NOVEMBER 4, 2018

# RoboND-Kinematics-Project

## GOALS OF THE PROJECT:

Goals of the project are to familiarize with the forward and inverse kinematics techniques, code in a python script to control a Kuka KR210 robotic arms, and use it to pick and place objects in a Gazebo simulated environment

## STEPS TOKEN TO COMPLETE PROJECT

1- Set up your ROS Workspace
2- Downloaded the project repository into the src directory of your ROS Workspace.
3- Experiment with the forward kinematics environment and get familiar with the robot.
4- Launch in demo mode.
5- Perform Kinematic Analysis for the robot following the project rubric.
6- Fill in the IK_server.py with your Inverse Kinematics code.

## RUBRIC POINTS

### KINEMATIC ANALYSIS

1. Run the forward kinematics demo and evaluate the kr210.urdf.xacro file to perform kinematic analysis of Kuka KR210 robot and derive its DH parameters.
2. Derive DH Parameters from URDF file

| **Links** | Alpha(n-1) | A(i-1) | d(i-1) | q |
|---|---|---|---|---|
| **0>1** | -pi / 2 | 0 | 0.75 | Q1 |
| **1>2** | 0 | 0.35 | 0 | Q2 |
| **2>3** | -pi / 2 | 1.25 | 0.75 | Q3 |
| **3>4** | pi / 2 | -0.0054 | 1.5 | Q4 |
| **4>5** | -pi / 2 | 0 | 0 | Q5 |
| **5>6** | 0 | 0 | 0 | Q6 |
| **6>EE** | 0 | 0 | 0.303 | 0 |

3. Create individual transformation matrices at each joint. In addition, also generate a generalized homogeneous transform between base link and gripper link using only end-effector (gripper) pose. Mathematically, the eventual homogeneous transform is expressed as:

```
T0_7 = T0_1 * T1_2 * T2_3 * T3_4 * T4_5 * T5_6* T6_7
```

4. We decoupled the IK problem into Inverse Position and Inverse orientation

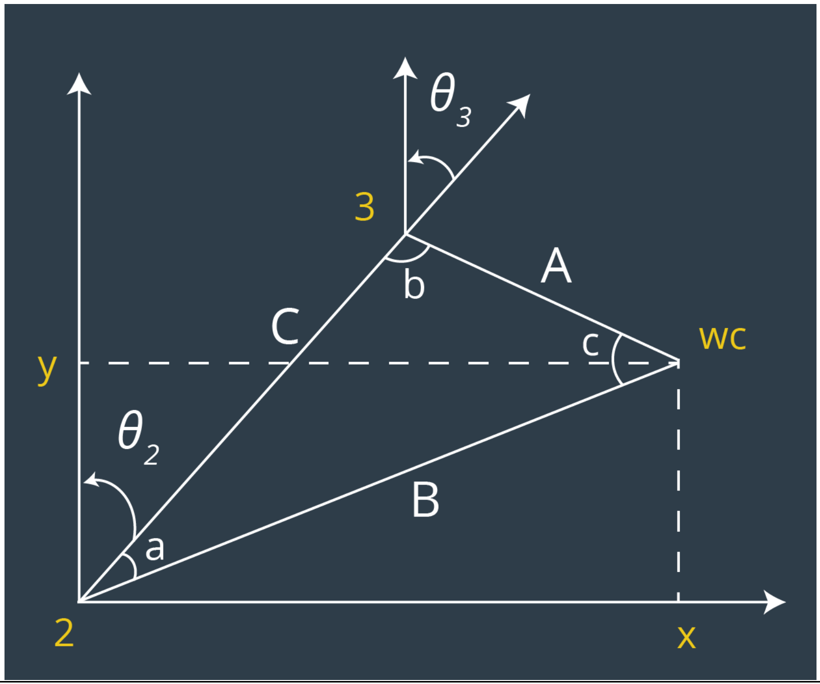
   a) At the first part we calculated the wrist center formula from the EE position

$$^{0}\boldsymbol{r}_{WC/0} = {}^{0}\boldsymbol{r}_{EE/0} - d \cdot {}^{0}_{6}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - d \cdot {}^{0}_{6}R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

```
nx = Rrpy[0, 2]
        ny = Rrpy[1, 2]
        nz = Rrpy[2, 2]

        Wc_x = px - 0.303 * nx        #d=.303
        Wc_y = py - 0.303 * ny
        Wc_z = pz - 0.303 * nz
```

   b) Then we calculated q1,q2,q3 angles formula with respect to end effector position from the geometry of the arm




```
theta1 = atan2(Wc_y, Wc_x)
theta2 = pi / 2 - a_angle - atan2(Wc_z - 0.75, R_Wc_xzPlane)
theta3 = pi / 2 - (b_angle + .036)
```

c) The following code is used to get q4-q6

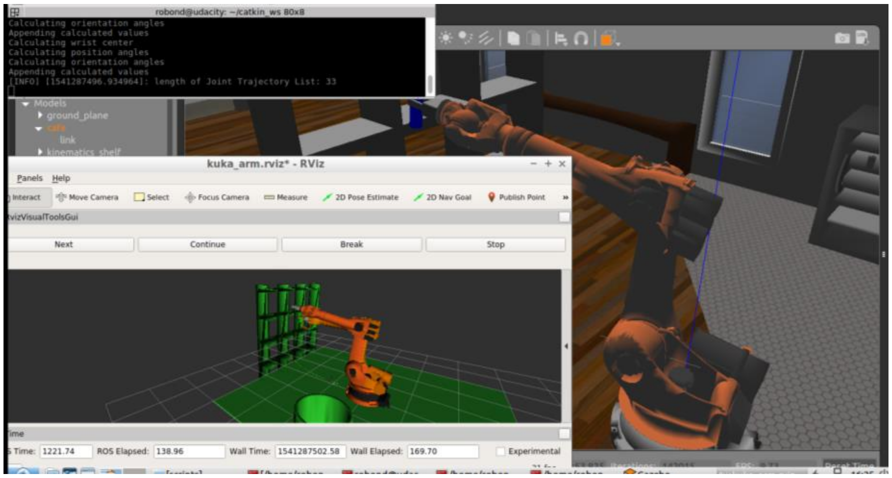
```
R0_3 = T0_3.evalf(subs={'q1': theta1, 'q2': theta2, 'q3': theta3})
R3_6 = R0_3.T * Rrpy
theta5 = atan2(sqrt(R3_6[0,2]**2 + R3_6[2,2]**2), R3_6[1,2])
# Choosing between multiple possible solutions:
if sin(theta5) < 0:
    theta4 = atan2(-R3_6[2,2], R3_6[0,2])
    theta6 = atan2(R3_6[1,1], -R3_6[1,0])
else:
    theta4 = atan2(R3_6[2,2], -R3_6[0,2])
    theta6 = atan2(-R3_6[1,1], R3_6[1,0])

print 'Appending calculated values'
```

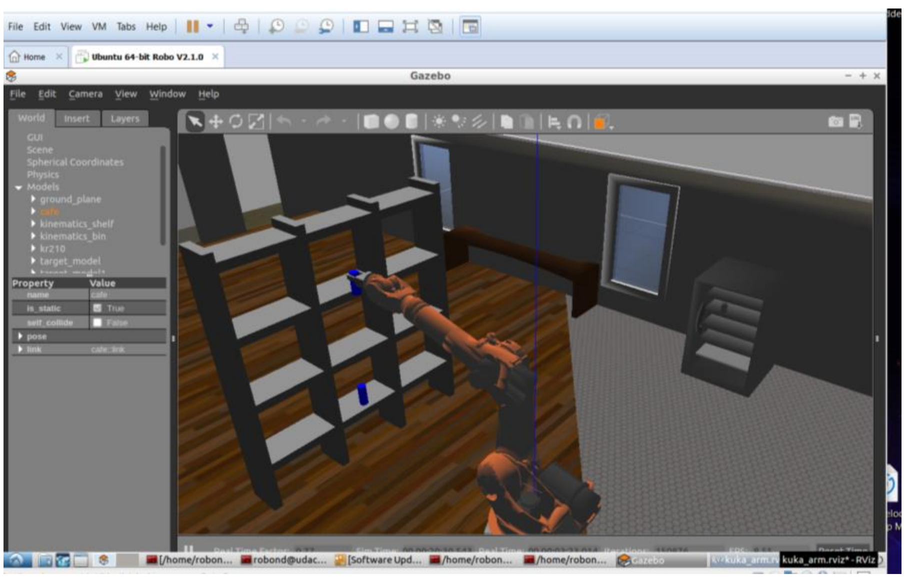
## PROJECT IMPLEMENTATION

1) After editing IK_server.py I debugged it with `IK_debug.py` and obtained satisfactory results for all test cases
2) Here is photos for the whole process

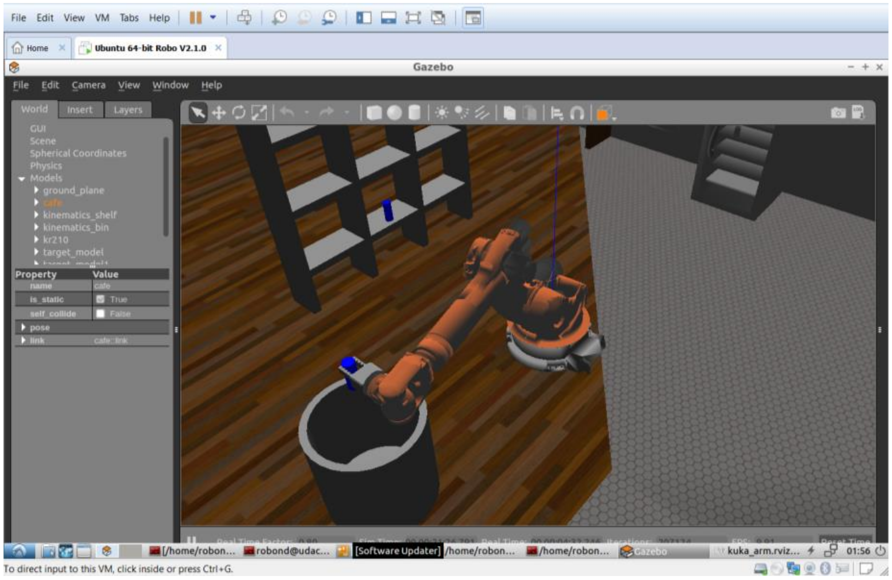
### GRASPING THE OBKECT



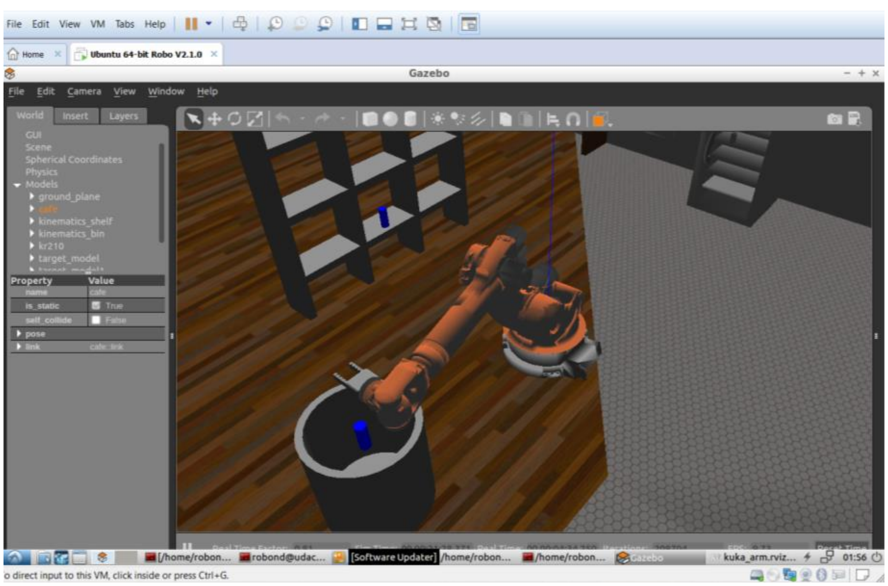

## RETRIEVE TO POSITION



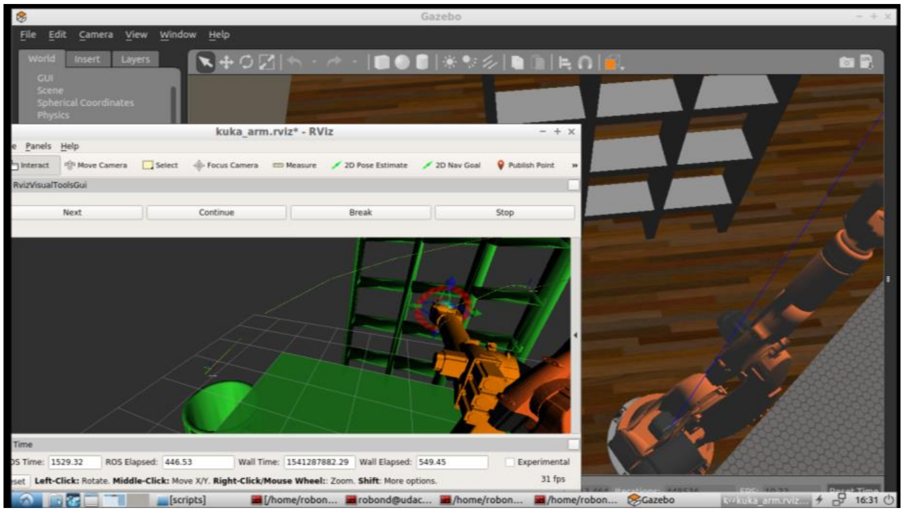

## MOVING TO THE BIN

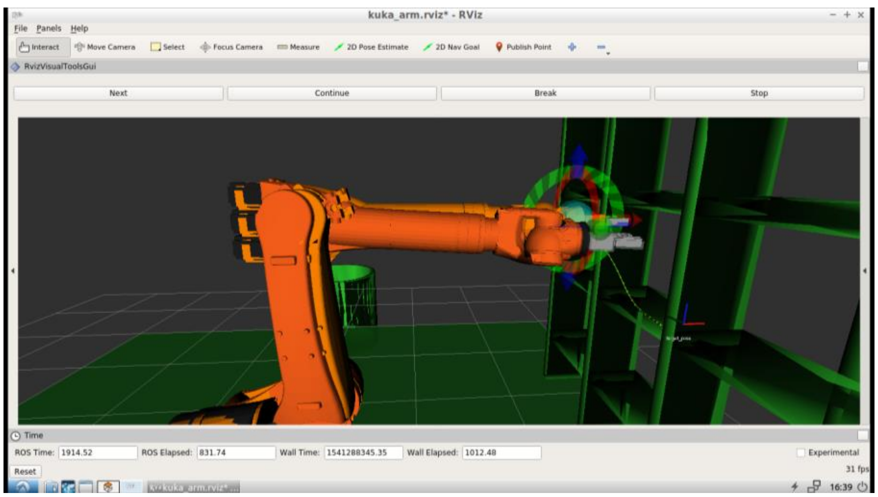


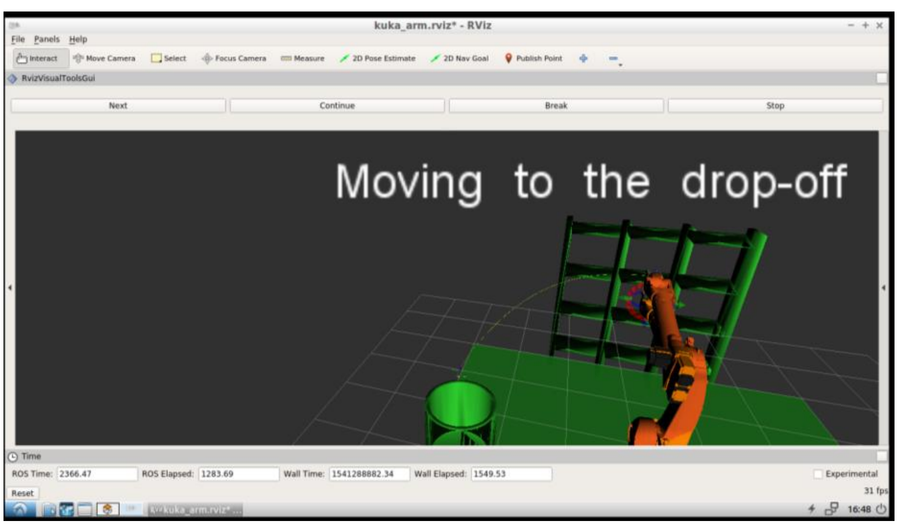
## RELEASING THE OBJECT

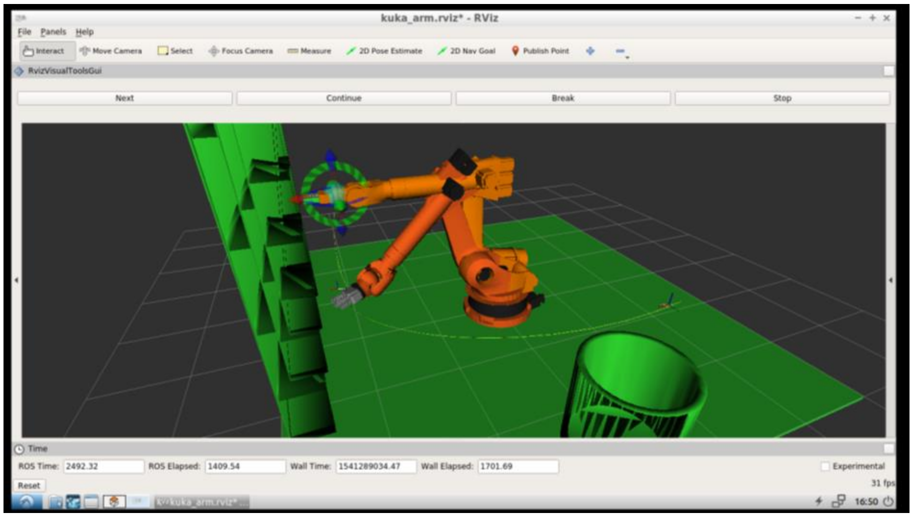


## RVIZ PATH TO THE TARGET

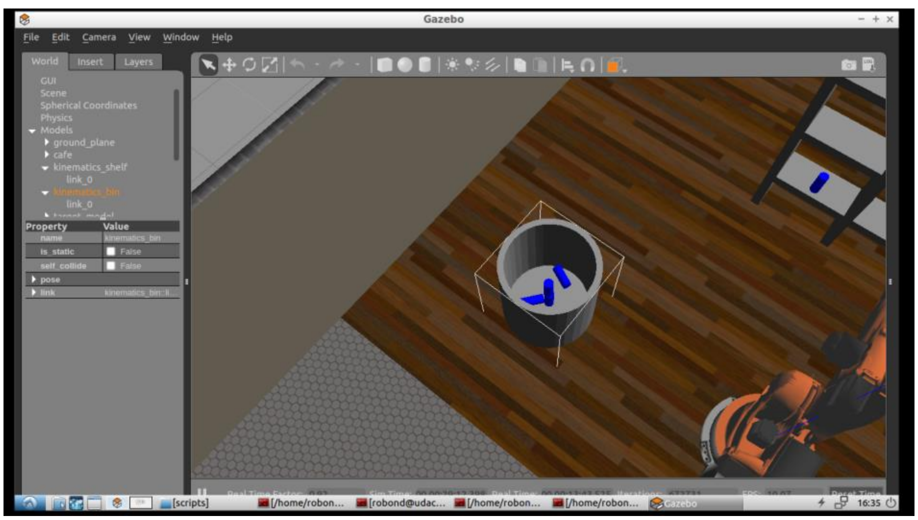

Gazebo
kuka_arm.rviz* - RViz
Next
Continue
Break
Stop

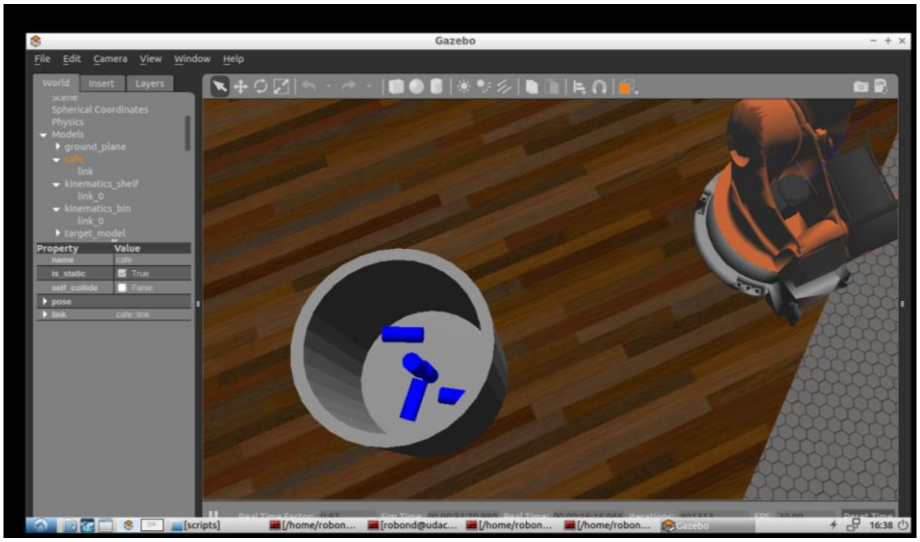

kuka_arm.rviz* - RViz
Next
Continue
Break
Stop

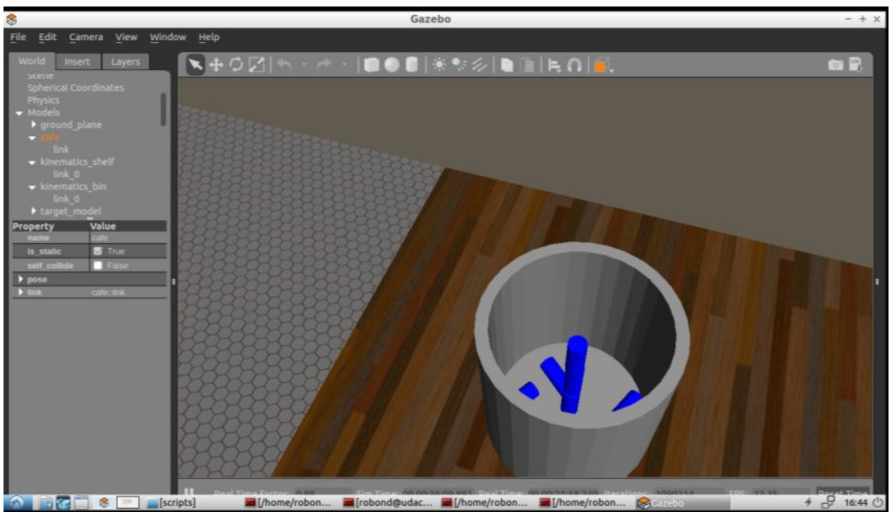

kuka_arm.rviz* - RViz
Next
Continue
Break
Stop
Moving to the drop-off

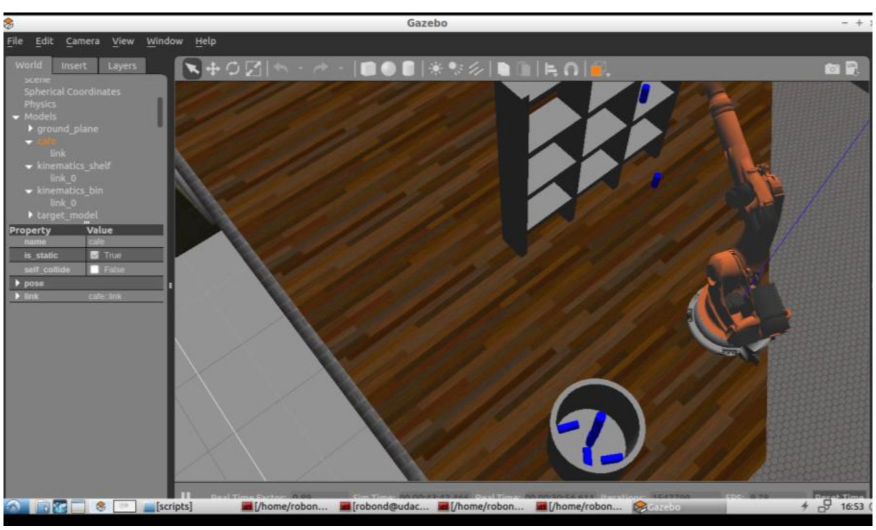



## FINAL RESULT




Gazebo
File Edit Camera View Window Help
World Insert Layers
Spherical Coordinates
Physics
Models
ground_plane
kinematics_shelf
kinematics_bin
target_model
Property Value
[scripts]


Gazebo
File Edit Camera View Window Help
World Insert Layers
Spherical Coordinates
Physics
Models
ground_plane
kinematics_shelf
kinematics_bin
target_model
Property Value
[scripts]


Gazebo
File Edit Camera View Window Help
World Insert Layers
Spherical Coordinates
Physics
Models
ground_plane
kinematics_shelf
kinematics_bin
target_model
Property Value
[scripts]