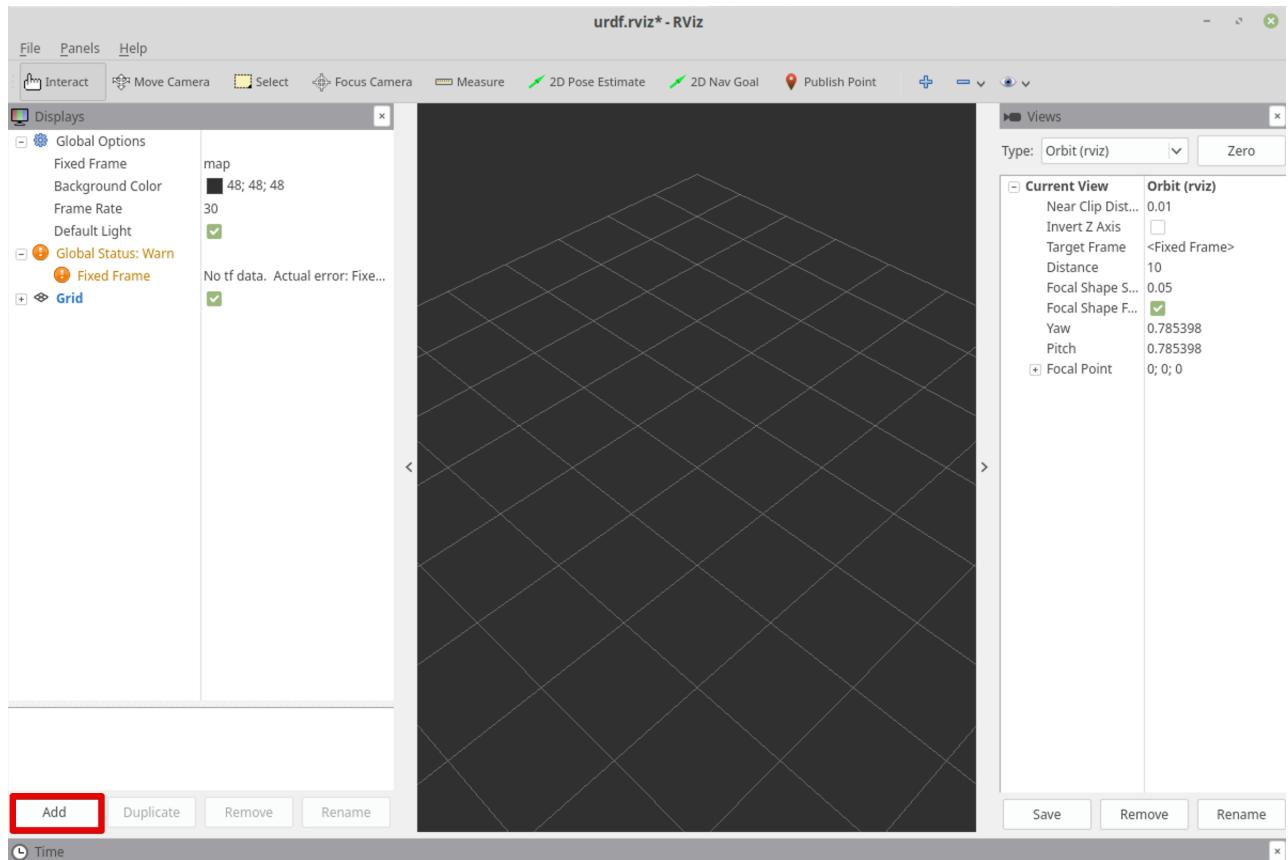
1.19 만약 입력인자인 model에서 pkg 경로를 인식시키고 싶다면

```
pw@melodic:~$ roslaunch urdf_exam display.launch model=$(find urdf_exam)/urdf/01_myfirst.urdf
... logging to /home/pw/.ros/log/e77a9ee8-9bb6-11ea-b077-001c420be203/roslaunch-melodic-4022.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

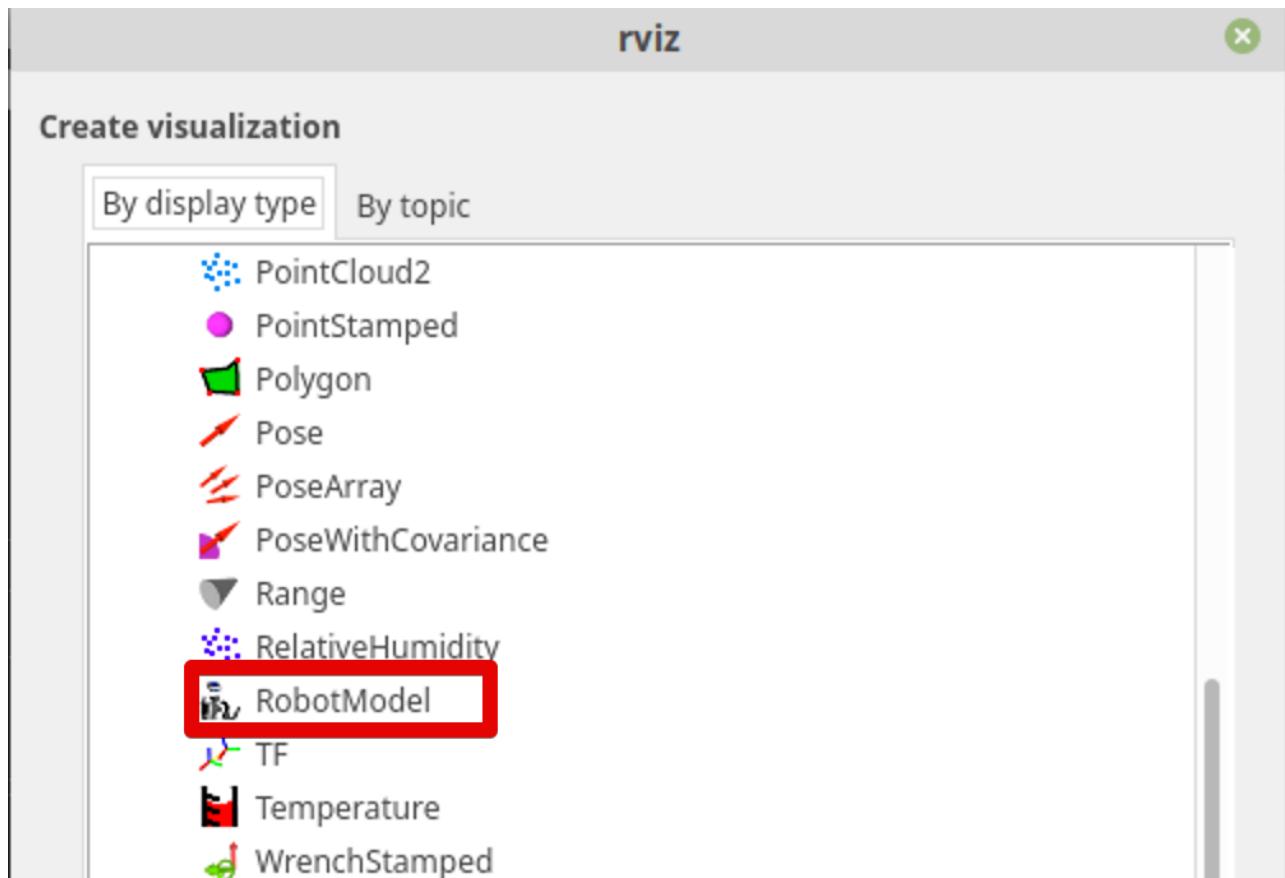
started roslaunch server http://melodic:41469/
SUMMARY
=====
PARAMETERS
* /robot_description: <?xml version="1....
```

- \$(find <pkg name>) 인자를 사용한다

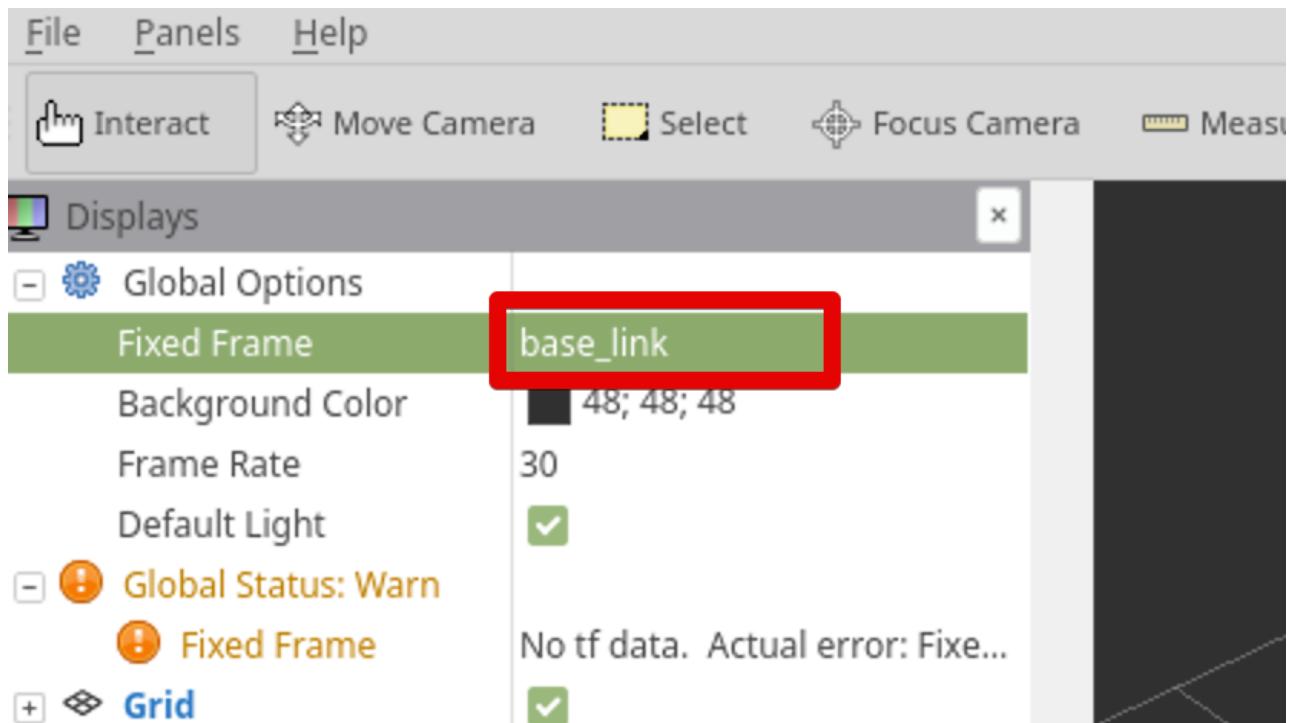
1.20 rviz가 실행되면 add를 누른다



1.21 RobotModel을 추가한다



1.22 Fixed Frame 옵션을 선택해서 base_link로 변경한다

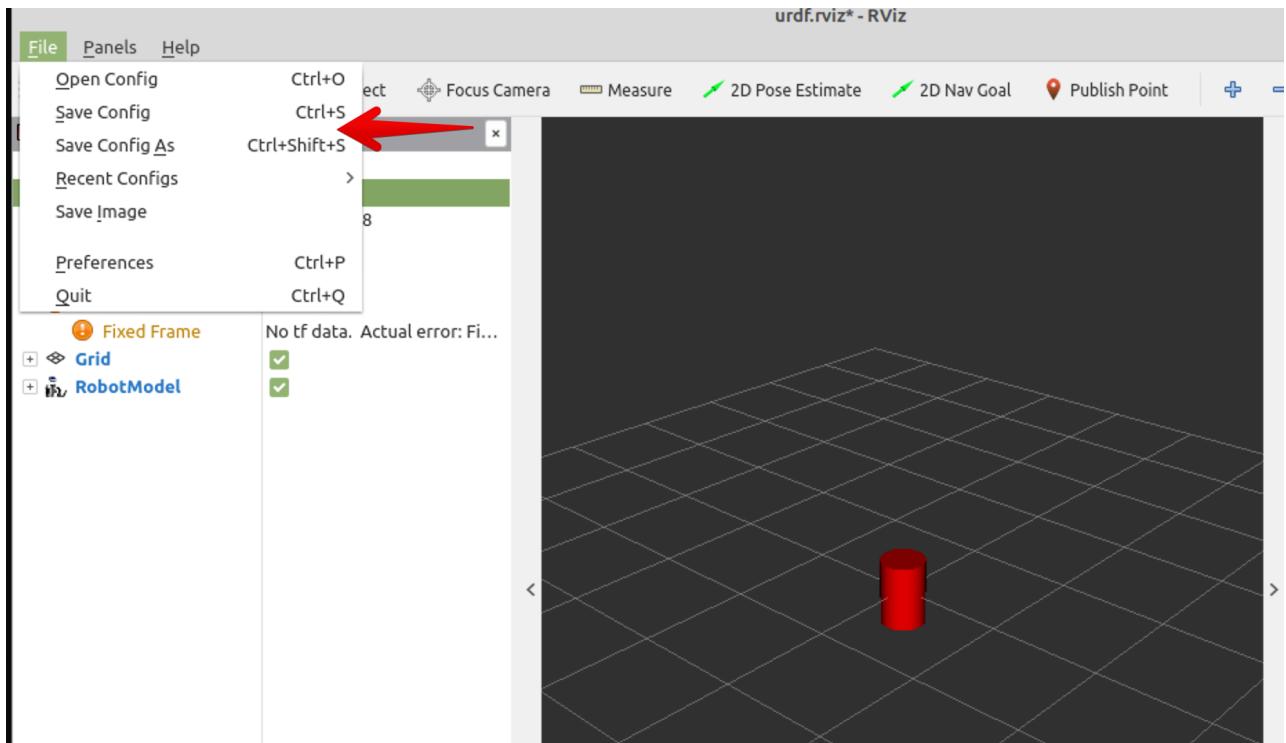


1.23 매번 이렇게 하기 싫다면

```
mira.urdf      x   display.launch      x
1 <?xml version="1.0"?>
2 <launch>
3   <arg name="model" />
4   <param name="robot_description" textfile="$(arg model)"/>
5   <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
6   <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
7   <node name="rviz" pkg="rviz" type="rviz" args="-d $(find urdf_exam)/urdf.rviz" />
8 </launch>
```

- display.launch에서 rviz를 실행할 때, 환경설정을 읽도록 설정해 두었다

1.24 rviz가 실행될 때



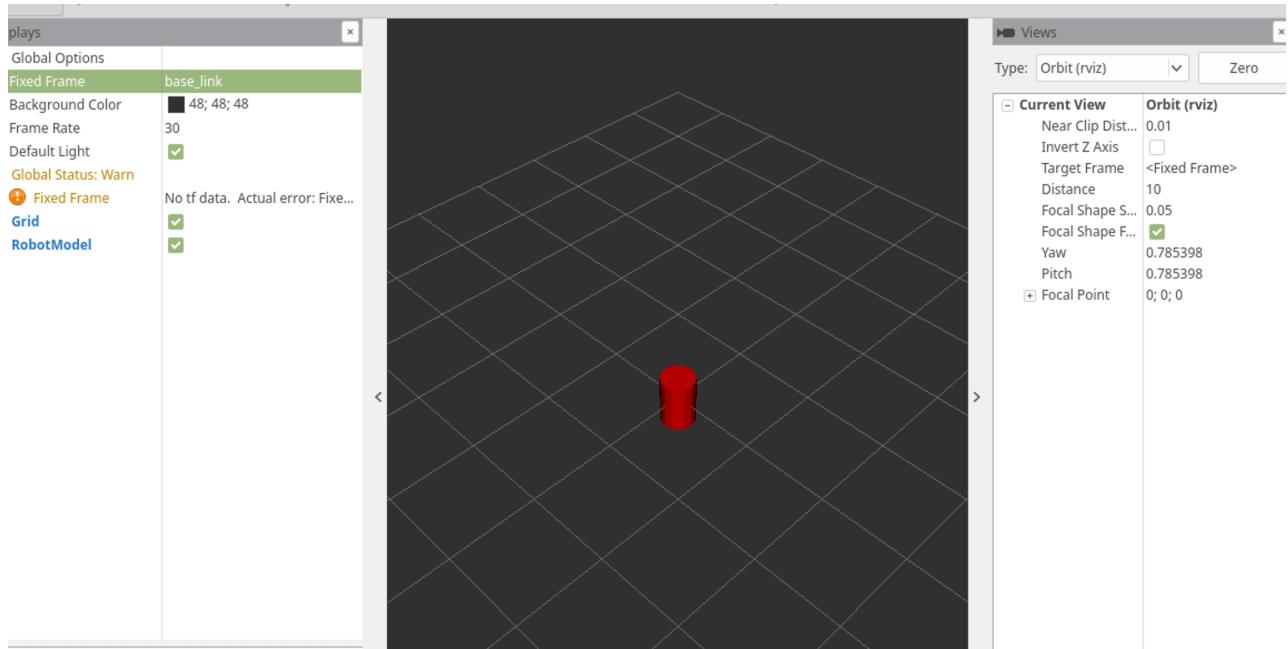
- config를 저장한다

1.25 rviz 확장명의 환경이 저장된 것이 보인다

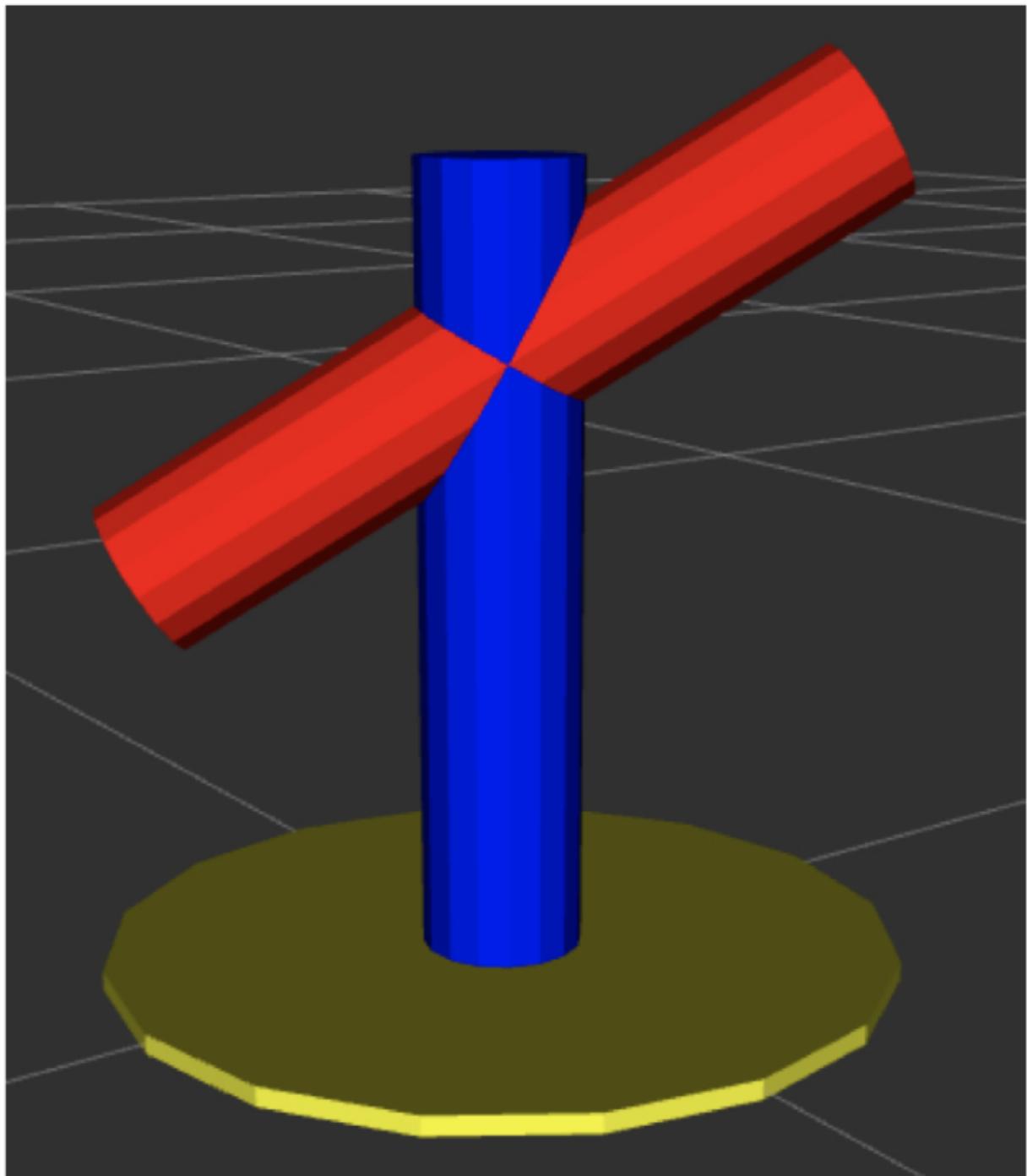
```
pw@melodic:~/ws/src/urdf_exam$ ls -l
total 32
drwxr-xr-x 2 pw pw 4096 May 21 23:49 launch
drwxr-xr-x 2 pw pw 4096 May 21 23:49 src
drwxr-xr-x 2 pw pw 4096 May 22 07:15 urdf
-rw-r--r-- 1 pw pw 4685 May 22 08:02 urdf.rviz
pw@melodic:~/ws/src/urdf_exam$
```

- 보통은 config나 config_rviz 폴더에 넣어 두는 경우가 많다
- 이제 다시 roslaunch를 실행해보면 편해진 것을 확인할 수 있다.

1.26 화면에 cylinder가 나타난다

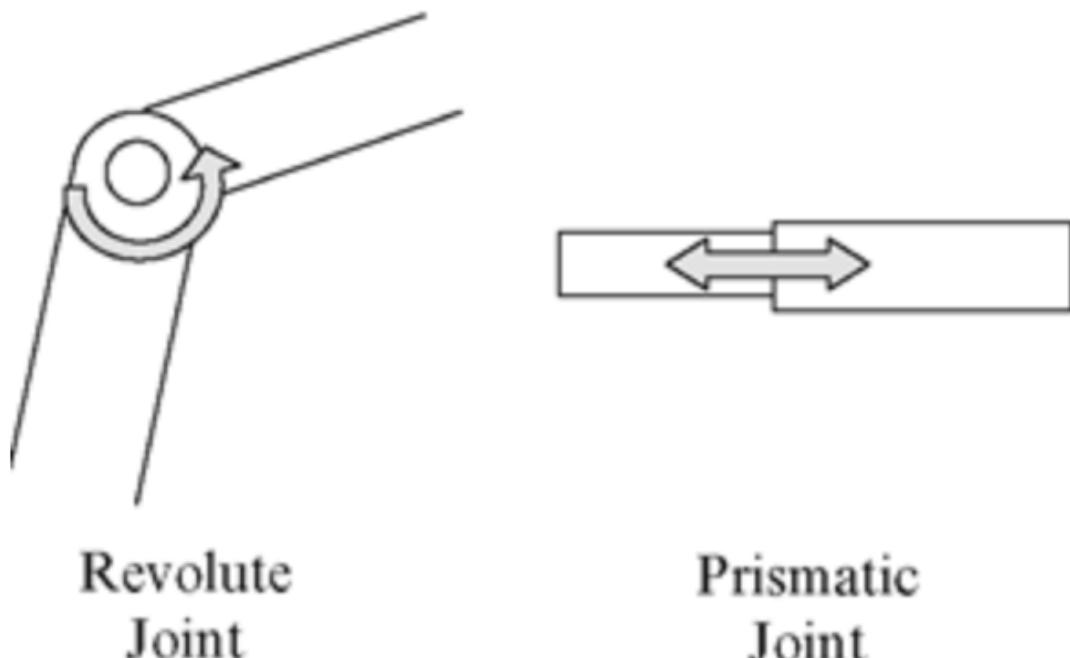


2 Pan/Tilt



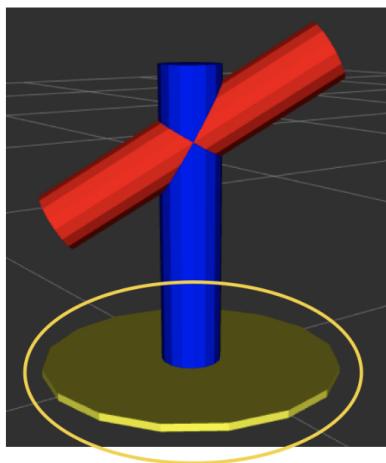
- Pan/tilt 시스템은 두 축으로 이뤄진 일종의 로봇암
- CCTV 등 다양한 분야를 가지고 있음

2.1 Robot Joint



- 가장 많이 사용하는 두 조인트

2.2 Base_link

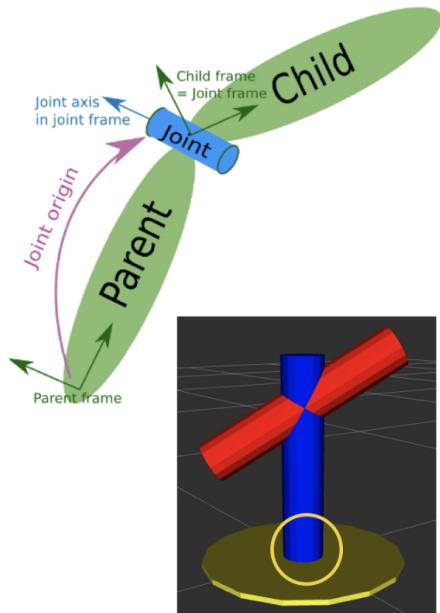


```

4 <link name="base_link">
5   <visual>
6     <geometry>
7       <cylinder length="0.01" radius="0.2"/>
8     </geometry>
9     <origin rpy="0 0 0" xyz="0 0 0"/>
0   <material name="yellow">
1     <color rgba="1 1 0 1"/>
2   </material>
3 </visual>
4
5   <collision>
6     <geometry>
7       <cylinder length="0.03" radius="0.2"/>
8     </geometry>
9     <origin rpy="0 0 0" xyz="0 0 0"/>
0   </collision>
1
2   <inertial>
3     <mass value="1"/>
4     <inertia ixx="1.0" ixy="0.0" ixz="0.0" iyy="1.0" iyz="0.0" izz="1.0"/>
5   </inertial>
6 </link>

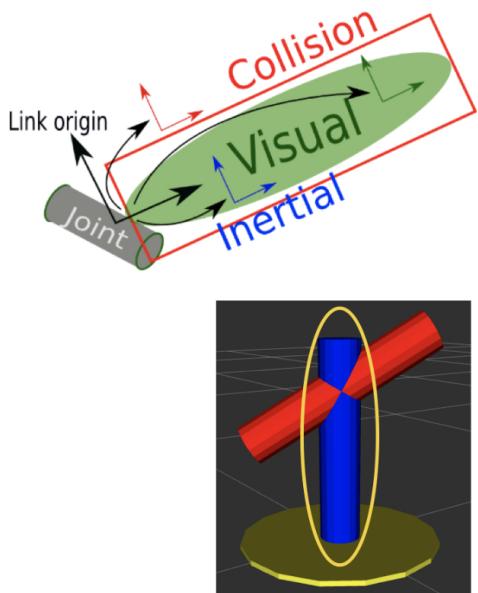
```

2.3 pan_joint



```
<joint name="pan_joint" type="revolute">
  <parent link="base_link"/>
  <child link="pan_link"/>
  <origin xyz="0 0 0.1" />
  <axis xyz="0 0 1" />
  <limit effort="300" velocity="0.1" lower="-3.14" upper="3.14" />
  <dynamics damping="50" friction="1" />
</joint>
```

2.4 pan_link

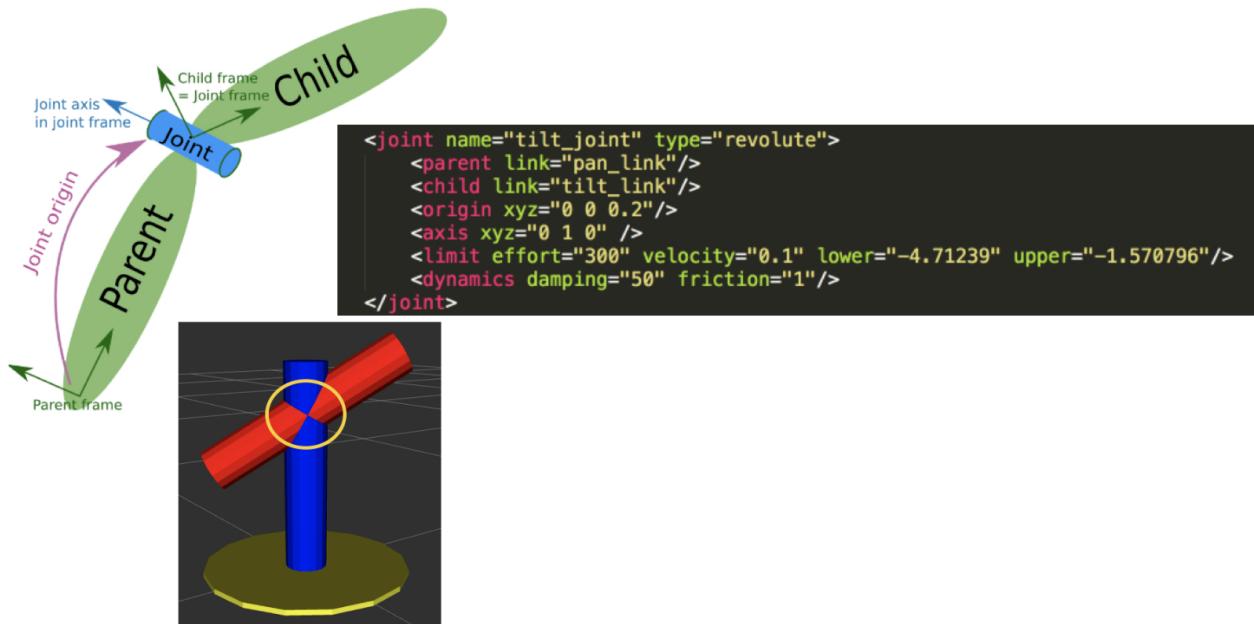


```
<link name="pan_link">
  <visual>
    <geometry>
      <cylinder length="0.4" radius="0.04"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0.09"/>
    <material name="red">
      <color rgba="0 0 1 1"/>
    </material>
  </visual>

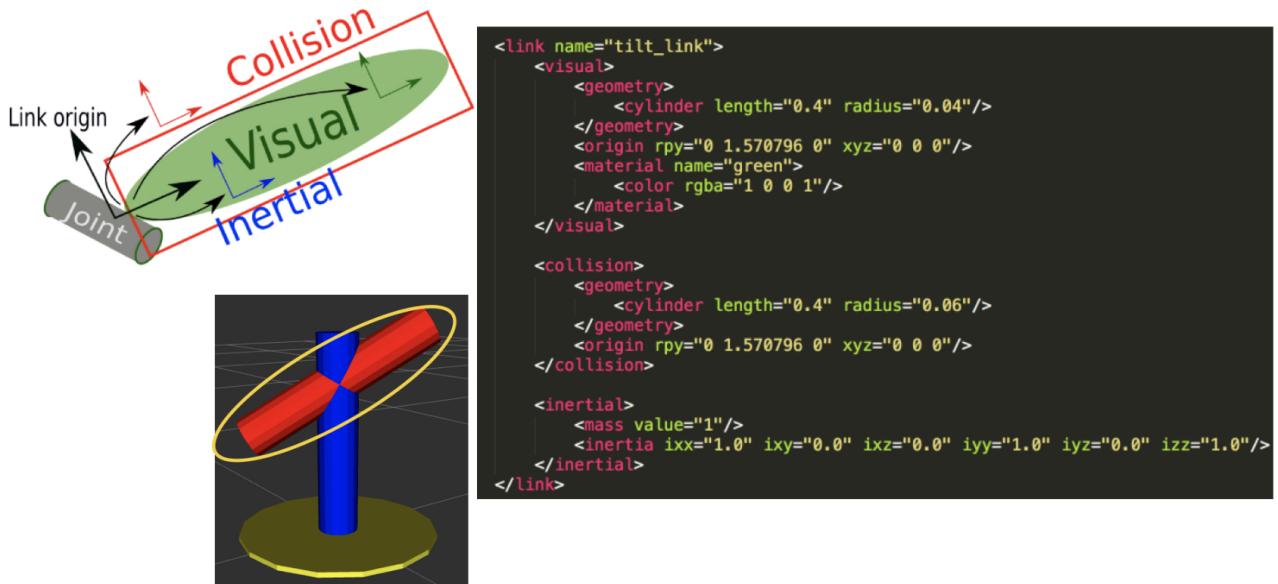
  <collision>
    <geometry>
      <cylinder length="0.4" radius="0.06"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0.09"/>
  </collision>

  <inertial>
    <mass value="1"/>
    <inertia ixx="1.0" ixy="0.0" ixz="0.0" iyy="1.0" iyz="0.0" izz="1.0" />
  </inertial>
</link>
```

2.5 tilt_joint



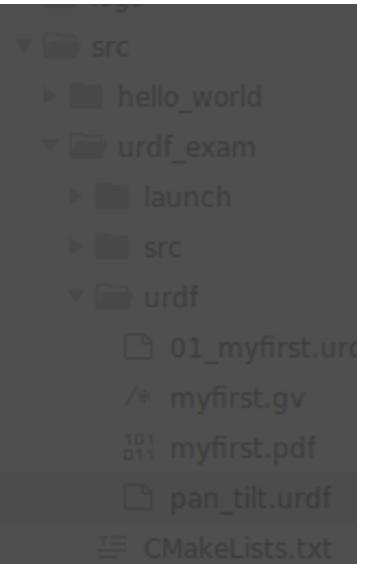
2.6 tilt_link



2.7 urdf 폴더에 pan_tilt.urdf 파일 생성

```
pw@pwROS:~/catkin_ws/src/urdf_exam$  
pw@pwROS:~/catkin_ws/src/urdf_exam$ tree  
.  
├── CMakeLists.txt  
├── launch  
│   └── display.launch  
├── package.xml  
└── src  
    └── urdf  
        ├── 01_myfirst.urdf  
        ├── myfirst.gv  
        └── pan_tilt.urdf
```

3 directories, 7 files



- `subl pan_tilt.urdf`라는 명령을 urdf_exam/urdf 폴더안에서 실행하면 됨

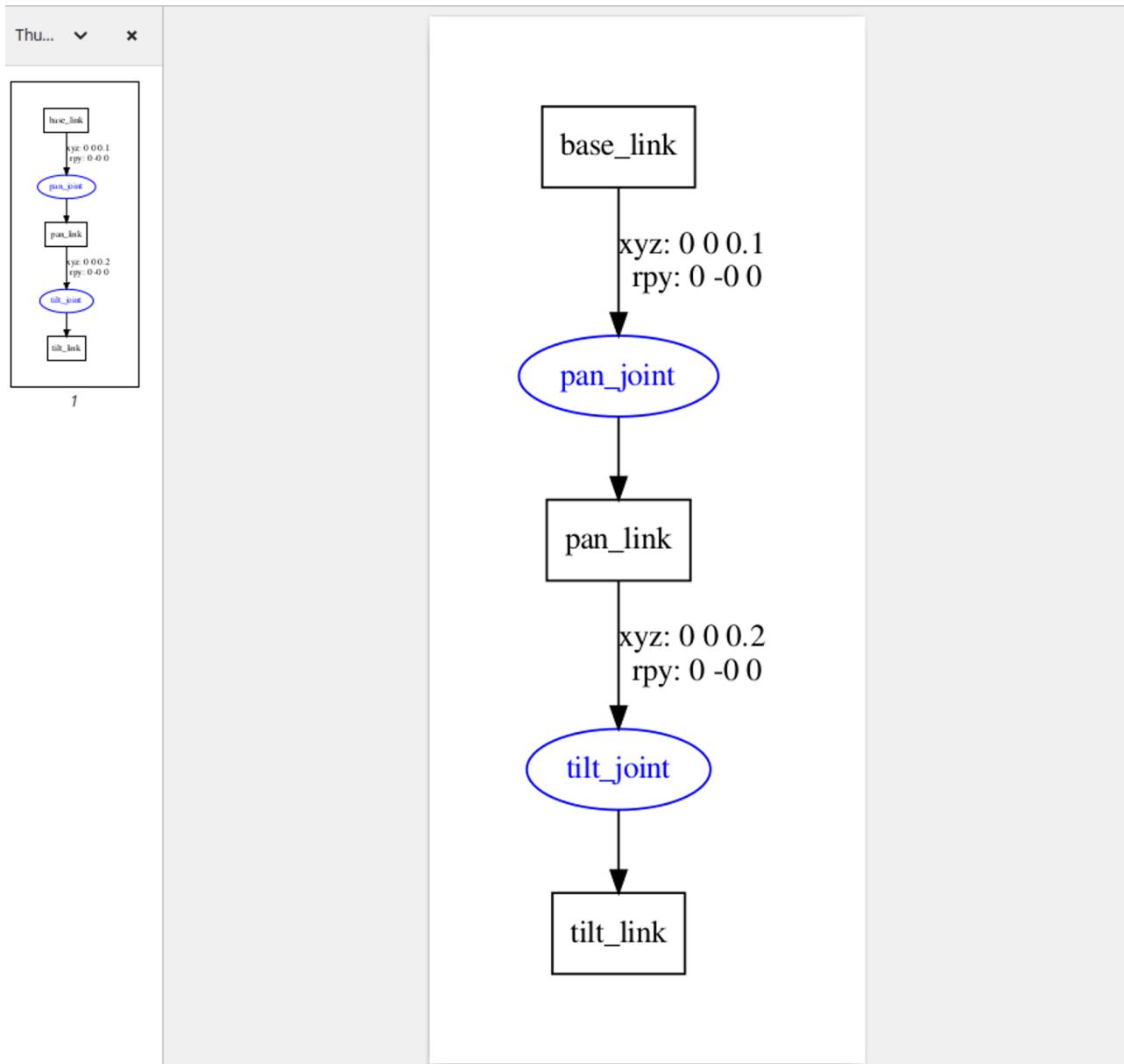
2.8 check_urdf

```
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ check_urdf pan_tilt.urdf
robot name is: ex_urdf_pan_tilt
----- Successfully Parsed XML -----
root Link: base_link has 1 child(ren)
    child(1): pan_link
        child(1): tilt_link
```

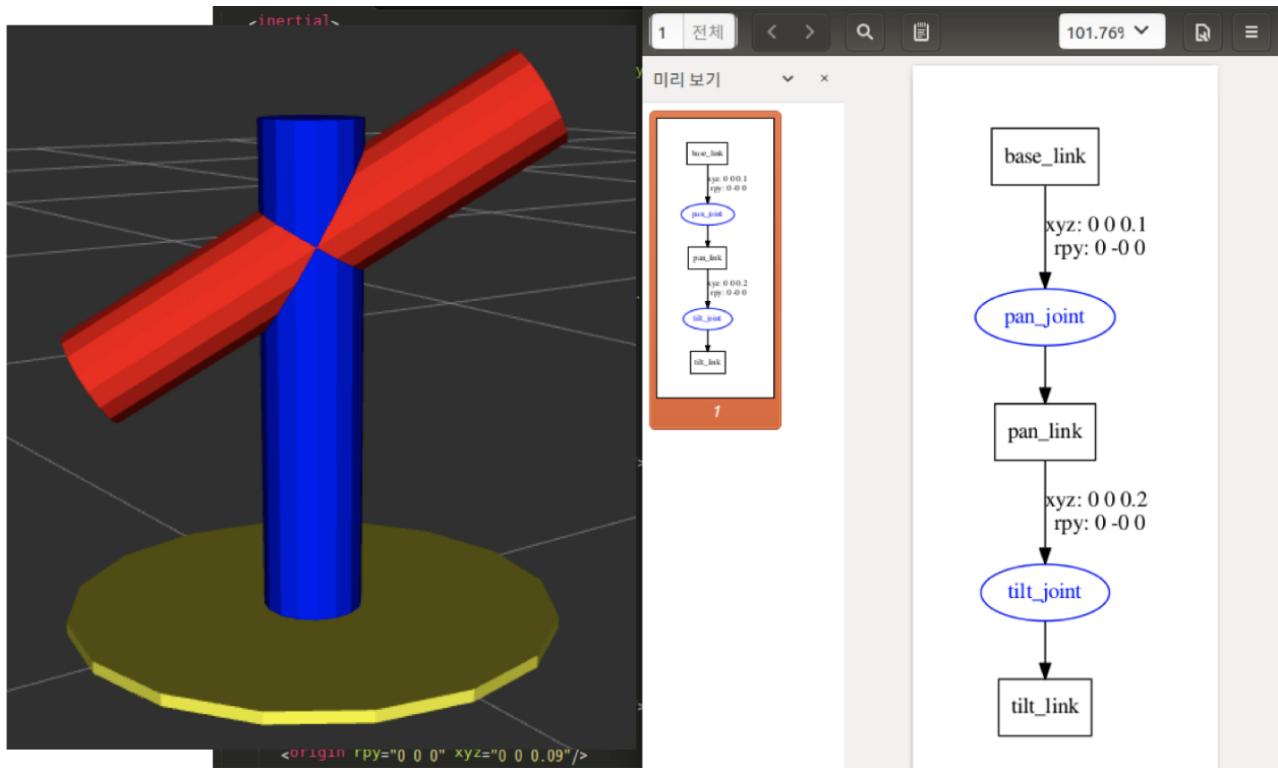
2.9 urdf_to_graphiz

```
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ urdf_to_graphiz pan_tilt.urdf
Created file ex_urdf_pan_tilt.gv
Created file ex_urdf_pan_tilt.pdf
pw@pwROS:~/catkin_ws/src/urdf_exam/urdf$ evince ex_urdf_pan_tilt.pdf
```

2.10 그레프로 urdf 확인



2.11 최종



2.12 launch 폴더에 display.launch 변경

```

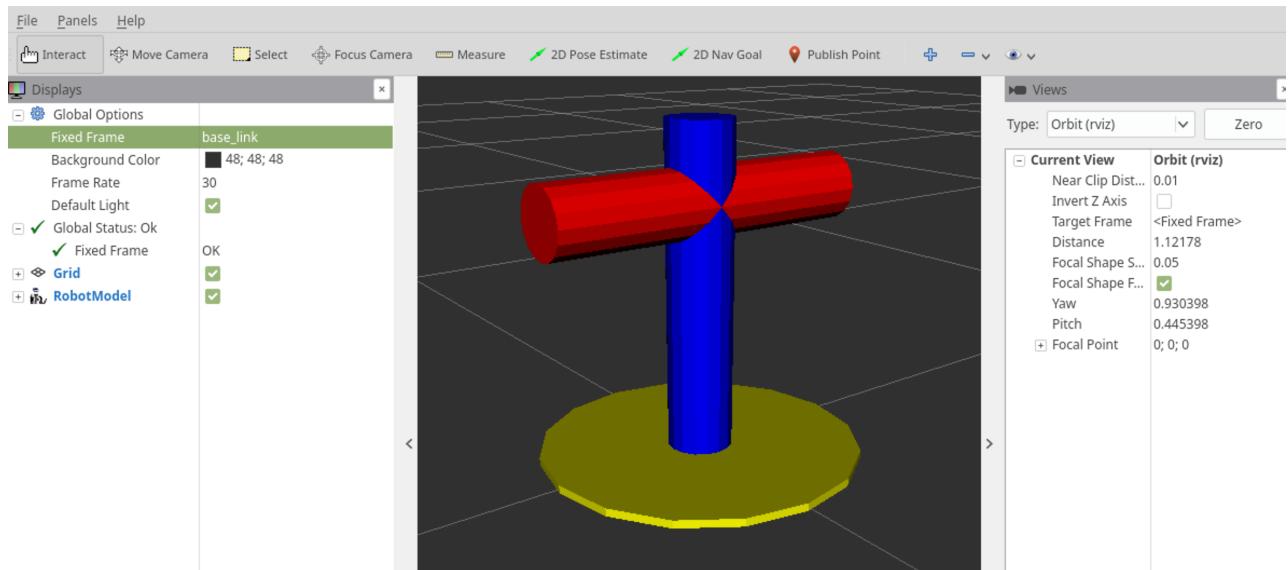
1  <?xml version="1.0"?>
2  <launch>
3      <arg name="model" />
4
5      <param name="robot_description" textfile="$(arg model)" />
6      <param name="use_gui" value="true"/>
7      <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
8      <node name="robot_state_publisher" pkg="robot_state_publisher" type="state_publisher" />
9      <node name="rviz" pkg="rviz" type="rviz" args="-d $(find urdf_exam)/urdf.rviz" required="True" />
10     </launch>

```

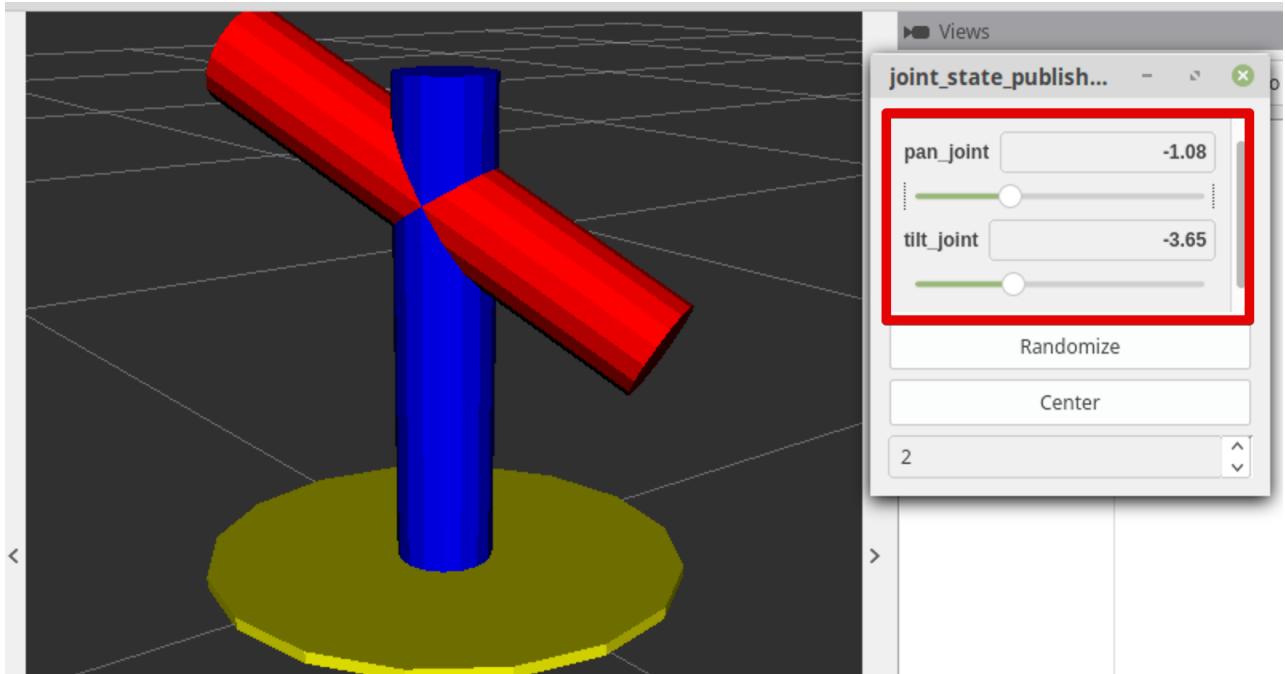
2.13 그리고 joint_state_publisher_gui를 설치

```
pw@melodic:~$ sudo apt install ros-melodic-joint-state-publisher
ros-melodic-joint-state-publisher      ros-melodic-joint-state-publisher-gui
pw@melodic:~$ sudo apt install ros-melodic-joint-state-publisher*
[sudo] password for pw:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ros-melodic-joint-state-publisher' for glob 'ros-melodic-joint-state-publisher*'
Note, selecting 'ros-melodic-joint-state-publisher-gui' for glob 'ros-melodic-joint-state-publisher*'
'
ros-melodic-joint-state-publisher is already the newest version (1.12.15-1bionic.20200320.132142).
ros-melodic-joint-state-publisher set to manually installed.
The following NEW packages will be installed:
```

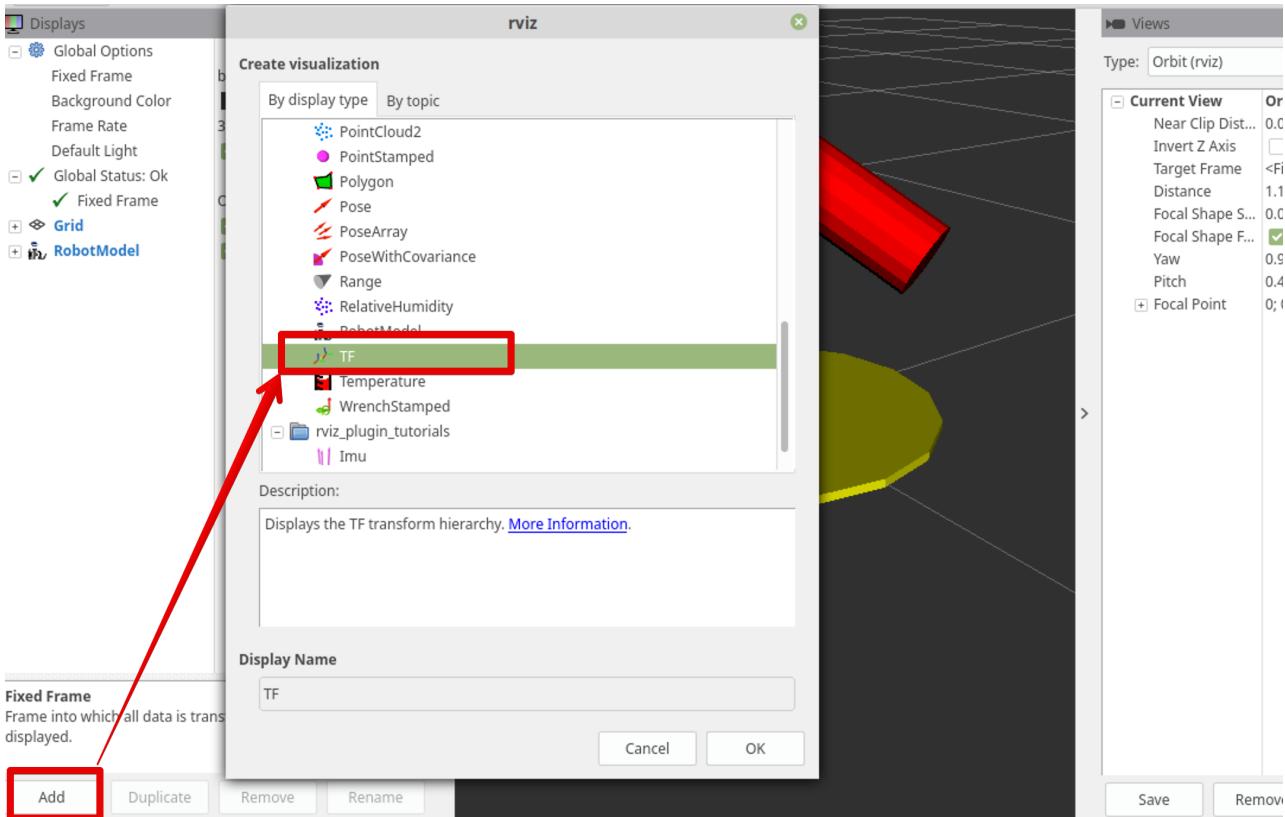
2.14 roslaunch urdf_exam display.launch model:=\$(find urdf_exam)/urdf/pan_tilt.urdf

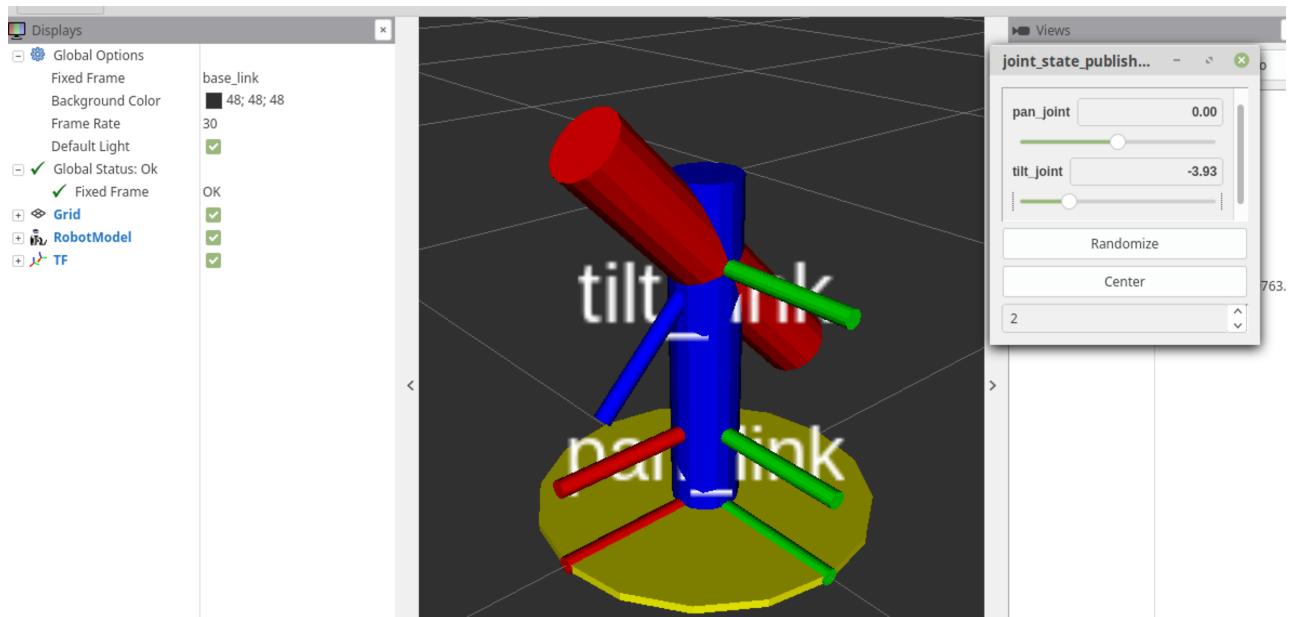


2.15 gui 조작하기



2.16 TF 추가





2.17 rqt_graph

