

GAZEBO

AI ROBOT

Exported on 12/04/2021

Table of Contents

1 GAZEBO.....	5
1.1 Why GAZEBO.....	5
1.2 Gazebo에서 필요한 수치	5
1.3 일단 간단하게 mira_simple.urdf 복사본 하나 생성	5
1.4 base_link에 inertial 항목 추가.....	6
1.5 inertia란?	6
1.6 Tensor of inertia	7
1.7 inertial의 계산	8
1.8 base_link의 collision 설정	9
1.9 roll_M1_link의 inertial 설정	9
1.10 물리엔진 시뮬레이션을 위한 gazebo 파라미터 설정	10
1.11 pitch_M2_link에 inertia 추가	10
1.12 pitch_M2_link에 gazebo 물리수치 입력	11
1.13 yaw_M3_link에 inertial 추가	11
1.14 yaw_M3_link의 gazebo 물리 수치 입력	12
1.15 head_link에 inertial 추가	12
1.16 head_link에 collision 추가	13
1.17 head_link 물성치 추가	13
1.18 left_eye_link에 inertial 추가	14
1.19 left_eye 물성치	14
1.20 right_eye_link에 inertial 추가	15
1.21 right_eye 물성치.....	15
1.22 지금까지의 실습파일	16
2 GOD의 영역에 가볼까	17
2.1 gazebo에 spawn하기 위해 mira_spawn.launch 작성	17
2.2 gazebo관련 pkg 설치	17
2.3 실행	17
2.4 play	18
2.5 터미널에 이런 에러가 나타난다면	19

2.6	~/.ignition/fuel/config.yaml 을 열어서	19
2.7	표시된 url을	19
2.8	api.ignitionrobotics.org 로 변경.....	20
2.9	그리고 다시 실행하면 에러가 사라지고 없다	20
2.10	아마 물리엔진을 살짝 구경할 수 있을 겁니다.....	20
2.11	mira_spawn.launch	20
2.12	보너스 - 재미를 찾아보자.....	21
2.13	dae 파일	22
2.14	복사하고	22
2.15	head_link 부분 수정	23
2.16	약간 무섭나요?	23
3	구동축 만들어 주기	24
3.1	urdf 파일을 복사해서 만들어서 시작하자.....	24
3.2	joint tag.....	24
3.3	ROS controllers	24
3.4	roll_joint에 액츄에이터 추가.....	25
3.5	pitch_joint에 엑츄에이터 추가.....	26
3.6	yaw_joint에 엑츄에이터 추가.....	27
3.7	gazebo 제어기 플러그인 추가	27
3.8	config 폴더 하나 추가	28
3.9	각 조인트 제어기 설정	28
3.10	joint-state-controller 설치	29
3.11	trajectory-controller 설치	29
3.12	effort controller 설치.....	30
3.13	mira_gazebo_controller.urdf를 만들자	30
3.14	실행해보자.....	32
3.15	이제 버틴다.....	32
3.16	rosrun rqt_gui rqt_gui	33
3.17	만약 각 관절의 PID값을 실시간으로 변경하고 싶다면	33
4	GAZEBO에서 센서 사용하기	34
4.1	urdf에 카메라 plugin 추가	34

4.2	다시 roslaunch urdf_exam mira_spawn_controller.launch.....	35
4.3	rostopic list를 보면 카메라 관련 토픽이 나타난 것이 보인다.....	35
4.4	rosrun rqt_image_view rqt_image_view.....	36
4.5	적당히	36
4.6	배치해보자.....	37
4.7	나타난다~^^	37

1 GAZEBO

1.1 Why GAZEBO



Simulation

Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. At your fingertips is a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces.

- URDF에서 rviz로 로봇을 관찰하는 것은 즐거운 일이지만, 물성(중력, 관성, 충돌)을 보여주지는 않는다
- 이런 경우 물리엔진이 탑재가 된 프로그램이 필요한데
- ROS 유저는 GAZEBO를 많이 사용함

1.2 Gazebo에서 필요한 수치

- Inertia: 물체의 관성
- Gazebo Physical Properties: 색상, 마찰 등
- Collisions: 충돌 (바닥위에 놓여있는 속성 등에 필요)
- Gazebo Sensors: 카메라, 컨트롤러, 센서 등

1.3 일단 간단하게 mira_simple.urdf 복사본 하나 생성

```
pw@melodic:~/ws/src/urdf_exam/urdfs$ cd ..description/models/mira/meshes
pw@melodic:~/ws/src/urdf_exam/urdfs$ cp mira_simple.urdf mira_gazebo.urdf
pw@melodic:~/ws/src/urdf_exam/urdfs$
```

1.4 base_link에 inertial 항목 추가

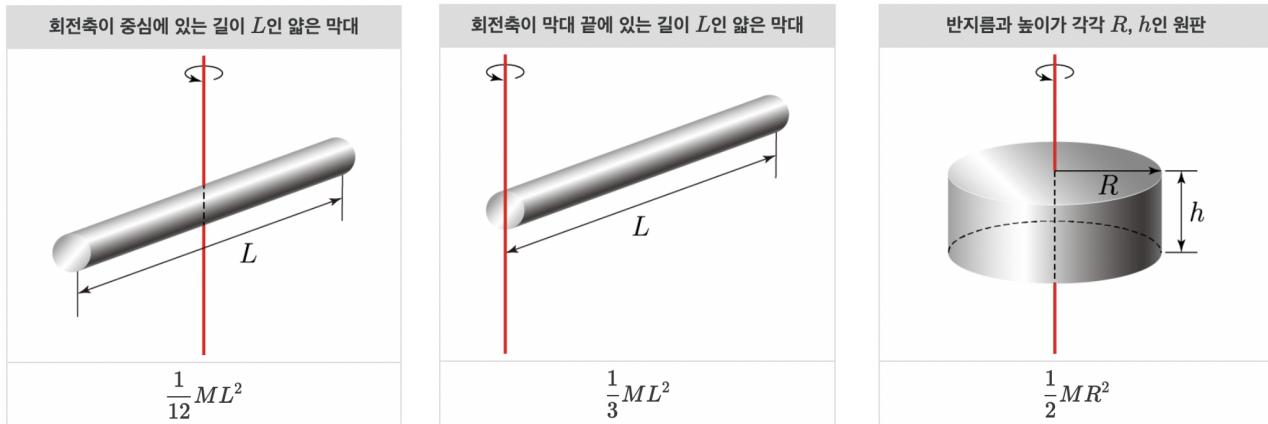
```

24   <link name="base_link">
25     <inertial>
26       <origin xyz="0 0 0" rpy="0 0 0"/>
27       <mass value="0.18" />
28       <inertia ixx="0.0002835" ixy="0.0" ixz="0.0" iyy="0.0002835" iyz="0.0" izz="0.000324"/>
29     </inertial>
30
31     <visual>
32       <origin rpy="0.0 0 0" xyz="0 0 0"/>
33       <geometry>
34         <mesh filename="package://urdf_exam/meshes/mira_body_v3.dae"/>
35       </geometry>
36     </visual>
37   </link>
38
39

```

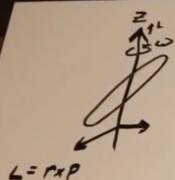
- **origin**: 해당 요소의 위치와 자세를 지정
- **mass**: 요소의 질량
- **inertia**: 관성행렬 inertia matrix

1.5 inertia란?



- 물체가 자신의 회전을 유지하는 정도를 나타내는 회전량
- 그래서 x, y, z 축 중심 회전 성분과 x에서 바라본 x축 회전, 등의 요소들이 있음
- 간단한 예) <https://pinkwink.kr/37>
- 간단한 예) <https://pinkwink.kr/352>
- 실제 작업에서 inertia는 기계 설계 파트에서 제공해 줄 때가 많다
- 그러나 로봇엔지니어가 직접 계산할 수도 있어야한다 (대충이라도)

1.6 Tensor of inertia



$$\begin{aligned}
 L &= \sum \vec{\ell}_\alpha = \sum \vec{r}_\alpha \times \vec{p}_\alpha = \sum \vec{r}_\alpha \times M \vec{v}_\alpha & \vec{v}_\alpha &= \vec{\omega} \times \vec{r}_\alpha \\
 &= \sum m \vec{r}_\alpha \times (\vec{\omega} \times \vec{r}_\alpha) & \vec{r}_\alpha &= (x_\alpha, y_\alpha, z_\alpha) \\
 \vec{r}_\alpha \times (\vec{\omega} \times \vec{r}) &= (y^2 + z^2) \omega_x + (-xy) \omega_y + (-xz) \omega_z \\
 &\quad + (-yx) \omega_x + (x^2 + z^2) \omega_y + (-yz) \omega_z \\
 &\quad + (-zx) \omega_x + (-zy) \omega_y + (x^2 + y^2) \omega_z \\
 \sum m \vec{r}_\alpha \times (\vec{\omega} \times \vec{r}_\alpha) &= I_{xx} \omega_x + I_{xy} \omega_y + I_{xz} \omega_z \\
 &\quad + I_{yx} \omega_x + I_{yy} \omega_y + I_{yz} \omega_z \\
 &\quad + I_{zx} \omega_x + I_{zy} \omega_y + I_{zz} \omega_z = I \vec{\omega} \\
 I_{xx} &= \sum m (y_\alpha^2 + z_\alpha^2) \\
 I_{xy} &= -\sum m x_\alpha y_\alpha \quad I \vec{\omega} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad I \vec{\omega} - \lambda \mathbf{1} \vec{\omega} = \mathbf{0} \\
 I_{xz} &=
 \end{aligned}$$

$\mathbf{1} = [1 \ 1 \ 1]$

- 관성 텐서는 관성 모멘트가 회전축에 따라 달라지는 것을 보완하고, 평면 상의 강체 회전 뿐만 아니라 3차원 상의 회전을 기술하기 위한 물리량이다.

1.7 inertial의 계산

Body	Figure	Mass Center	Moments of Inertia
Rectangular Parallelepiped		—	$I_{xx} = \frac{1}{12}m(a^2 + l^2)$ $I_{yy} = \frac{1}{12}m(b^2 + l^2)$ $I_{zz} = \frac{1}{12}m(a^2 + b^2)$ $I_{y_1y_1} = \frac{1}{12}mb^2 + \frac{1}{3}ml^2$
Circular Cylinder		—	$I_{xx} = \frac{1}{4}mr^2 + \frac{1}{12}ml^2$ $I_{x_1x_1} = \frac{1}{4}mr^2 + \frac{1}{3}ml^2$ $I_{zz} = \frac{1}{2}mr^2$
Semicylinder		$\bar{x} = \frac{4r}{3\pi}$	$I_{xx} = I_{yy} = \frac{1}{4}mr^2 + \frac{1}{12}ml^2$ $I_{x_1x_1} = I_{y_1y_1} = \frac{1}{4}mr^2 + \frac{1}{3}ml^2$ $I_{zz} = \frac{1}{2}mr^2$
Circular Cylindrical Shell		—	$I_{xx} = \frac{1}{2}mr^2 + \frac{1}{12}ml^2$ $I_{x_1x_1} = \frac{1}{2}mr^2 + \frac{1}{3}ml^2$ $I_{zz} = mr^2$
Half Cylindrical Shell		$\bar{x} = \frac{2r}{\pi}$	$I_{xx} = I_{yy} = \frac{1}{2}mr^2 + \frac{1}{12}ml^2$ $I_{x_1x_1} = I_{y_1y_1} = \frac{1}{2}mr^2 + \frac{1}{3}ml^2$ $I_{zz} = mr^2$
Sphere		—	$I_{zz} = \frac{2}{5}mr^2$
Spherical Shell		—	$I_{zz} = \frac{2}{3}mr^2$
Hemisphere		$\bar{x} = \frac{3r}{8}$	$I_{xx} = I_{yy} = I_{zz} = \frac{2}{5}mr^2$
Hemispherical Shell		$\bar{x} = \frac{r}{2}$	$I_{xx} = I_{yy} = I_{zz} = \frac{2}{3}mr^2$
Uniform Slender Rod		—	$I_{yy} = \frac{1}{12}ml^2$ $I_{y_1y_1} = \frac{1}{3}ml^2$

1.8 base_link의 collision 설정

```

23     <link name="base_link">
24
25         <inertial>
26             <origin xyz="0 0 0" rpy="0 0 0"/>
27             <mass value="0.18" />
28             <inertia ixx="0.0002835" ixy="0.0" ixz="0.0" iyy="0.0002835" iyz="0.0" izz="0.000324"/>
29         </inertial>
30
31         <collision>
32             <origin xyz="0 0 0" rpy="0 0 0"/>
33             <geometry>
34                 <cylinder radius="0.06" length="0.09"/>
35             </geometry>
36         </collision>
37
38         <visual>
39             <origin rpy="0.0 0 0" xyz="0 0 0"/>
40             <geometry>
41                 <mesh filename="package://urdf_exam/meshes/mira_body_v3.dae"/>
42             </geometry>
43         </visual>
44     </link>
45

```

1.9 roll_M1_link의 inertial 설정

```

47     <link name="roll_M1_link">
48
49         <inertial>
50             <origin xyz="0 0 0" rpy="0 0 0"/>
51             <mass value="0.001" />
52             <inertia ixx="2.7083333333e-08" ixy="0.0" ixz="0.0" iyy="2.7083333333e-08" iyz="0.0" izz="5e-08"/>
53         </inertial>
54
55         <visual>
56             <origin rpy="0 0 0" xyz="0 0 0"/>
57             <geometry>
58                 <cylinder length="0.005" radius="0.01"/>
59             </geometry>
60             <material name="red"/>
61         </visual>
62     </link>
63
64

```

1.10 물리엔진 시뮬레이션을 위한 gazebo 파라미터 설정

```

58         <geometry>
59             <cylinder length="0.005" radius="0.01"/>
60         </geometry>
61     <material name="red"/>
62   </visual>
63 </link>
64
65 <gazebo reference="roll_M1_link">
66   <kp>100000.0</kp>
67   <kd>100000.0</kd>
68   <mu1>10.0</mu1>
69   <mu2>10.0</mu2>
70   <material>Gazebo/Red</material>
71 </gazebo>
72
73 <joint name="roll_joint" type="revolute">
74

```



- mu1: 정적 마찰 계수. 간단히 말해서, 물체가 움직이기 시작할 때까지 마찰
- mu2: 동적 마찰 계수. 간단히 말해서, 물체가 움직일 때 마찰
- kp: 정적 접촉 강성. 예를 들어 링크 재질이 대리석에 가까운지 (매우 강하고 큰 값) 고무 (부드러운 재질, 낮은 값)에 더 가까운지를 결정
- kd: 동적 접촉 강성.

1.11 pitch_M2_link에 inertia 추가

```

81 <link name="pitch_M2_link">
82
83   <inertial>
84     <origin xyz="0 0 0" rpy="0 0 0"/>
85     <mass value="0.001" />
86     <inertia ixx="2.7083333333e-08" ixy="0.0" ixz="0.0" iyy="2.7083333333e-08" iyz=
87       "0.0" izz="5e-08"/>
88   </inertial>
89
90   <visual>
91     <origin rpy="0 0 0" xyz="0 0 0"/>
92     <geometry>
93       <cylinder length="0.005" radius="0.01"/>
94     </geometry>
95   </visual>
96 </link>
97

```



1.12 pitch_M2_link에 gazebo 물리수치 입력

```

93         </geometry>
94         <material name="green"/>
95     </visual>
96 </link>
97
98     <gazebo reference="pitch_M2_link">
99         <kp>100000.0</kp>
100        <kd>100000.0</kd>
101        <mu1>10.0</mu1>
102        <mu2>10.0</mu2>
103        <material>Gazebo/Green</material>
104    </gazebo>
105
106
107    <joint name="pitch_joint" type="revolute">
108        <parent link="roll_M1_link"/>
109        <child link="pitch_M2_link"/>
110        <origin xyz="0 0 0" rpy="0 -1.5708 0" />

```

1.13 yaw_M3_link에 inertial 추가

```

115     <link name="yaw_M3_link">
116
117         <inertial>
118             <origin xyz="0 0 0" rpy="0 0 0"/>
119             <mass value="0.001" />
120             <inertia ixx="2.7083333333e-08" ixy="0.0" ixz="0.0" iyy="2.7083333333e-08" iyz=
121                 "0.0" izz="5e-08"/>
122         </inertial>
123
124         <visual>
125             <origin rpy="0 0 0" xyz="0 0 0"/>
126             <geometry>
127                 <cylinder length="0.005" radius="0.01"/>
128             </geometry>
129             <material name="blue"/>
130         </visual>
131
132     <joint name="yaw_joint" type="continuous">
133         <parent link="pitch_M2_link"/>
134         <child link="yaw_M3_link"/>

```

1.14 yaw_M3_link의 gazebo 물리 수치 입력

```

126           <cylinder length="0.005" radius="0.01" />
127       </geometry>
128     <material name="blue"/>
129   </visual>
130 </link>
131
132   <gazebo reference="yaw_M3_link">
133     <kp>100000.0</kp>
134     <kd>100000.0</kd>
135     <mu1>10.0</mu1>
136     <mu2>10.0</mu2>
137     <material>Gazebo/Blue</material>
138   </gazebo>
139
140
141   <joint name="yaw_joint" type="continuous">
142     <parent link="pitch_M2_link"/>
143     <child link="yaw_M3_link"/>
144     <origin xyz="0.01 0 0" rpy="0 1.5708 0" />
145     <limit effort="0.1" velocity="0.01" />

```



1.15 head_link에 inertial 추가

```

143   <link name="yaw_M3_link">
144     <origin xyz="0.01 0 0" rpy="0 1.5708 0" />
145     <limit effort="0.1" velocity="0.01" />
146     <axis xyz="0 0 1" />
147   </joint>
148
149   <link name="head_link">
150
151     <inertial>
152       <origin xyz="0 0 0" rpy="0 0 0" />
153       <mass value="0.02" />
154       <inertia ixx="2.61666666667e-05" ixy="0.0" ixz="0.0" iyy="2.61666666667e-05" iyz=
155         "0.0" izz="3.6e-05" />
156     </inertial>
157
158     <visual>
159       <origin rpy="0.0 0 0" xyz="0 0 0" />
160       <geometry>
161         <mesh filename="package://urdf_exam/meshes/mira_head_v5.dae" />
162       </geometry>
163     </visual>
164   </link>
165
166   <joint name="base_head_joint" type="fixed">

```



1.16 head_link에 collision 추가

```

149     <link name="head_link">
150
151     <inertial>
152         <origin xyz="0 0 0" rpy="0 0 0"/>
153         <mass value="0.02" />
154         <inertia ixx="2.61666666667e-05" ixy="0.0" ixz="0.0" iyy="2.61666666667e-05" iyz="0.0" izz="3.6e-05"/>
155     </inertial>
156
157     <collision>
158         <origin xyz="0 0 0.015" rpy="0 0 0"/>
159         <geometry>
160             <sphere radius="0.06"/>
161         </geometry>
162     </collision>
163
164     <visual>
165         <origin rpy="0.0 0 0" xyz="0 0 0"/>
166         <geometry>
167             <mesh filename="package://urdf_exam/meshes/mira_head_v5.dae"/>
168         </geometry>
169     </visual>
170 </link>

```

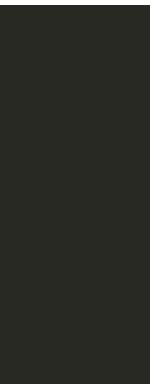


1.17 head_link 물성치 추가

```

166         </geometry>
167     </visual>
168 </link>
169
170     <gazebo reference="head_link">
171         <kp>100000.0</kp>
172         <kd>100000.0</kd>
173         <mu1>10.0</mu1>
174         <mu2>10.0</mu2>
175         <material>Gazebo/White</material>
176     </gazebo>
177
178     <joint name="base_head_joint" type="fixed">
179         <parent link="yaw_M3_link"/>
180         <child link="head_link"/>
181

```



1.18 left_eye_link에 inertial 추가

```

175     <origin xyz="0 0 0" rpy="0 0 0"/>
176   </joint>
177
178   <link name="left_eye_link">
179
180     <inertial>
181       <origin xyz="0 0 0" rpy="0 0 0"/>
182       <mass value="0.001" />
183       <inertia ixx="9.1875e-09" ixy="0.0" ixz="0.0" iyy="9.1875e-09" iyz="0.0" izz=
184         1.378125e-08"/>
185     </inertial>
186
187     <visual>
188       <origin rpy="0.0 0 0" xyz="0 0 0"/>
189       <geometry>
190         <!--
191           <cylinder radius="0.00525" length="0.00525"/>
192         -->
193         <mesh filename="package://urdf_exam/meshes/mira_eye_v4.dae"/>
194       </geometry>
195     </visual>
196   </link>

```



1.19 left_eye 물성치

```

95     <mesh filename="package://urdf_exam/meshes/mira_eye_v4.dae"/>
96   </geometry>
97 </visual>
98
99   <gazebo reference="left_eye_link">
100     <kp>100000.0</kp>
101     <kd>100000.0</kd>
102     <mu1>10.0</mu1>
103     <mu2>10.0</mu2>
104     <material>Gazebo/White</material>
105   </gazebo>
106
107   <joint name="head_lefteye_joint" type="fixed">
108     <parent link="head_link"/>
109     <child link="left_eye_link"/>
110
111
112
113
114

```



1.20 right_eye_link에 inertial 추가

```

94         </geometry>
95     </visual>
96   </link>
97
98   <joint name="head_lefteye_joint" type="fixed">
99     <parent link="head_link"/>
100    <child link="left_eye_link"/>
101    <origin xyz="0.0095 0.057 0.0085" rpy="-1.5708 0 0"/>
102  </joint>
103
104  <link name="right_eye_link">
105
106    <inertial>
107      <origin xyz="0 0 0" rpy="0 0 0"/>
108      <mass value="0.001" />
109      <inertia ixx="9.1875e-09" ixy="0.0" ixz="0.0" iyy="9.1875e-09" iyz="0.0" izz=
110        1.378125e-08"/>
111    </inertial>
112
113    <visual>
114      <origin rpy="0.0 0 0" xyz="0 0 0"/>
115      <geometry>
116        <!--
117          <cylinder radius="0.00525" length="0.00525"/>
118        -->
119        <mesh filename="package://urdf_exam/meshes/mira_eye_v4.dae"/>
120    </geometry>

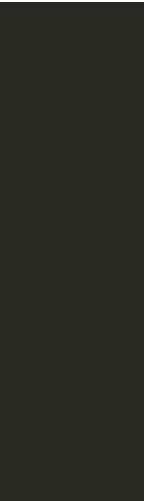
```

1.21 right_eye 물성치

```

2           <mesh filename="package://urdf_exam/meshes/mira_eye_v4.dae"/>
3         </geometry>
4       </visual>
5     </link>
6
7     <gazebo reference="right_eye_link">
8       <kp>100000.0</kp>
9       <kd>100000.0</kd>
10      <mu1>10.0</mu1>
11      <mu2>10.0</mu2>
12      <material>Gazebo/White</material>
13    </gazebo>
14
15
16
17   <joint name="head_righteye_joint" type="fixed">
18     <parent link="head_link"/>
19     <child link="right_eye_link"/>
20     <origin xyz="-0.0095 0.057 0.0085" rpy="-1.5708 0 0"/>

```



1.22 지금까지의 실습파일

mira_gazebo.urdf

(see page 5)

2 GOD의 영역에 가볼까

2.1 gazebo에 spawn하기 위해 mira_spawn.launch 작성

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <launch>
3      <param name="robot_description" command="cat '$(find urdf_exam)/urdf/mira_gazebo.urdf'"/>
4
5      <arg name="x" default="0"/>
6      <arg name="y" default="0"/>
7      <arg name="z" default="0.2"/>
8
9      <include file="$(find gazebo_ros)/launch/empty_world.launch">
10         <arg name="paused" value="true"/>
11         <arg name="use_sim_time" value="true"/>
12         <arg name="gui" value="true"/>
13     </include>
14
15     <node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" output="screen"
16           args="-urdf -param robot_description -model mira -x $(arg x) -y $(arg y) -z $(arg z)" />
17
18 </launch>

```



2.2 gazebo 관련 pkg 설치

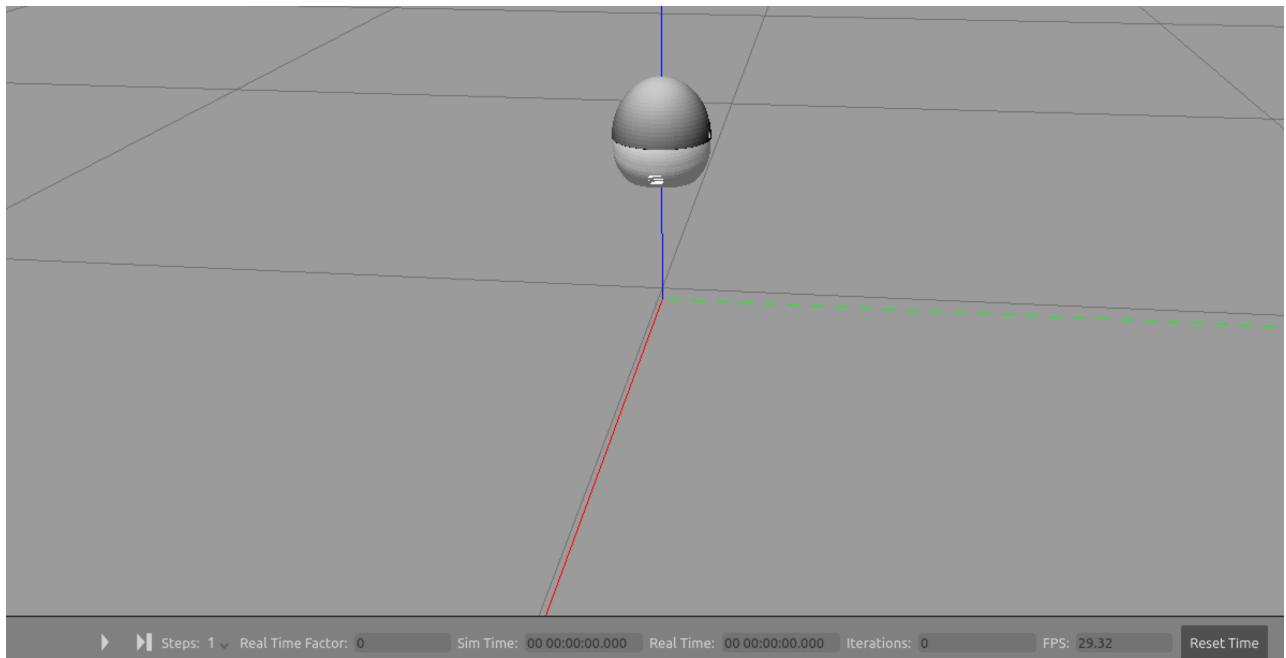
```

pw@melodic:~/ws/src/urdf_exam/urdf$ sudo apt install ros-melodic-gazebo-
ros-melodic-gazebo-dev          ros-melodic-gazebo-ros-control
ros-melodic-gazebo-msgs          ros-melodic-gazebo-ros-control-dbgsym
ros-melodic-gazebo-plugins       ros-melodic-gazebo-ros-dbgsym
ros-melodic-gazebo-plugins-dbgsym ros-melodic-gazebo-ros-pkgs
ros-melodic-gazebo-ros
pw@melodic:~/ws/src/urdf_exam/urdf$ sudo apt install ros-melodic-gazebo-* ←
[sudo] password for pw:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ros-melodic-gazebo-ros-control' for glob 'ros-melodic-gazebo-*'
Note, selecting 'ros-melodic-gazebo-ros-dbgsym' for glob 'ros-melodic-gazebo-*'
Note, selecting 'ros-melodic-gazebo-plugins' for glob 'ros-melodic-gazebo-*'
Note, selecting 'ros-melodic-gazebo-ros_pkgs' for glob 'ros-melodic-gazebo-*'

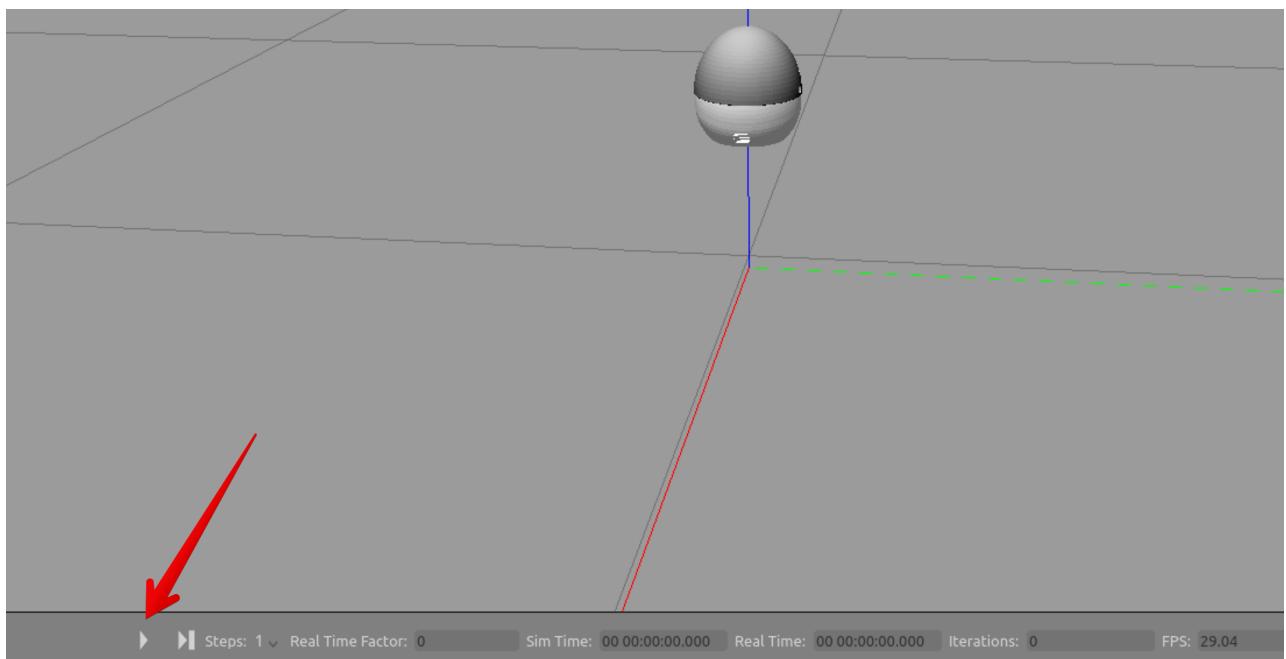
```

2.3 실행

- roslaunch urdf_exam mira_spawn.launch



2.4 play



2.5 터미널에 이런 에러가 나타난다면

```

[INFO] [1590239231.562443, 0.000000]: Waiting for service /gazebo/spawn_urdf_model
[INFO] [1590239232.089949307]: waitForService: Service [/gazebo/set_physics_properties] is now available.
[INFO] [1590239232.110633801]: Physics dynamic reconfigure ready.
[INFO] [1590239232.165839, 0.000000]: Calling service /gazebo/spawn_urdf_model
[INFO] [1590239232.293912, 0.000000]: Spawn status: SpawnModel: Successfully spawned entity
spawn_urdf-4] process has finished cleanly
log file: /home/pw/.ros/log/4fc4e860-9cf6-11ea-9a72-001c420be203/spawn_urdf-4*.log
[Err] [REST.cc:205] Error in REST request ←
libcurl: (51) SSL: no alternative certificate subject name matches target host name 'api.ignitionfuel.org'

```

2.6 ~/.ignition/fuel/config.yaml 을 열어서

```

ROS1 is activated
pw@melodic:~$ subl ~/.ignition/fuel/config.yaml
pw@melodic:~$ 

```

2.7 표시된 url을

```

1  ---
2  # The list of servers.
3  servers:
4  -
5  |   name: osrf
6  |   url: https://api.ignitionfuel.org ←
7  #
8  |   # name: another_server
9  |   # url: https://myserver
10 |
11
12 # Where are the assets stored in disk.
13 # cache:
14 #   path: /tmp/ignition/fuel
15

```



2.8 api.ignitionrobotics.org 로 변경

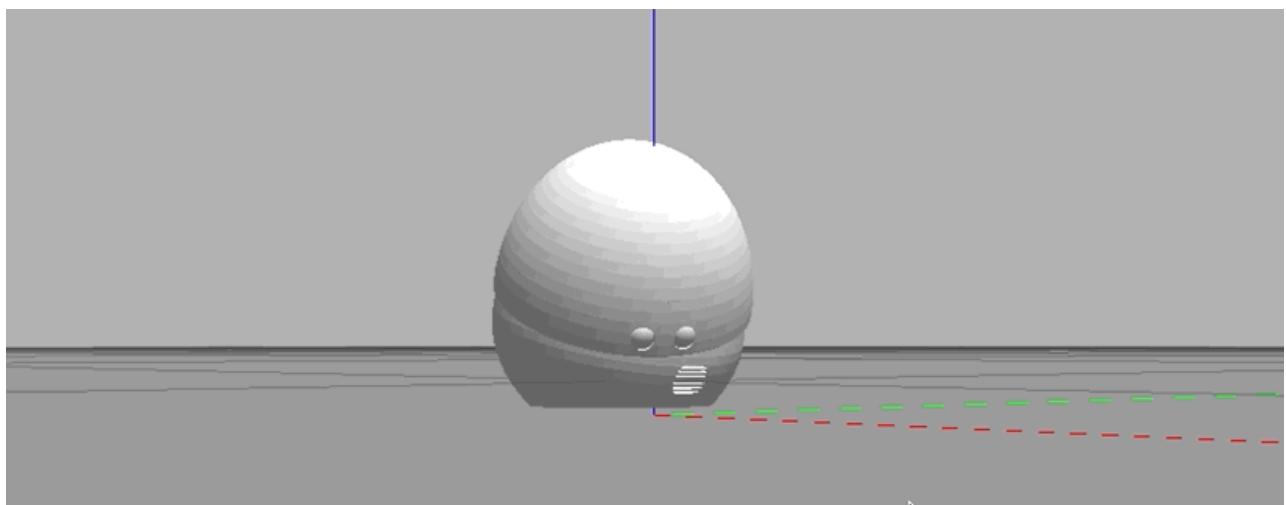
```

1  ---
2  # The list of servers.
3  servers:
4  -
5      name: osrf
6      url: https://api.ignitionrobotics.org ←
7  #
8      # name: another_server
9      # url: https://myserver
10
11
12 # Where are the assets stored in disk.
13 # cache:
14 #   path: /tmp/ignition/fuel

```

2.9 그리고 다시 실행하면 에러가 사라지고 없다

2.10 아마 물리엔진을 살짝 구경할 수 있을 겁니다.



2.11 mira_spawn.launch

```

<?xml version="1.0" encoding="UTF-8"?>
<launch>
    <param name="robot_description" command="cat '$(find urdf_exam)/urdf/mira_gazebo.urdf'"/>

    <arg name="x" default="0"/>
    <arg name="y" default="0"/>
    <arg name="z" default="0.2"/>

    <include file="$(find gazebo_ros)/launch/empty_world.launch">

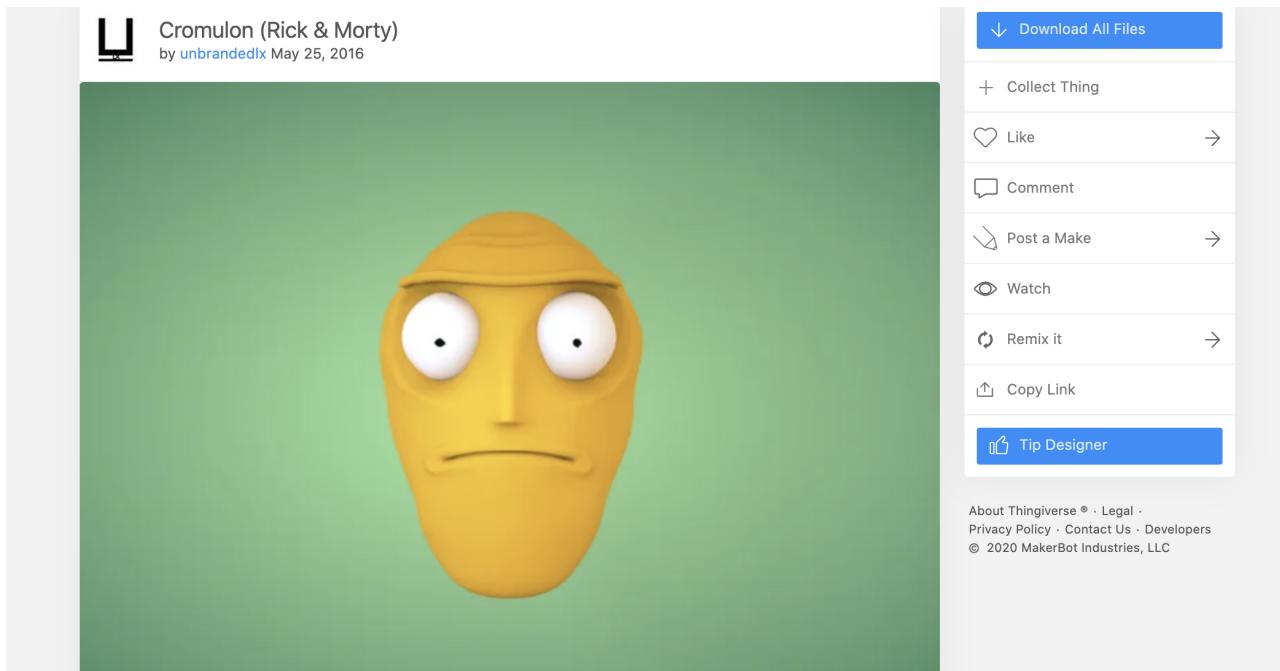
```

```
<arg name="paused" value="true"/>
<arg name="use_sim_time" value="true"/>
<arg name="gui" value="true"/>
</include>

<node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" output="screen"
      args="-urdf -param robot_description -model mira -x $(arg x) -y $(arg y) -z $(arg z)" />

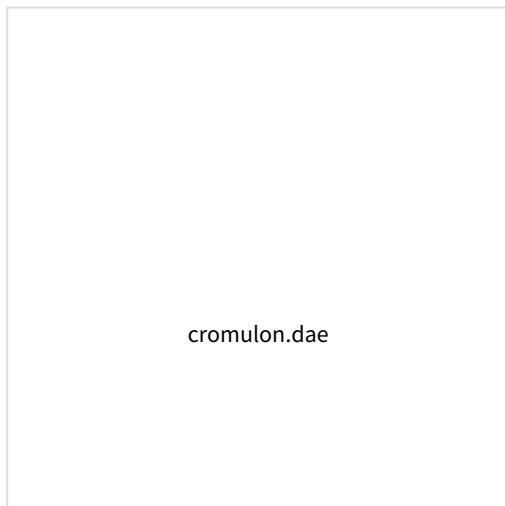
</launch>
```

2.12 보너스 - 재미를 찾아보자

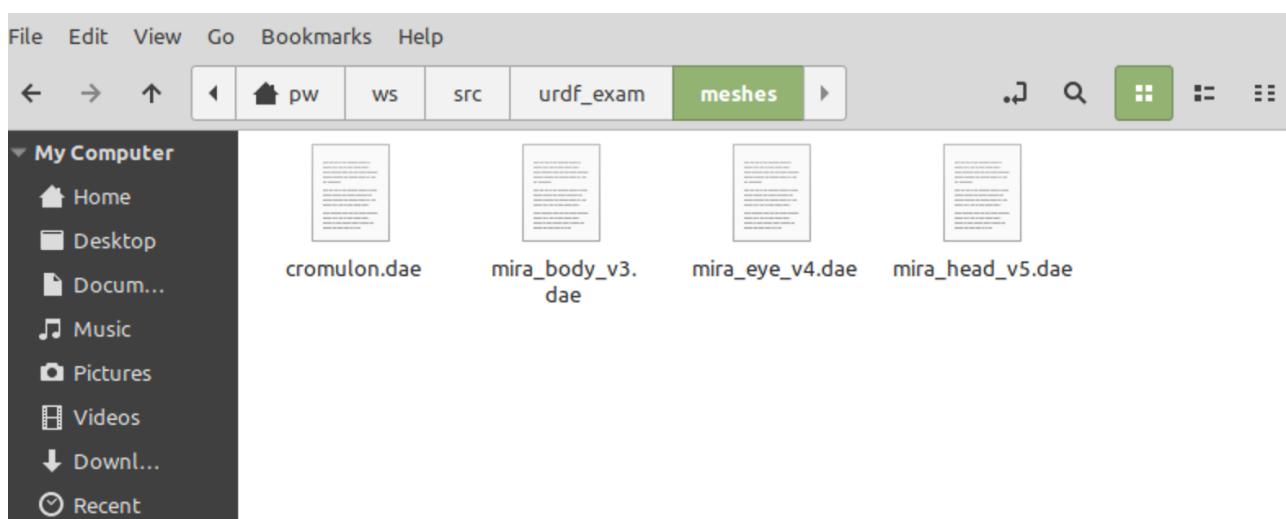


- <https://www.thingiverse.com/thing:1587203>

2.13 dae 파일



2.14 복사하고

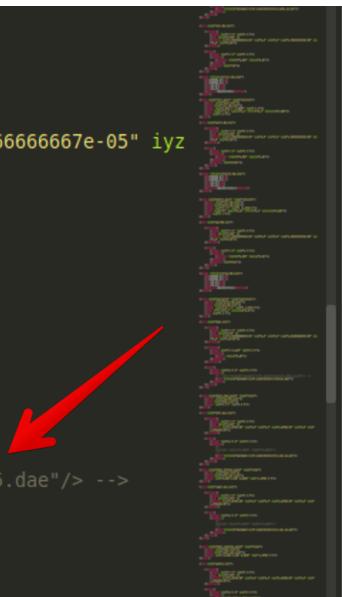


2.15 head_link 부분 수정

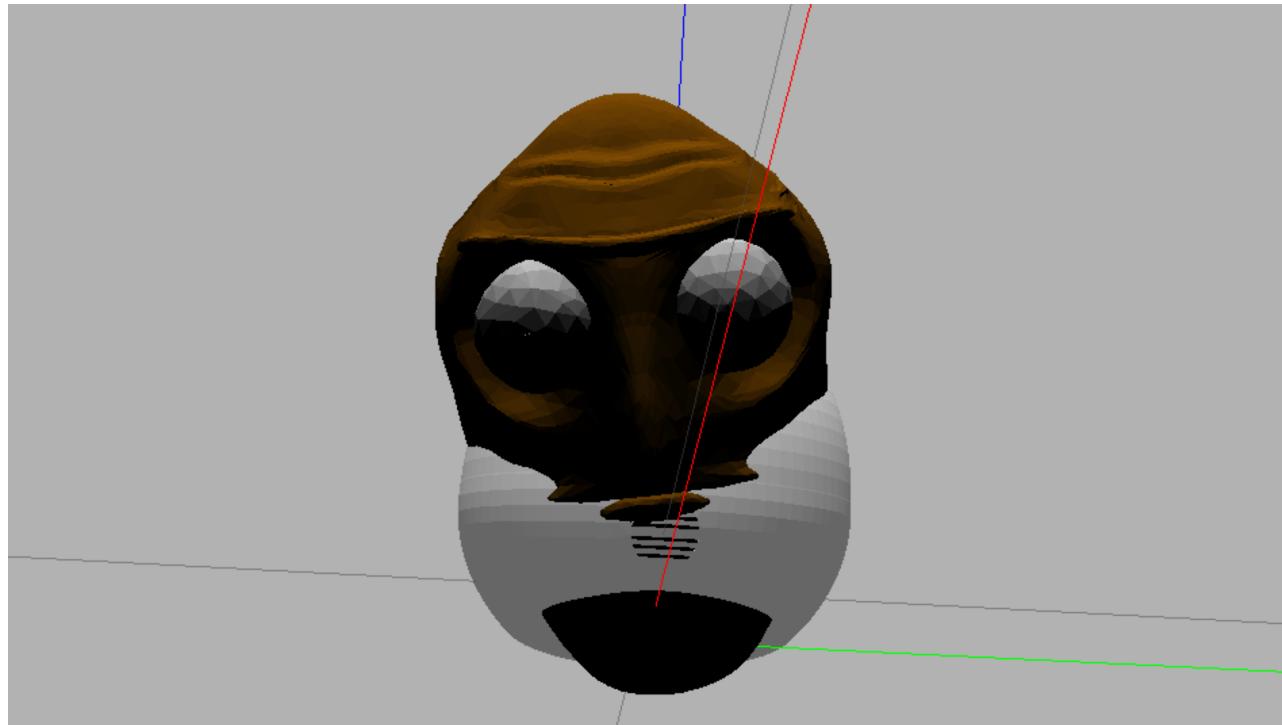
```

149   <link name="head_link">
150
151     <inertial>
152       <origin xyz="0 0 0" rpy="0 0 0"/>
153       <mass value="0.02" />
154       <inertia ixx="2.61666666667e-05" ixy="0.0" ixz="0.0" iyy="2.61666666667e-05" iyz
155         ="0.0" izz="3.6e-05"/>
156     </inertial>
157
158     <collision>
159       <origin xyz="0 0 0.015" rpy="0 0 0"/>
160       <geometry>
161         <sphere radius="0.06"/>
162       </geometry>
163     </collision>
164
165     <visual>
166       <origin rpy="0.0 0 0" xyz="0 0 0"/>
167       <geometry>
168         <!-- <mesh filename="package://urdf_exam/meshes/mira_head_v5.dae"/> -->
169         <mesh filename="package://urdf_exam/meshes/cromulon.dae"/>
170       </geometry>
171     </visual>
172   </link>

```



2.16 약간 무섭나요?



3 구동축 만들어 주기

3.1 urdf 파일을 복사해서 만들어서 시작하자

```
pw@melodic:~/ws/src/urdf_exam$ cp mira_gazebo.urdf mira_gazebo_controller.urdf
pw@melodic:~/ws/src/urdf_exam$ ls
mira_gazebo_controller.urdf  mira_gazebo.urdf  mira_simple.urdf
pw@melodic:~/ws/src/urdf_exam$
```

3.2 joint tag

```
73      <joint name="roll_joint" type="revolute">
74          <parent link="base_link"/>
75          <child link="roll_M1_link"/>
76          <origin xyz="0.0023 0 -0.0005" rpy="0 0 0"/>
77          <limit lower="-0.2" upper="0.2" effort="0.1" velocity="0.005"/>
78          <axis xyz="1 0 0"/>
79      </joint>
```

- 조인트의 한계와 속도, 힘과 관련된 부분의 설정
- 그러나 딱 맞아 떨어지지는 않아서 자주 튜닝이 필요하다

3.3 ROS controllers

- **effort_controllers**: Torque / current based controller
 - *joint_effort_controller*
 - *joint_position_controller*
 - *joint_velocity_controller*
- **position_controllers**
 - *joint_position_controller*
- **velocity_controllers**
 - *joint_velocity_controller*
- **joint_state_controller**
 - *joint_state_controller*



3.4 roll_joint에 액추에이터 추가

```

73   <joint name="roll_joint" type="revolute">
74     <parent link="base_link"/>
75     <child link="roll_M1_link"/>
76     <origin xyz="0.0023 0 -0.0005" rpy="0 0 0"/>
77     <limit lower="-0.2" upper="0.2" effort="0.1" velocity="0.005"/>
78     <axis xyz="1 0 0"/>
79   </joint>
80
81   <transmission name="tran1">
82     <type>transmission_interface/SimpleTransmission</type>
83     <joint name="roll_joint">
84       <hardwareInterface>EffortJointInterface</hardwareInterface>
85     </joint>
86     <actuator name="motor1">
87       <hardwareInterface>EffortJointInterface</hardwareInterface>
88       <mechanicalReduction>1</mechanicalReduction>
89     </actuator>
90   </transmission>
91
92   <link name="pitch_M2_link">
93

```

```

<transmission name="tran1">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="roll_joint">
    <hardwareInterface>EffortJointInterface</hardwareInterface>
  </joint>
  <actuator name="motor1">
    <hardwareInterface>EffortJointInterface</hardwareInterface>
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
</transmission>

```

3.5 pitch_joint에 엑추에이터 추가

```

118    <joint name="pitch_joint" type="revolute">
119        <parent link="roll_M1_link"/>
120        <child link="pitch_M2_link"/>
121        <origin xyz="0 0 0" rpy="0 -1.5708 0"/>
122        <limit lower="0" upper="0.44" effort="0.1" velocity="0.005"/>
123        <axis xyz="0 1 0"/>
124    </joint>
125
126    <transmission name="tran2">
127        <type>transmission_interface/SimpleTransmission</type>
128        <joint name="pitch_joint">
129            <hardwareInterface>EffortJointInterface</hardwareInterface>
130        </joint>
131        <actuator name="motor2">
132            <hardwareInterface>EffortJointInterface</hardwareInterface>
133            <mechanicalReduction>1</mechanicalReduction>
134        </actuator>
135    </transmission>
136
137    <link name="yaw_M3_link">
138
139        <inertial>
140            <origin xyz="0 0 0" rpy="0 0 0"/>
141            <mass value="0.001" />
142            <inertia ixx="2.7083333333e-08" ixy="0.0" ixz="0.0" iyy="2.7083333333e-08" iyz="0.0" izz="1.4166666667e-08" />
143
144    </link>
145
146    <collision name="pitch_collision">
147        <geometry><box size="0.05 0.05 0.05"/></geometry>
148        <material><script type="file" value="material/white"></script></material>
149    </collision>
150
151    <visual name="pitch_visual">
152        <geometry><box size="0.05 0.05 0.05"/></geometry>
153        <material><script type="file" value="material/white"></script></material>
154    </visual>
155
156</robot>

```

126 ~ 135 행

```

<transmission name="tran2">
    <type>transmission_interface/SimpleTransmission</type>
    <joint name="pitch_joint">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
    </joint>
    <actuator name="motor2">
        <hardwareInterface>EffortJointInterface</hardwareInterface>
        <mechanicalReduction>1</mechanicalReduction>
    </actuator>
</transmission>

```

3.6 yaw_joint에 엑츄에이터 추가

```

63   <joint name="yaw_joint" type="continuous">
64     <parent link="pitch_M2_link"/>
65     <child link="yaw_M3_link"/>
66     <origin xyz="0.01 0 0" rpy="0 1.5708 0"/>
67     <limit effort="0.1" velocity="0.01"/>
68     <axis xyz="0 0 1"/>
69   </joint>
70
71   <transmission name="tran3">
72     <type>transmission_interface/SimpleTransmission</type>
73     <joint name="yaw_joint">
74       <hardwareInterface>EffortJointInterface</hardwareInterface>
75     </joint>
76     <actuator name="motor3">
77       <hardwareInterface>EffortJointInterface</hardwareInterface>
78       <mechanicalReduction>1</mechanicalReduction>
79     </actuator>
80   </transmission>
81
82
83   <link name="head_link">

```

```

<transmission name="tran3">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="yaw_joint">
    <hardwareInterface>EffortJointInterface</hardwareInterface>
  </joint>
  <actuator name="motor3">
    <hardwareInterface>EffortJointInterface</hardwareInterface>
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
</transmission>

```

3.7 gazebo 제어기 플러그인 추가

```

18   </material>
19   <material name="black">
20     <color rgba="0 0 0 1"/>
21   </material>
22
23   <gazebo>
24     <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
25       <robotNamespace>/mira</robotNamespace>
26     </plugin>
27   </gazebo>
28
29
30   <link name="base_link">
31
32     <inertial>
33       <origin xyz="0 0 0" rpy="0 0 0"/>

```

```

<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    <robotNamespace>/mira</robotNamespace>

```

```
</plugin>
</gazebo>
```

3.8 config 폴더 하나 추가

```

    urdf_exam
      config (red arrow) <--> mira_gazebo.urdf
      launch
      meshes
      scripts
      src
      urdf
      CMakeLists.txt
      package.xml
      urdf.rviz
  
```

```

    7   <material name="red">
  8     <color rgba="0.8 0 0 1"/>
  9   </material>
 10   <material name="green">
 11     <color rgba="0 0.8 0 1"/>
 12   </material>
 13   <material name="grey">
 14     <color rgba="0.75 0.75 0.75 1"/>
 15   </material>
 16   <material name="white">
 17     <color rgba="1.0 1.0 1.0 1"/>
 18   </material>
 19   <material name="black">
 20     <color rgba="0 0 0 1"/>
 21   </material>
 22
  
```

3.9 각 조인트 제어기 설정

FOLDERS

```

    urdf_exam
      config (red arrow) <--> mira_gazebo.urdf
      launch
      meshes
      scripts
      src
      urdf
      CMakeLists.txt
      package.xml
      urdf.rviz
  
```

mira_gazebo.urdf

```

  1 mira:
  2   # Publish all joint states -----
  3   joint_state_controller:
  4     type: joint_state_controller/JointStateController
  5     publish_rate: 50
  6
  7   # Position Controllers -----
  8   roll_joint_position_controller:
  9     type: effort_controllers/JointPositionController
 10    joint: roll_joint
 11    pid: {p: 1.0, i: 1.0, d: 0.0}
 12    pitch_joint_position_controller:
 13      type: effort_controllers/JointPositionController
 14      joint: pitch_joint
 15      pid: {p: 1.0, i: 1.0, d: 0.0}
 16    yaw_joint_position_controller:
 17      type: effort_controllers/JointPositionController
 18      joint: yaw_joint
 19      pid: {p: 1.0, i: 1.0, d: 0.0}
  
```

```

mira:
  # Publish all joint states -----
  joint_state_controller:
    type: joint_state_controller/JointStateController
    publish_rate: 50

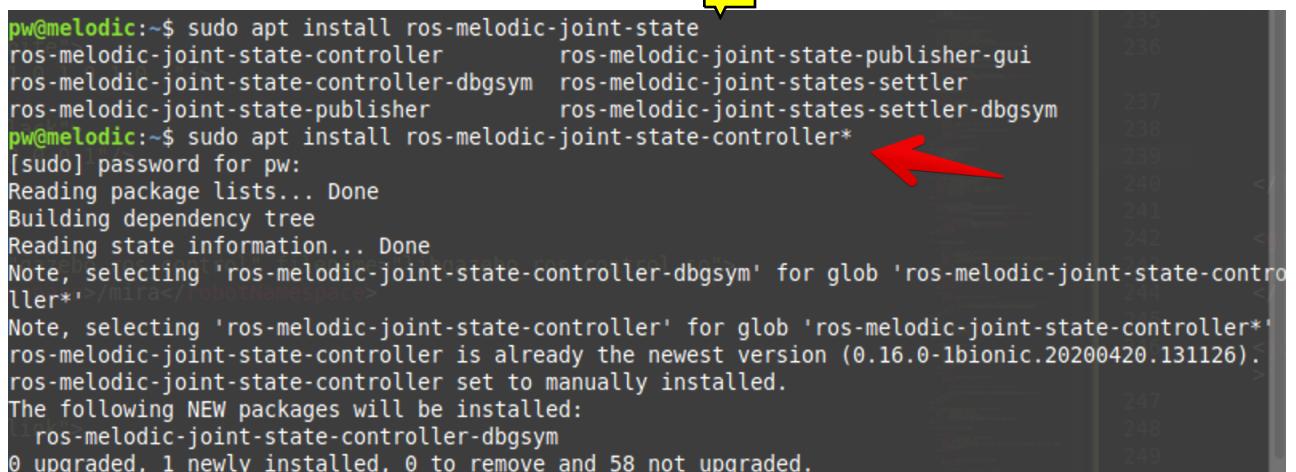
  # Position Controllers -----
  roll_joint_position_controller:
    type: effort_controllers/JointPositionController
    joint: roll_joint
    pid: {p: 1.0, i: 1.0, d: 0.0}
  
```

```

pitch_joint_position_controller:
  type: effort_controllers/JointPositionController
  joint: pitch_joint
  pid: {p: 1.0, i: 1.0, d: 0.0}
yaw_joint_position_controller:
  type: effort_controllers/JointPositionController
  joint: yaw_joint
  pid: {p: 1.0, i: 1.0, d: 0.0}

```

3.10 joint-state-controller 설치

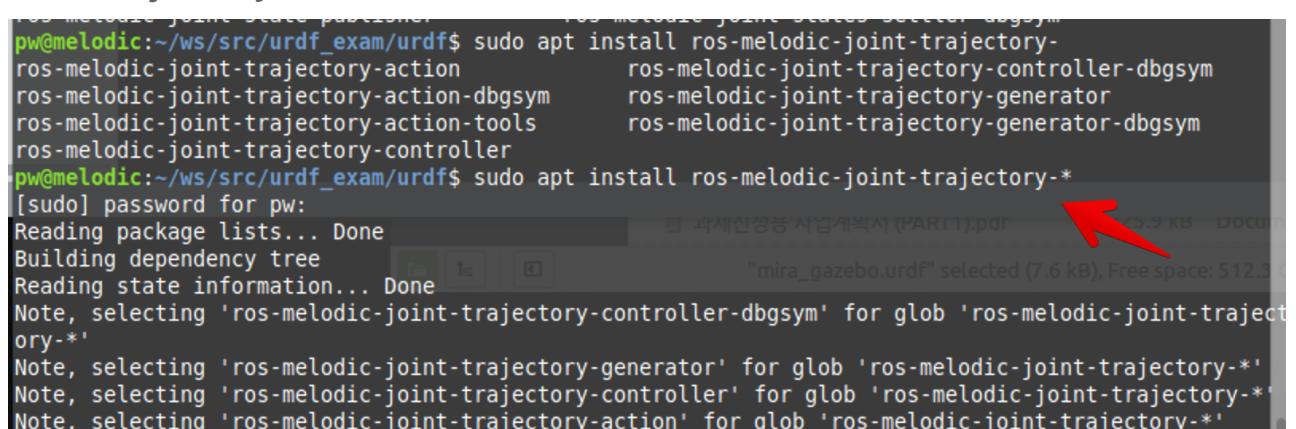


```

pw@melodic:~$ sudo apt install ros-melodic-joint-state
ros-melodic-joint-state-controller      ros-melodic-joint-state-publisher-gui
ros-melodic-joint-state-controller-dbgSYM  ros-melodic-joint-states-settler
ros-melodic-joint-state-publisher      ros-melodic-joint-states-settler-dbgSYM
pw@melodic:~$ sudo apt install ros-melodic-joint-state-controller*
[sudo] password for pw:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ros-melodic-joint-state-controller-dbgSYM' for glob 'ros-melodic-joint-state-controller*'
Note, selecting 'ros-melodic-joint-state-controller' for glob 'ros-melodic-joint-state-controller*'
ros-melodic-joint-state-controller is already the newest version (0.16.0-1bionic.20200420.131126).
ros-melodic-joint-state-controller set to manually installed.
The following NEW packages will be installed:
  ros-melodic-joint-state-controller-dbgSYM
0 upgraded, 1 newly installed, 0 to remove and 58 not upgraded.

```

3.11 trajectory-controller 설치



```

pw@melodic:~/ws/src/urdf_exam/urdf$ sudo apt install ros-melodic-joint-trajectory-
ros-melodic-joint-trajectory-action      ros-melodic-joint-trajectory-controller-dbgSYM
ros-melodic-joint-trajectory-action-dbgSYM  ros-melodic-joint-trajectory-generator
ros-melodic-joint-trajectory-action-tools   ros-melodic-joint-trajectory-generator-dbgSYM
ros-melodic-joint-trajectory-controller
pw@melodic:~/ws/src/urdf_exam/urdf$ sudo apt install ros-melodic-joint-trajectory-*
[sudo] password for pw:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ros-melodic-joint-trajectory-controller-dbgSYM' for glob 'ros-melodic-joint-trajectory-*'
Note, selecting 'ros-melodic-joint-trajectory-generator' for glob 'ros-melodic-joint-trajectory-*'
Note, selecting 'ros-melodic-joint-trajectory-controller' for glob 'ros-melodic-joint-trajectory-*'
Note, selecting 'ros-melodic-joint-trajectory-action' for glob 'ros-melodic-joint-trajectory-*'

```

3.12 effort controller 설치

```
pw@melodic:~/ws/src/urdf_exam/urdf$ sudo apt install ros-melodic-effort-controllers
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ros-melodic-effort-controllers
0 upgraded, 1 newly installed, 0 to remove and 58 not upgraded.
Need to get 80.0 kB of archives.
After this operation, 339 kB of additional disk space will be used.
Get:1 http://packages.ros.org/ros/ubuntu bionic/main amd64 ros-melodic-effort-controllers amd64 0.16.0-1bionic.20200420.132256 [80.0 kB]
Fetched 80.0 kB in 4s (21.4 kB/s)
Selecting previously unselected package ros-melodic-effort-controllers.
```



3.13 mira_gazebo_controller.urdf를 만들자

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <launch>
3    <param name="robot_description" command="cat '$(find urdf_exam)/urdf/mira_gazebo_controller.urdf'"/>
4
5    <arg name="x" default="0"/>
6    <arg name="y" default="0"/>
7    <arg name="z" default="0.045"/>
8
9    <include file="$(find gazebo_ros)/launch/empty_world.launch">
10      <arg name="paused" value="true"/>
11      <arg name="use_sim_time" value="true"/>
12      <arg name="gui" value="true"/>
13    </include>
14
15    <node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" output="screen"
16      args="-urdf -param robot_description -model mira -x $(arg x) -y $(arg y) -z $(arg z)" />
17
18    <rosparam file="$(find urdf_exam)/config/mira_control.yaml" command="load"/>
19
20    <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false"
21      output="screen" ns="/mira" args="roll_joint_position_controller pitch_joint_position_controller
yaw_joint_position_controller joint_state_controller"/>
22
23    <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
24      respawn="false" output="screen">
25      <remap from="/joint_states" to="/mira/joint_states" />
26    </node>
27
28  </launch>
```



mira_spawn_controller.la...

mira_gazebo_controller.u...

(see page 5)

(see page 5)

```

<?xml version="1.0" encoding="UTF-8"?>
<launch>
  <param name="robot_description" command="cat '$(find urdf_exam)/urdf/mira_gazebo_controller.urdf'" />

  <arg name="x" default="0"/>
  <arg name="y" default="0"/>
  <arg name="z" default="0.045"/>

  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="paused" value="true"/>
    <arg name="use_sim_time" value="true"/>
    <arg name="gui" value="true"/>
  </include>

  <node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" output="screen"
        args="-urdf -param robot_description -model mira -x $(arg x) -y $(arg y) -z $(arg z)" />

  <rosparam file="$(find urdf_exam)/config/mira_control.yaml" command="load"/>

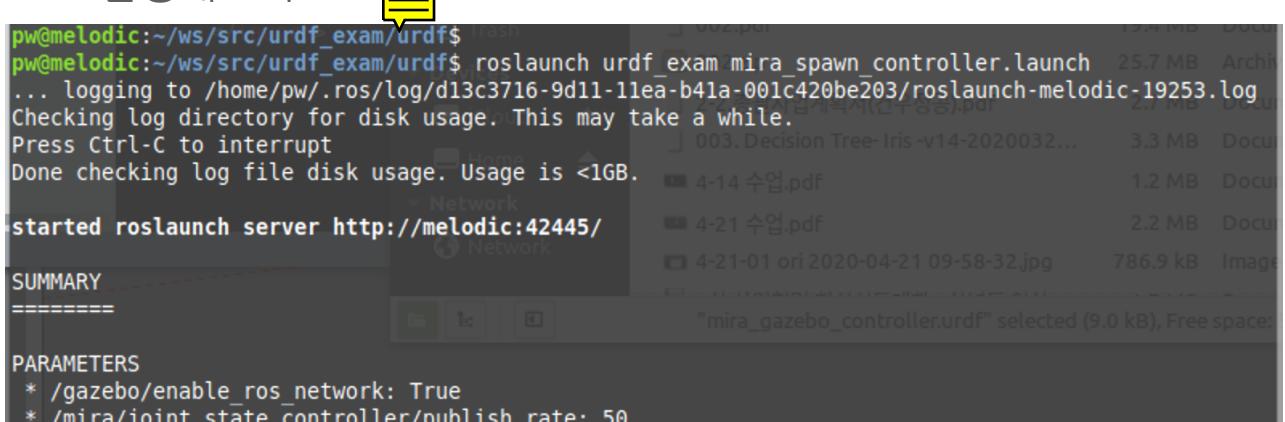
  <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false"
        output="screen" ns="/mira" args="roll_joint_position_controller pitch_joint_position_controller
yaw_joint_position_controller joint_state_controller"/>

  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
        respawn="false" output="screen">
    <remap from="/joint_states" to="/mira/joint_states" />
  </node>

</launch>

```

3.14 실행해보자



```

pw@melodic:~/ws/src/urdf_exam/urdfs$ roslaunch urdf_exam mira_spawn_controller.launch
... logging to /home/pw/.ros/log/d13c3716-9d11-11ea-b41a-001c420be203/roslaunch-melodic-19253.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:42445/

```

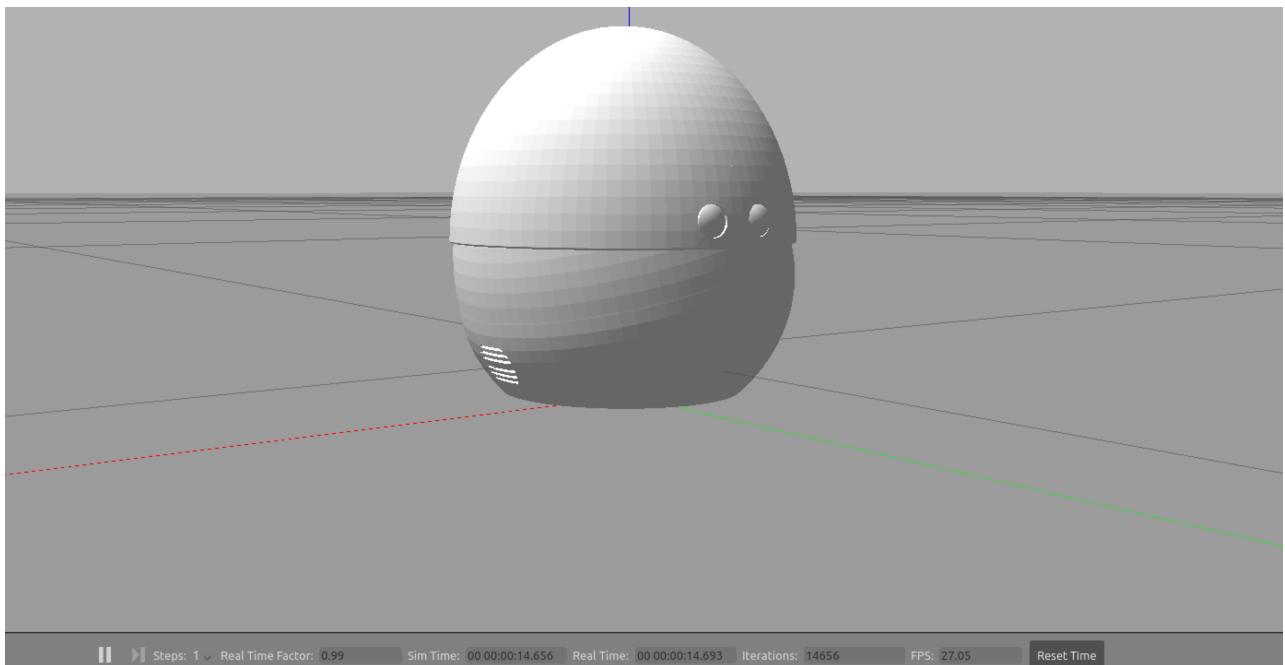
SUMMARY

```

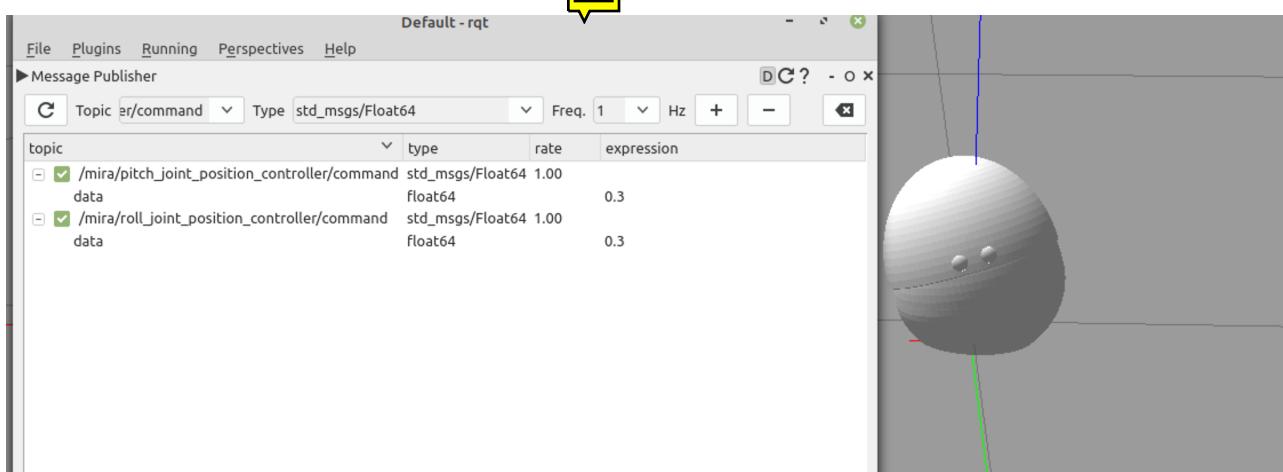
PARAMETERS
* /gazebo/enable_ros_network: True
* /mira/joint_state_controller/publish_rate: 50

```

3.15 이제 버틴다

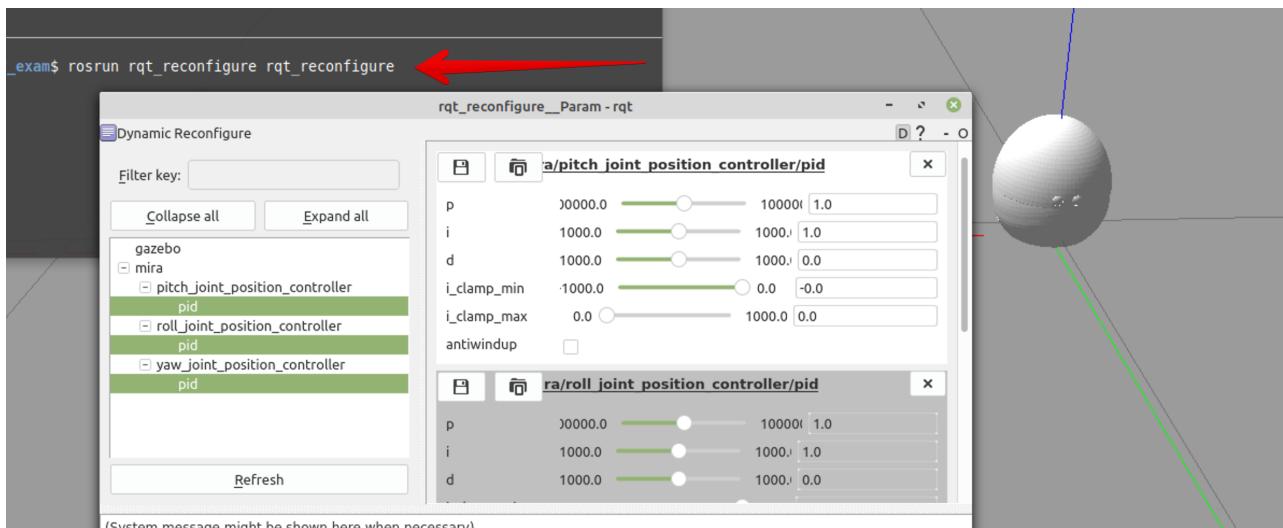


3.16 rosrun rqt_gui rqt_gui



- topic을 발행해서 각도를 결정할 수 있다

3.17 만약 각 관절의 PID값을 실시간으로 변경하고 싶다면



4 GAZEBO에서 센서 사용하기

4.1 urdf에 카메라 plugin 추가



```

297     </visual>
298   </link>
299
300   <gazebo reference="camera_link">
301     <sensor type="camera" name="camera1">
302       <update_rate>15.0</update_rate>
303       <camera name="head">
304
305         <pose>0 0 0 0 0 1.57</pose>
306
307         <horizontal_fov>1.3962634</horizontal_fov>
308         <image>
309           <width>400</width>
310           <height>400</height>
311           <format>R8G8B8</format>
312         </image>
313         <clip>
314           <near>0.01</near>
315           <far>100</far>
316         </clip>
317         <noise>
318           <type>gaussian</type>
319           <stddev>0.007</stddev>
320         </noise>
321       </camera>
322       <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
323         <alwaysOn>true</alwaysOn>
324         <updateRate>0.0</updateRate>
325         <cameraName>mira/camera1</cameraName>
326         <imageTopicName>image_raw</imageTopicName>
327         <cameraInfoTopicName>camera_info</cameraInfoTopicName>
328         <frameName>camera_link</frameName>

```

```

<gazebo reference="camera_link">
  <sensor type="camera" name="camera1">
    <update_rate>15.0</update_rate>
    <camera name="head">

      <pose>0 0 0 0 0 1.57</pose>

      <horizontal_fov>1.3962634</horizontal_fov>
      <image>
        <width>400</width>
        <height>400</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.01</near>
        <far>100</far>
      </clip>
      <noise>
        <type>gaussian</type>
        <stddev>0.007</stddev>
      </noise>
    </camera>
    <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
      <alwaysOn>true</alwaysOn>
      <updateRate>0.0</updateRate>
      <cameraName>mira/camera1</cameraName>
      <imageTopicName>image_raw</imageTopicName>
      <cameraInfoTopicName>camera_info</cameraInfoTopicName>
      <frameName>camera_link</frameName>

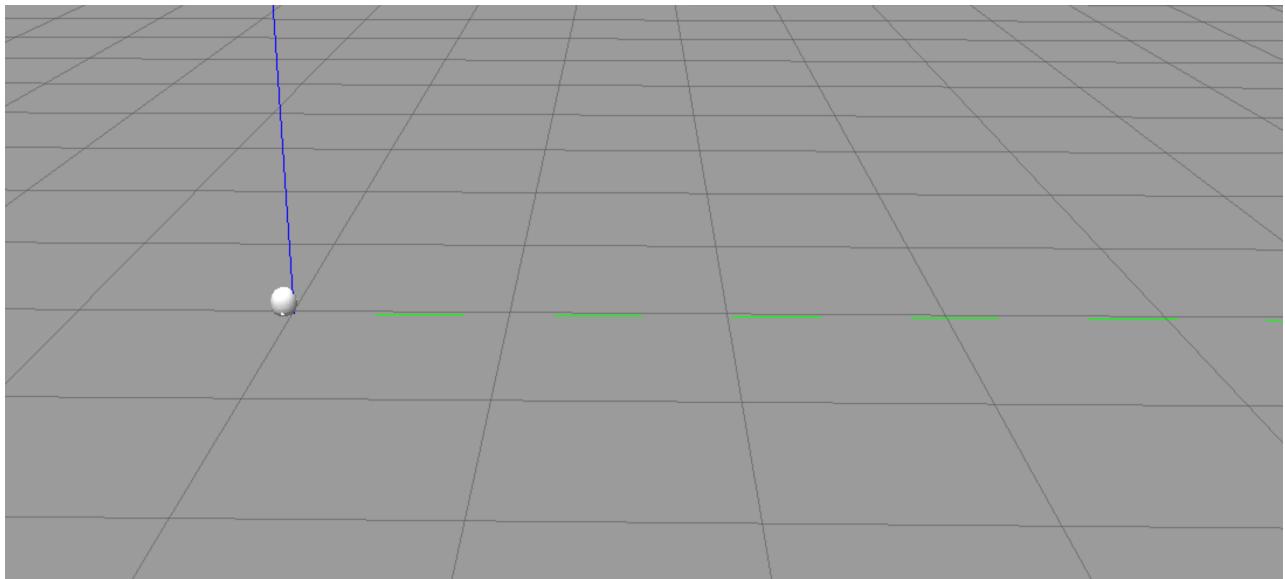
```

```

<hackBaseline>0.07</hackBaseline>
<distortionK1>0.0</distortionK1>
<distortionK2>0.0</distortionK2>
<distortionK3>0.0</distortionK3>
<distortionT1>0.0</distortionT1>
<distortionT2>0.0</distortionT2>
</plugin>
</sensor>
</gazebo>

```

4.2 다시 rosrun urdf_exam mira_spawn_controller.launch



- 화면을 조금 넓게 배치

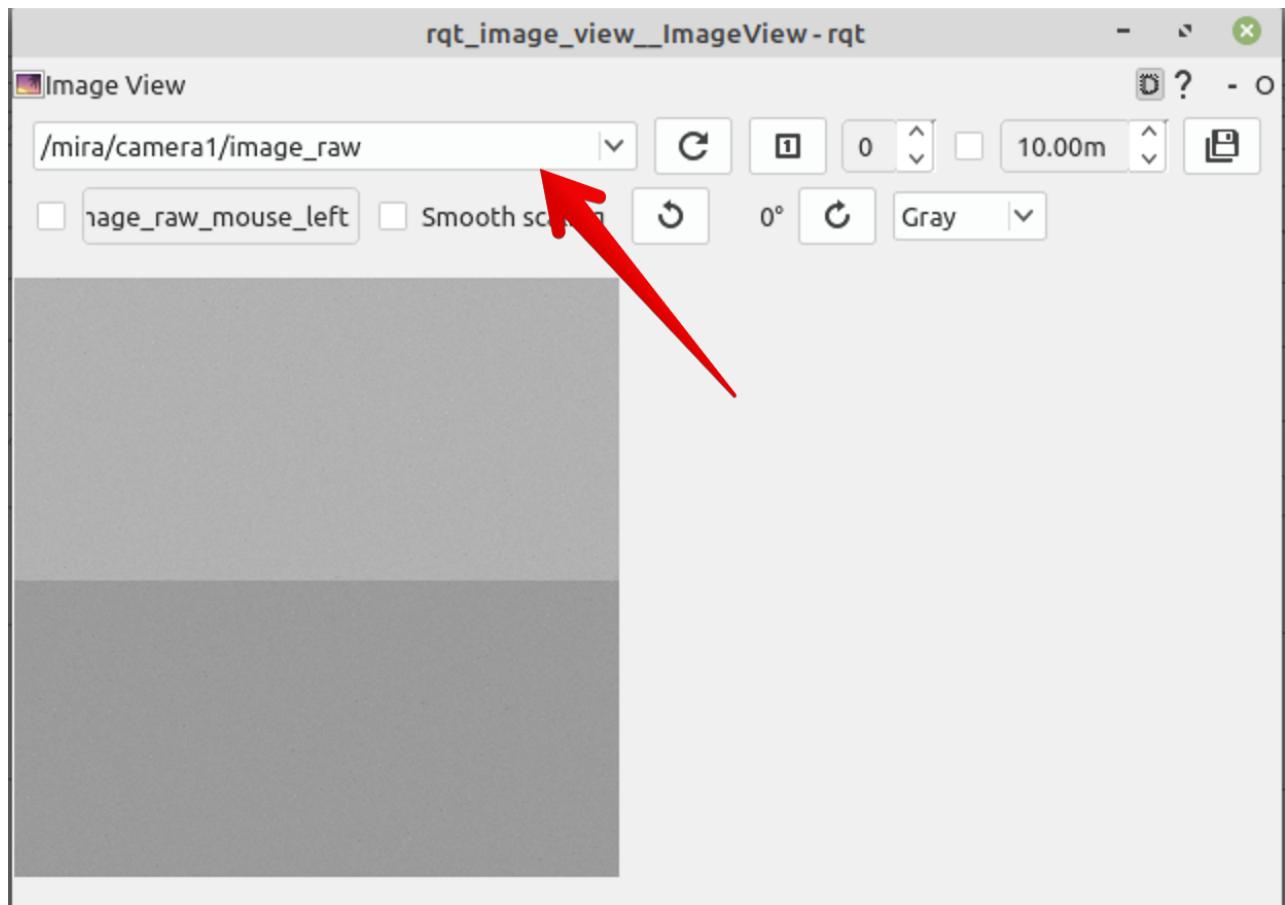
4.3 rostopic list를 보면 카메라 관련 토픽이 나타난 것이 보인다

```

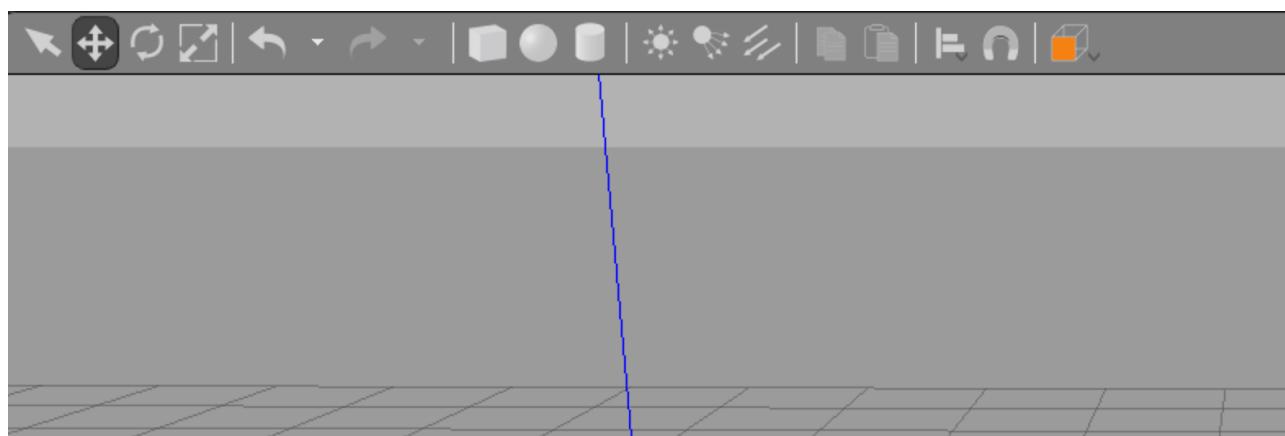
/gazebo/set_link_state
/gazebo/set_model_state
/mira/camera1/camera_info
/mira/camera1/image_raw
/mira/camera1/image_raw/compressed
/mira/camera1/image_raw/compressed/parameter_descriptions
/mira/camera1/image_raw/compressed/parameter_updates
/mira/camera1/image_raw/compressedDepth
/mira/camera1/image_raw/compressedDepth/parameter_descriptions
/mira/camera1/image_raw/compressedDepth/parameter_updates
/mira/camera1/image_raw/theora
/mira/camera1/image_raw/theora/parameter_descriptions
/mira/camera1/image_raw/theora/parameter_updates
/mira/camera1/parameter_descriptions
/mira/camera1/parameter_updates
/mira/joint_states

```

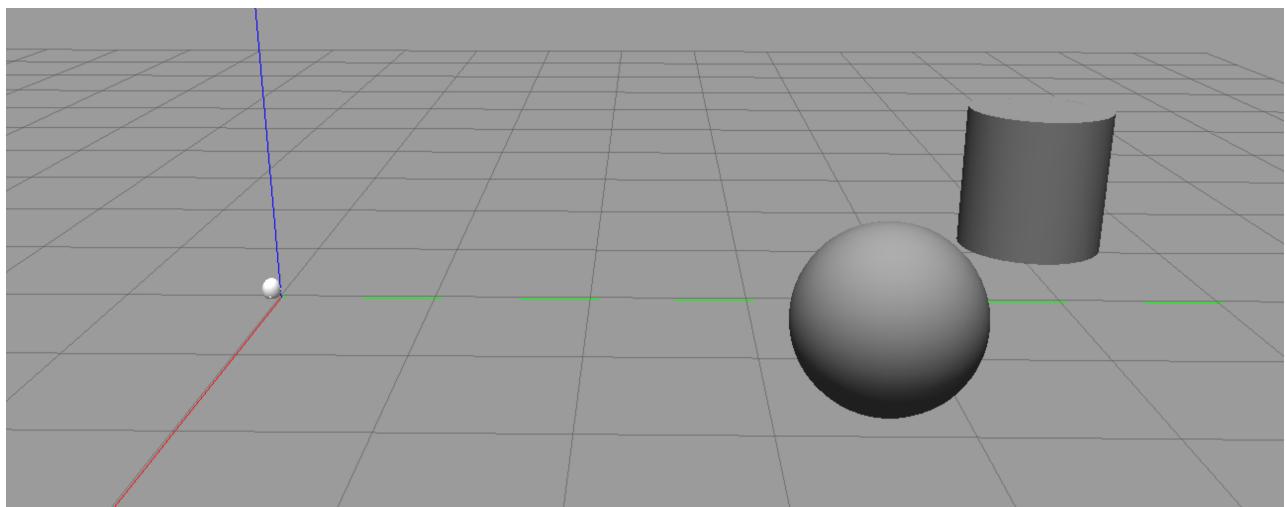
4.4 rosrun rqt_image_view rqt_image_view



4.5 적당히



4.6 배치해보자



4.7 나타난다~^_^

