Kinematic Analysis and Gazebo Simulation

KUKA 210 Robotic Arm

Date: Aug 6th, 2018

Menghong Feng

1 Overview

This report demonstrates the complete process of kinematic analysis procedure for KUKA 210 robotic arm. KUKA 210 is a 6 degree of freedom (DOF) spherical (RRP) robotic arm which has been widely used in industry. The analysis method presented in this report can also be applied in other types of robotic arm, such as cartesian (PPP), cylindrical (RPP), SCARA (RRP), and so on.

Kinematic analysis consists of two process: forward kinematic and inverse kinematic. Forward kinematic is defined as the process to compute end-effector position from specificed joint parameters value. Inverse kinematic, on the contrary, refers to process to recover joint condition, such as movement, angle, and position, by knowing the spatial information of end-effector. In this project, the cylinder object may appear in random location and robotic arm is required to pick it and place it to trash bin. To solve this problem we have to go through the process of forward and inverse kinematic analysis. The steps are summarized below:

1. Find joints and label out their axis orientation
2. Refer to urdf.xacro file to derive DH parameters
3. Use DH parameters to create individual transformation matrices
4. Generate a generalized homogeneous transform between base link and end-effector (gripper)
5. Perform inverse position kinematics and inverse orientation kinematics

2 Kinematic Analysis

2.1 Forward Kinematic

Figure 1 shows a real KUKA 210 robotic arm and its six joints labeled in figure. Figure 2 is a representation to demonstrate joints' axis and orientation. Wrist center is located in joint 5. An end-effector (gripper) is connected to joint 6 and labeled as joint 7. By referring to the urdf.xacro file from ROS (robotic operation system), the spatiail information can be found and collected in table 1. Figure 3 is used to visualize each joint's spatial relation.

Table 1: urdf.xacro joint parameters

$Joint_i$	x_i	y_i	z_i
0	0	0	0
1	0	0	0.33
2	0.35	0	0.42
3	0	0	1.25
4	0.96	0	-0.054
5	0.54	0	0
6	0.193	0	0
7	0.11	0	0

The joint parameters value obtained from urdf.xacro file can be used to compute DH table, which is shown in below. The twist angle α_i is measured between the z_{i-1} and z_i axis of adjacent joints. Link length a_i is measured between the distance of z_{i-1} and z_i axis of adjacent joints. Link offset d_i is the distance between x_{i-1} and x_i axis of adjacent joints. Revolute joint angle q_i or θ_i is the angle of joint, which is the orientation variable appeared later in the homogenous transformation.

Table 2: DH Table Parameters

α_0: 0	a_0: 0	d_1: 0.75	q_1: q_1
α_1: -π/2	a_1: 0.35	d_2: 0	q_2: q_2-π/2
α_2: 0	a_2: 1.25	d_3: 0	q_3: q_3
α_3: -π/2	a_3: -0.054	d_4: 1.5	q_4: q_4
α_4: π/2	a_4: 0	d_5: 0	q_5: q_5
α_5: -π/2	a_5: 0	d_6: 0	q_6: q_6
α_6: 0	a_6: 0	d_7: 0.303	q_7: 0

With the DH table parameters been collected, now it's time to develop individual transformation matrics and homogenous transform bewteen base link with end-effector (gripper). The transformation matrix is defined as below:

$$ {}^{i-1}_{i}T = R(x_{i-1}, \alpha_{i-1})T(x_{i-1}, a_{i-1})R(z_i, \theta_i)T(z_i, d_i) $$

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i - 1 \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_i - d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_i - d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ T_{0EE} = T_{01} \times T_{12} \times T_{23} \times T_{34} \times T_{45} \times T_{56} \times T_{6EE} $$

The first equation is individual transformation matric from joint i-1 to joint i, which consists of two translations and two rotations along z axis and x aixs. The matric form is shown as second equation. The thrid equation is the homogenous transformation between base link to end-effector(gripper).

One thing we need to be careful about is that the orientation of end-effector (gripper) defined in urdf file is different from the one in DH convention. As shown in figure 4, the green coordinate is urdf reference and orange coordinate is DH reference frame. To compensate the difference, we need to rotate the urdf reference frame to 180° along z axis and -90° along y axis. Three rotation matrices along x, y, and z axis are defined below along with the orientation correction equation:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad (1) \qquad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2) \qquad R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$R_{corr} = R_z(180°) \times R_y(-90°)$$

$$T_{0EE} = T_{0EE} \times R_{corr}$$

2.2 Inverse Kinematic

Inverse kinematic is the process of recovering joint condition by giving the position and orientation of end-effector (gripper). In this project, the position of end-effector can be retrieved by using ROS commend. The joints 4, 5, and 6 form together as spheirical wrist, with wrist centor at joint 5. Therefore, the first step is to recover the wrist centor (WC) position by applying inverse position kinematic analysis. The equation to find WC position is:

$$\begin{bmatrix} WC_x \\ WC_y \\ WC_z \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - {}^0_6R \begin{bmatrix} 0 \\ 0 \\ d_G \end{bmatrix}$$

Or the matrix form can be expressed as individual equation:

$$WC_x = p_x - (d_6 + l) \cdot n_x$$

$$WC_y = p_y - (d_6 + l) \cdot n_y$$

$$WC_z = p_z - (d_6 + l) \cdot n_z$$

Where $WC_{x,y,z}$ represent wrist centor (joint 5) position, $p_{x,y,z}$ are the position of end-effector (gripper), which are given in this project. $d_6 + l$ is the distance between wrist center to end-effector. $n_{x,y,z}$ form the vector along the z-axis of the gripper link, which can be obtained from end-effector pose R_{rpy} with respect to the base link by using $euler_from_quaternions()$ method.

The end-effector pose with respect to base link can be calculated as below. Correction matrice must be applied to convert reference frame from urdf to DH parameter frame. At last, the $n_{x,y,z}$ values can be extracted from R_{rpy}.

$$R_{rpy} = R_z(yaw) \times R_y(pitch) \times R_x(roll) \times R_{corr}$$

With the wrist center position been found, the inverse position kinematic has been completed. The next step is to perform inverse orientation kinematic to find the rotational angle of each joints. As shown in figure 5, the top view shows that rotation angle of joint 1 is simply the arctan of WC_y and WC_x. In figure 6 and 7, the side view shows that rotation angle of joint 2 and joint 3 can be calculated by using the equation below. Please be aware that the account for WC_y is due to the dynamic rotation movement robotic arm. WC_y equals zero when roboitc arm overlaps with x axis.

$$\theta_1 = \tan^{-1}\frac{WC_y}{WC_x}$$

$$\theta_2 = 90^\circ - \alpha - \tan^{-1}\frac{WC_z - d_1}{\sqrt{WC_x^2 + WC_y^2} - a_1}$$

$$\theta_3 = 90^\circ - \beta - \gamma$$

Where the angle α, β, and γ can be calculated as:

$$\alpha = \cos^{-1}\frac{side_1^2 + side_3^2 - side_2^2}{2 \times side_1 \times side_3}$$

$$\beta = \cos^{-1}\frac{side_2^2 + side_3^2 - side_1^2}{2 \times side_2 \times side_3}$$

$$\gamma = \tan^{-1}\frac{a_3}{0.96 + 0.54}$$

$$side_1 = \sqrt{(\sqrt{WC_x^2 + WC_y^2} - a_1)^2 + (WC_z - d_1)^2}$$

$$side_2(\theta_2) = d_4 = \sqrt{\sqrt{0.96^2 + 0.54^2} + 0.054^2}$$

$$side_2(\theta_3) = 0.96 + 0.54$$

$$side_3 = a2$$

The rotation angle of joint 4, 5, and 6 can be calculated by using Euler angle from quaternion method. The first step is to build the rotational matrix from joint 4 to 6, and the rotation angle of each joint can be extracted from it. The calculation steps are shown below:

$$R_{36} = inv(R_{03}) \times R_{rpy}$$

$$R_{36} = R_Z(\theta_4)R_Y(\theta_5)R_Z(\theta_6) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$R_{36} = \begin{bmatrix} \cos(\theta_4)\cos(\theta_5) & \cos(\theta_4)\sin(\theta_5)\sin(\theta_6) - \sin(\theta_4)\cos(\theta_6) & \cos(\theta_4)\sin(\theta_5)\cos(\theta_6) - \sin(\theta_4)\sin(\theta_5) \\ \sin(\theta_4)\cos(\theta_5) & \sin(\theta_4)\sin(\theta_5)\sin(\theta_6) + \cos(\theta_4)\cos(\theta_6) & \sin(\theta_4)\sin(\theta_5)\cos(\theta_6) - \cos(\theta_4)\sin(\theta_5) \\ -\sin(\theta_5) & \cos(\theta_5)\sin(\theta_6) & \cos(\theta_5)\cos(\theta_6) \end{bmatrix}$$

The rotation angle θ_4, θ_5, and θ_6 can be calculated as follow:

$$\theta_4 = \tan^{-1}\frac{r_{21}}{r_{11}}$$

$$\theta_5 = \tan^{-1}\frac{-r_{31}}{\sqrt{r_{11}^2 + r_{21}^2}}$$

$$\theta_6 = \tan^{-1}\frac{r_{32}}{r_{33}}$$

3 Project Implementation

The pick & place task has been performed 10 times. 8 out of 10 tasks succeed in picking object and placing it in trash bin. 2 out of 10 tasks fail for the same reason. I realize that inverse kinematic calculation process will takes forever if I use simplify function to round off float point results. Also the by assigning numerical values directly in equation will considerably reduce the calculation time. Another issue found in the task is that, though the task succeeeds at the end, the end-effector performs several unnecessary rotation during the task. It is because I forget to correct reference frame for wrist center. After I fix this issue the robotic behaves normally. The screenshots for successfully completing task are attached in the Appendix section.

In Appendix section, figure 8 to figure 20 shows each step of task in KViz monitor. Figure 21 to figure 24 are the screenshots of dynamic simulation in Gazebo. Figure 25 to figure 32 shows the complete commend lists in terminal.

4 Appendix

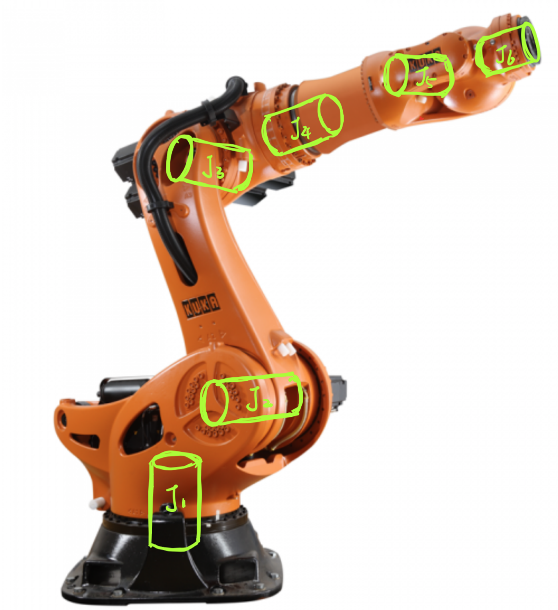

Figure 1: KUKA 210 Robotic Arm

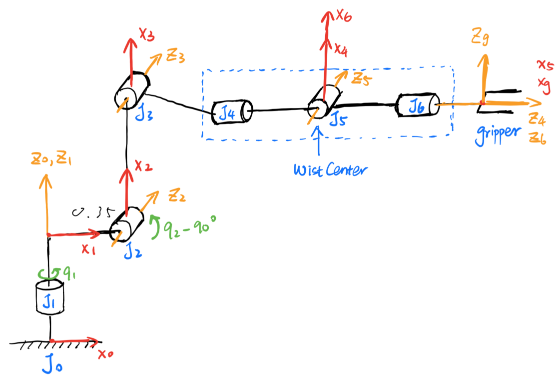

Figure 2: Axis Label and Orientation

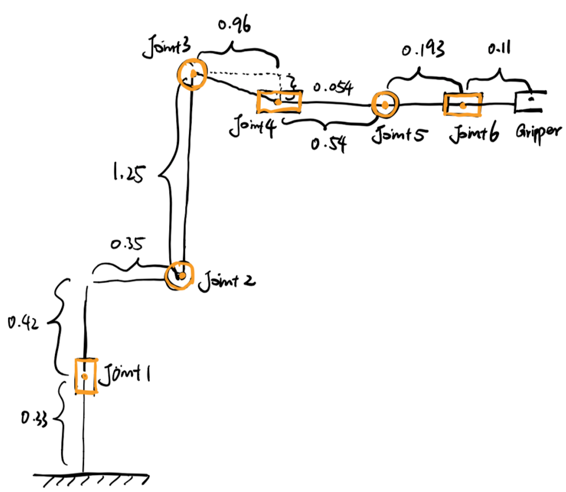

Figure 3: Joint Spatial Relation

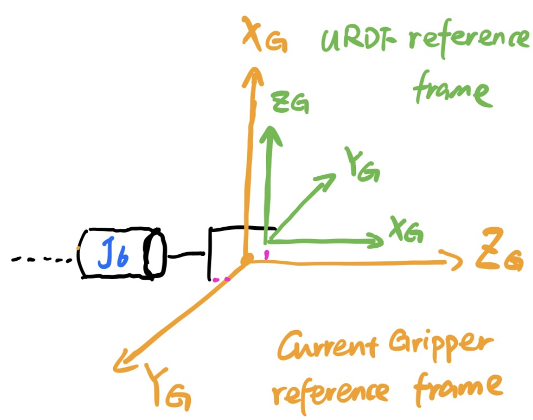

Figure 4: Reference Frame

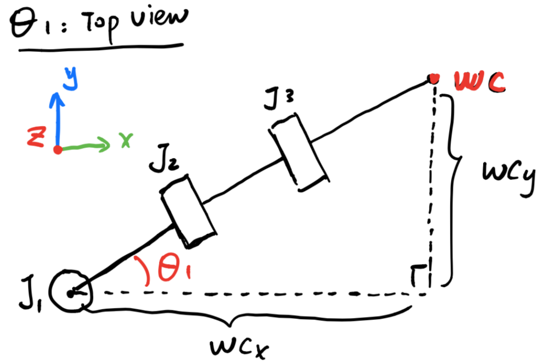

Figure 5: θ_1

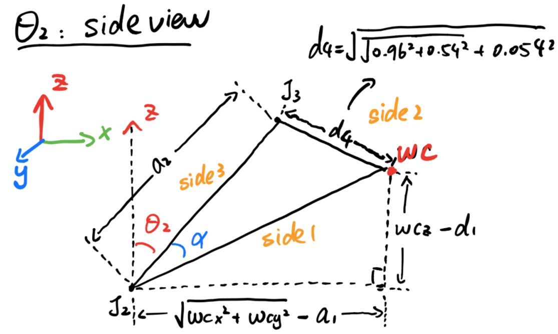

Figure 6: θ_2

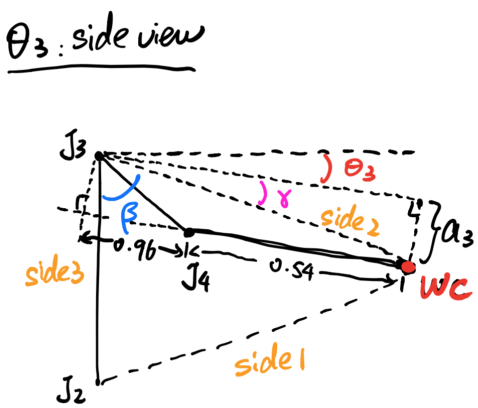

Figure 7: θ_3

Figure 8: KViz: Start

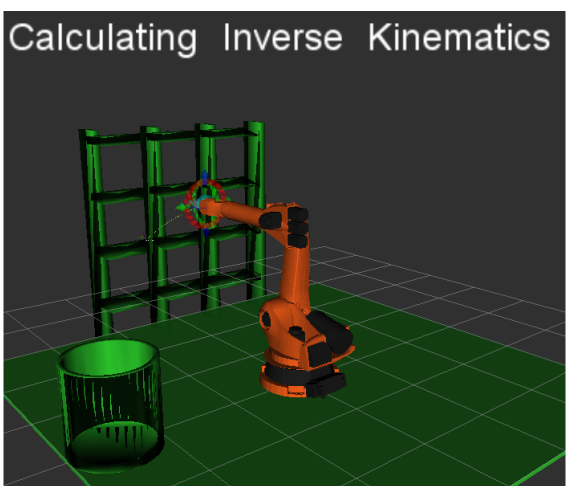

Figure 9: KViz: IK calculation

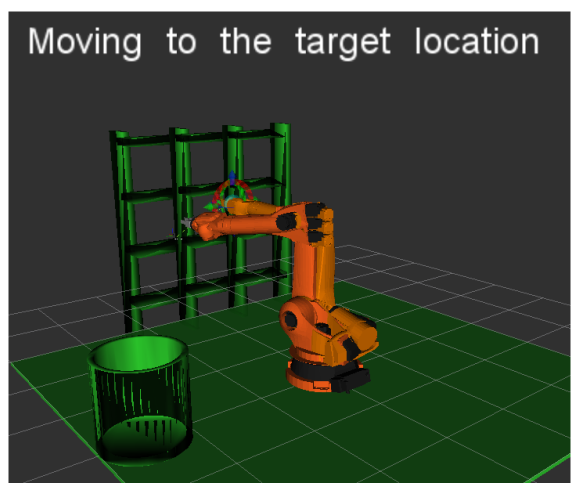

Figure 10: KViz: Moving to Target Location

Figure 11: KViz: Reach to Target Location

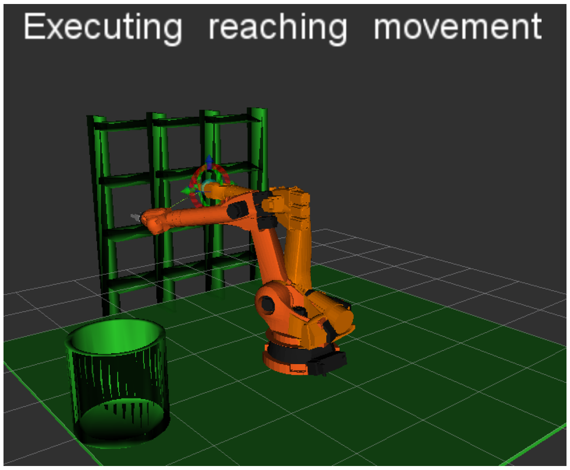

Figure 12: KViz: Executing Reaching Movement

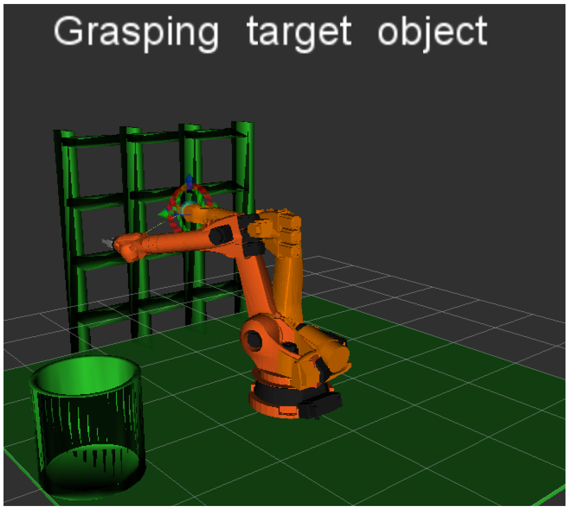

Figure 13: KViz: Grasping Target Object

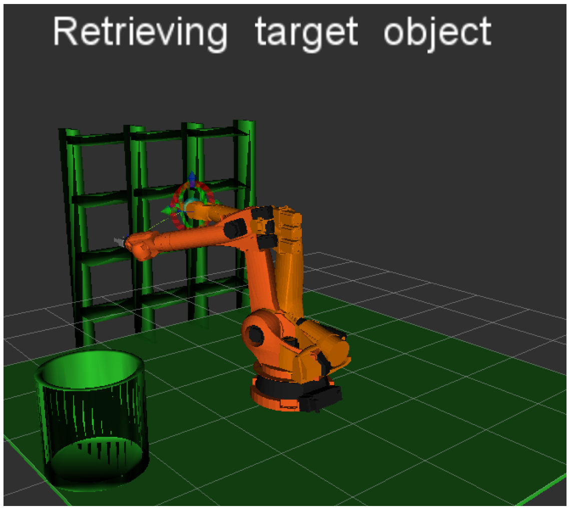

Figure 14: KViz: Retrieving Target Object

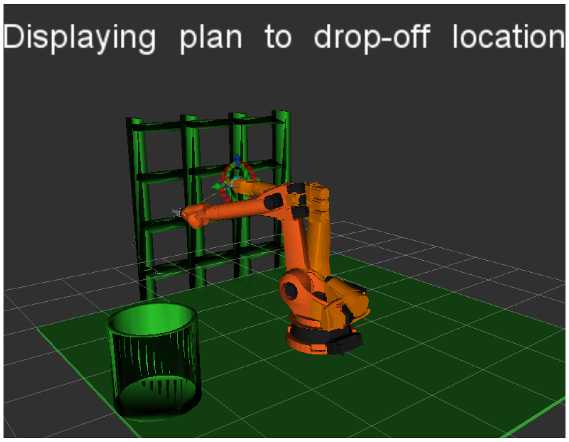

Figure 15: KViz: Display Plan to Drop-off Location

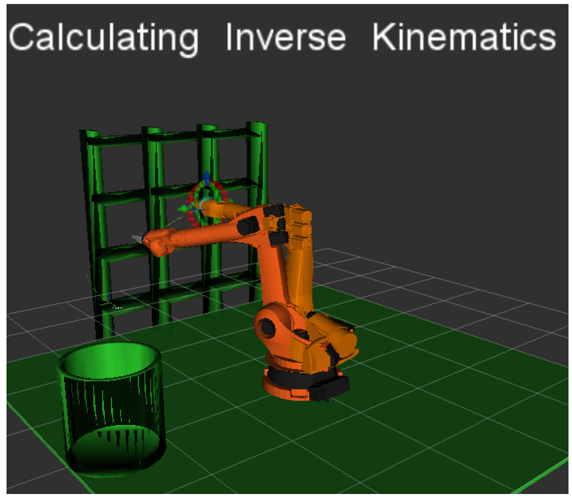

Figure 16: KViz: IK Calculation

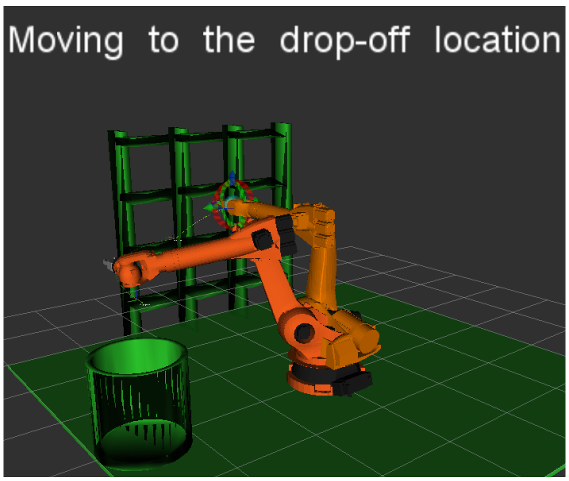

Figure 17: KViz: Moving to Drop-off Location

Figure 18: KViz: Reached to Drop-off Location

Figure 19: KViz: Releasing Target Object

Figure 20: KViz: End

Figure 21: Gazebo: Start

Figure 22: Gazebo: Reach and Grasp

Figure 23: Gazebo: Move to Drop-off Location

Figure 24: Gazebo: Release Target Object to Drop-off Location

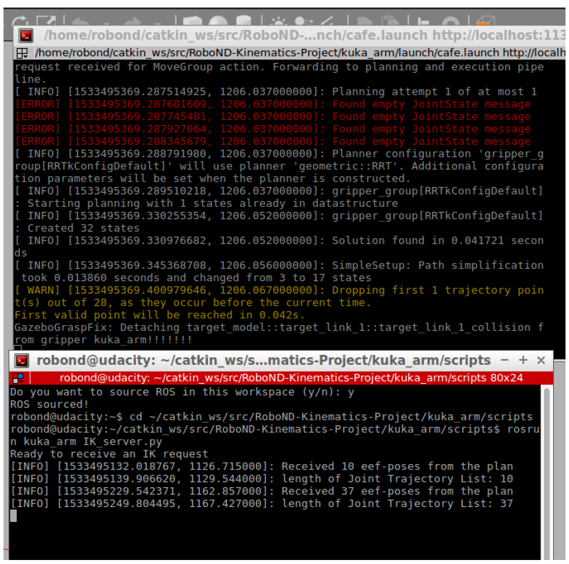

Figure 25: Run 1

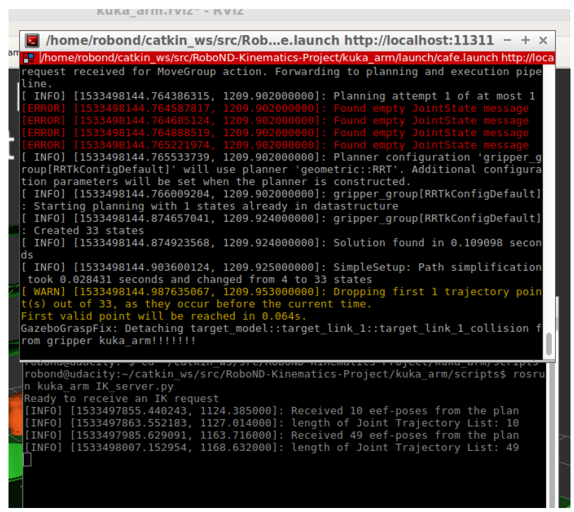

Figure 26: Run 2

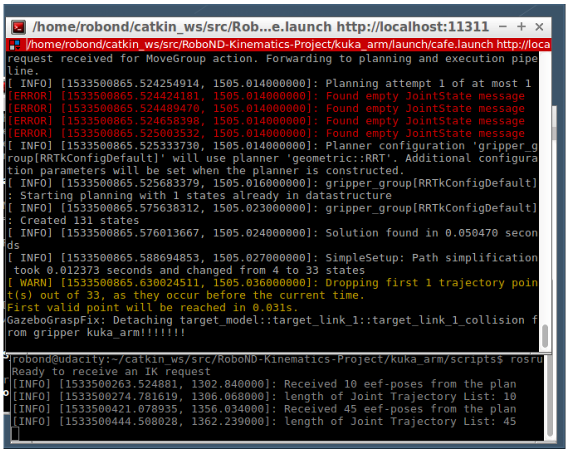

Figure 27: Run 3

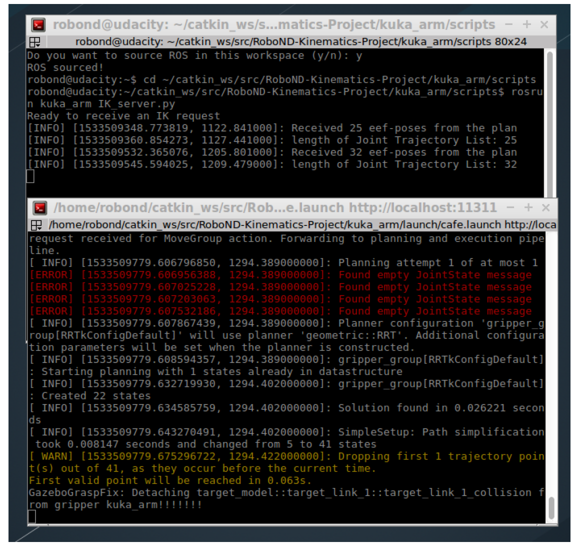

Figure 28: Run 4

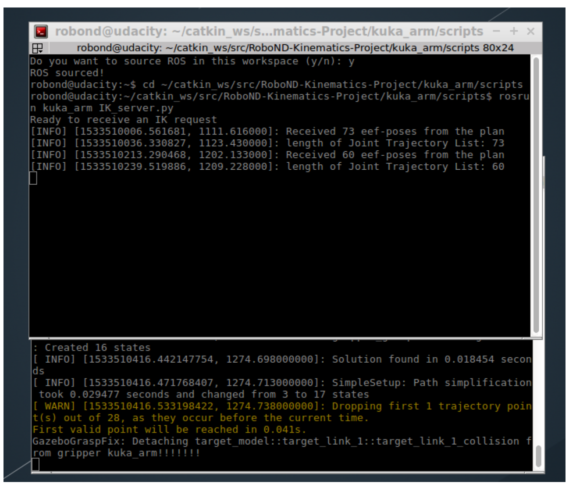

Figure 29: Run 5

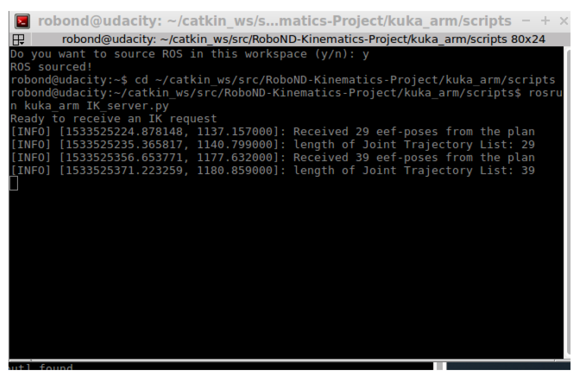

Figure 30: Run 6

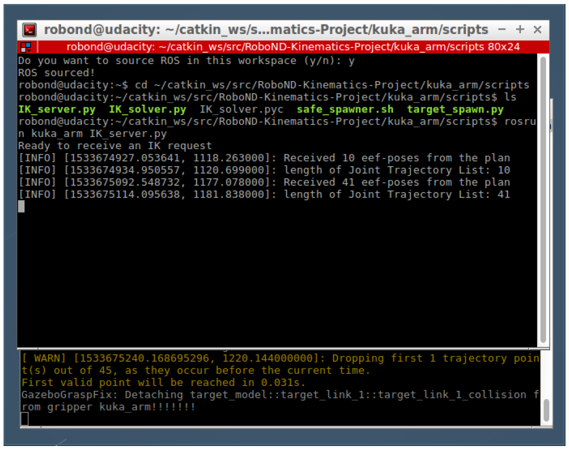

Figure 31: Run 7

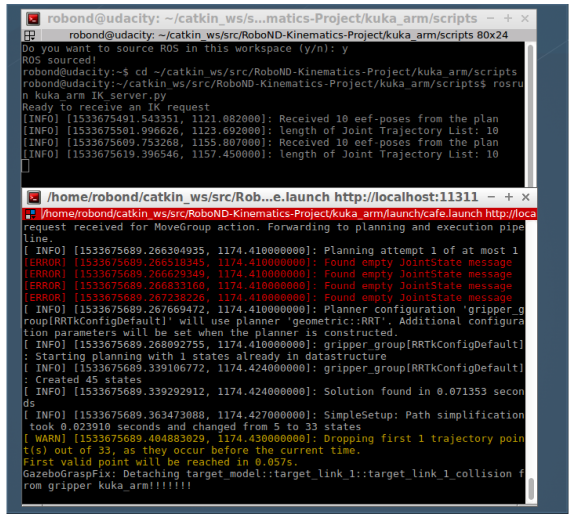

Figure 32: Run 8