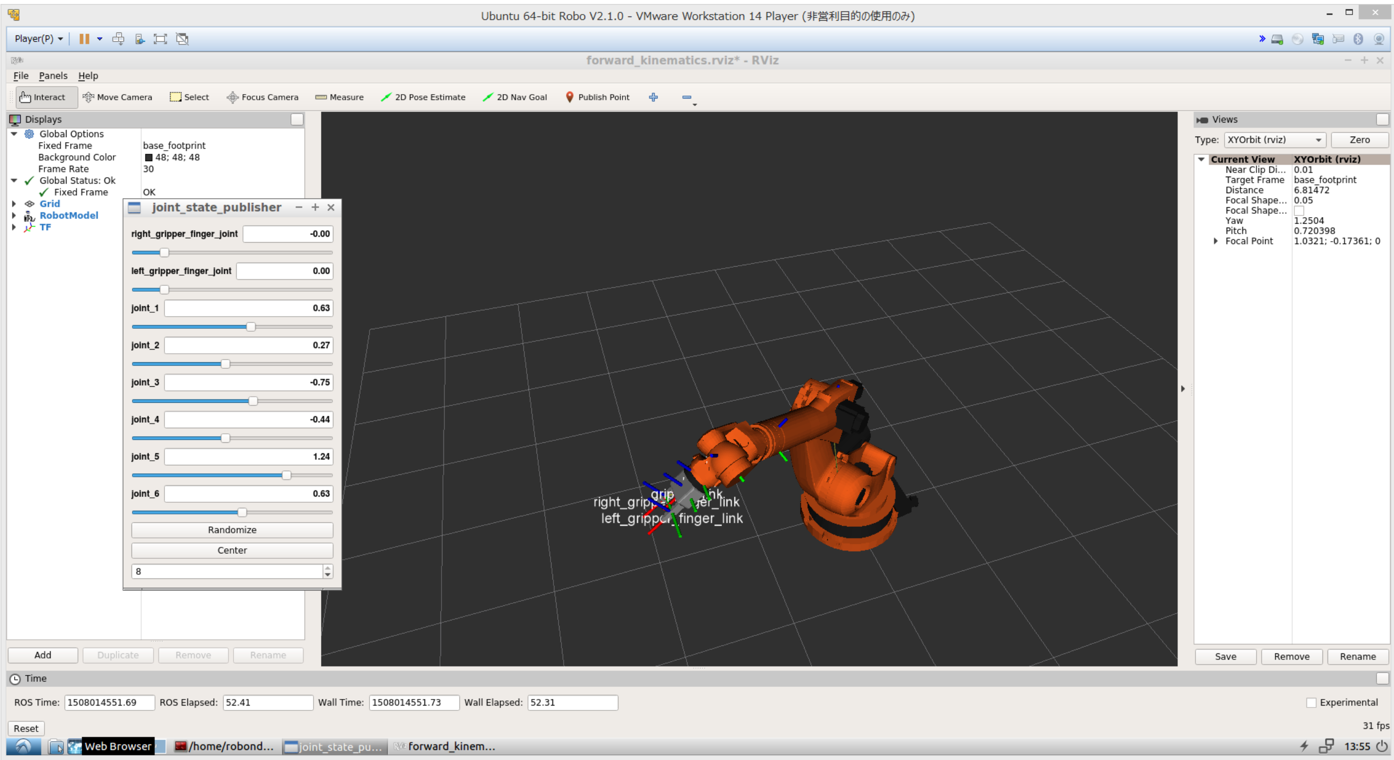
Project: Kinematics Pick & Place

Kinematic Analysis

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.



DH parameter table

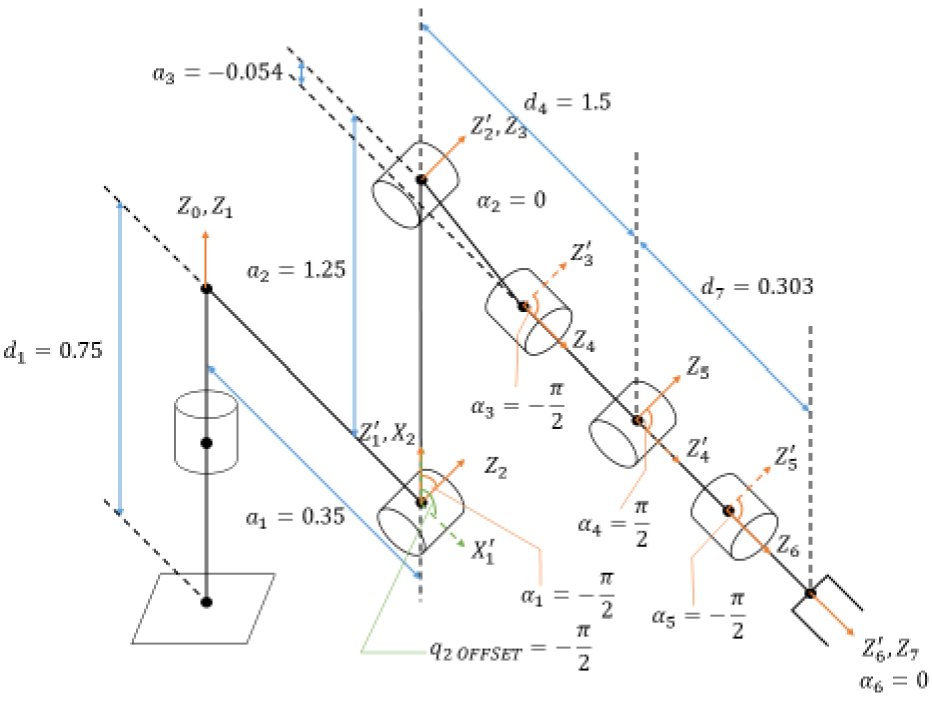
Links	α_{i-1}	a_{i-1}	d_i	θ_i
0 to 1	0	0	0.75	q_1
1 to 2	$-\frac{\pi}{2}$	0.35	0	$-\frac{\pi}{2}+q_2$
2 to 3	0	1.25	0	q_3
3 to 4	$-\frac{\pi}{2}$	-0.054	1.5	q_4
4 to 5	$\frac{\pi}{2}$	0	0	q_5
5 to 6	$-\frac{\pi}{2}$	0	0	q_6
6 to EE	0	0	0.303	0

Explanation about DH parameter table

α_{i-1} is the angle from Z_{i-1} to Z_i as visualized in the picture below.

a_i is the distance between Z_{i-1} and Z_i along with X_{i-1} as visualized in picture 1.
a_1 is 0.35, which is the distance from Joint 1 to Joint 2 along with X_1.
a_2 is 1.25, which is the distance from Joint 2 to Joint 3 along with X_2.
a_3 is -0.054 which is the distance from Joint 3 to Joint 4 along with X_3.

d_i is the distance between Z_{i-1} and Z_i along with Z_i as visualized in picture 1.
d_1 is 0.42 + 0.33 = 0.75. 0.33 is the distance from the ground to Joint 1. 0.44 is the distance from Joint 1 to Joint 2. Since neither d2 nor a1 can describe the distance from Joint 1 to Joint 2 along with Z_1, d_1 must include 0.42.
d_4 is 0.96 + 0.54 = 1.5. 0.96 is the distance from Joint 3 to Joint 4. 0.54 is the distance from Joing 4 to Joint 5. Since neither d_5 nor a_4 can describe the distance from Joint 4 to Joint 5 along with Z_4, d_4 must include 0.54.
d_7 is 0.193 + 0.11 = 0.303. 0.193 is the distance from Joint 5 to Joint 6. 0.11 is he distance from Joint 6 to the end effector. Since Z_6 and Z_7 are always in the same direction, we can safely add d_6 to d_7 to simplify the end effector position calculation.



Picture 1: Kuka KR210 joints

2. Using the DH parameter table you derived earlier, create individual transformation matrices about each joint. In addition, also generate a generalized homogeneous transform between base_link and gripper_link using only end-effector(gripper) pose.

The individual transformation matrices are calculated as T0_1, T1_2, T2_3, T3_4, T4_5, T5_6 and T6_G in IK_server.py.

The generalized homogeneous transform between base_link and gripper_link is calculated as T_Total in IK_server.py. The calculation is just multiplying all individual transformation matrices in the order of T0_1 to T6_G.

3. Decouple Inverse Kinematics problem into Inverse Position Kinematics and inverse Orientation Kinematics; doing so derive the equations to calculate all individual joint angles.

Calculation of the wrist center position

The end effector's angles (r, p and y in the end effector's local coordinate) and position relative to base_link (p_x, p_y, p_z) are given.

Calculate the rotation matrix R_{rpy} in the world base_link's coordinate as below.

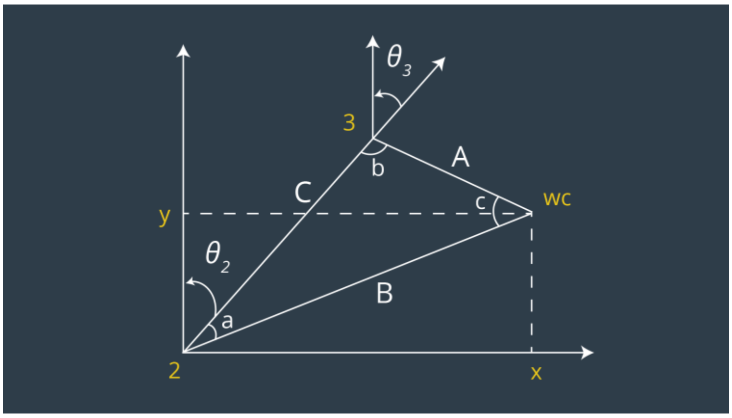
$$R_{rpy} = \begin{bmatrix} \cos(y) & -\sin(y) & 0 \\ \sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(p) & 0 & \sin(p) \\ 0 & 1 & 0 \\ -\sin(p) & 0 & \cos(p) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(r) & -\sin(r) \\ 0 & \sin(r) & \cos(r) \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Then R_{rpy}'s 3rd column vector N is the end effector's Z axis unit vector. The position of wrist center WC is

$$\mathrm{WC} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - d_7 N$$

Calculation of θ_1, θ_2 and θ_3

Use picture 2 as reference.



Picture 2: The side view of Joint2, 3 and Wrist Center

From DH parameter table, calculate the length of A, B and C as follows.

$A = d_4 = 1.5$

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

$C = a_2 = 1.25$

With cosine law, calculate angle $\angle$a, $\angle$b and $\angle$c.

$$\angle \mathrm{a} = \frac{\cos^{-1}(B^2 + C^2 - A^2)}{2BC}$$

$$\angle \mathrm{b} = \frac{\cos^{-1}(A^2 + C^2 - B^2)}{2AC}$$

$$\angle \mathrm{c} = \frac{\cos^{-1}(A^2 + B^2 - C^2)}{2AB}$$

Then, calculate θ_1 from WC's position in X-Y plane.

$$\theta_1 = \tan^{-1}\left(\frac{WC_y}{WC_x}\right)$$

Then, from picture2, calculate θ_2 as below.

$$\theta_2 = \frac{\pi}{2} - \angle \mathrm{a} - \tan^{-1}\left(\frac{WC_z - d_1}{\sqrt{WC_x^2 + WC_y^2} - a_1}\right)$$

Then, from picture2, calculate θ_3 as below. We consider the angle between Joint 3 and Joint 4 due to a_3. ($d_{J3\ to\ J4}$ in the equation below is the distance between Joint 3 and Joint 4 along with Z_4)

$$\theta_3 = \frac{\pi}{2} - \left(\angle \mathrm{b} + \tan^{-1}\left(\frac{a_3}{d_{J3\ to\ J4}}\right)\right)$$

Then, to calculate θ_4, θ_5 and θ_6, calculate the transform matrix from joint 4 to Joint 6.

$$T = T_{34}T_{45}T_{56}$$

$$= \begin{bmatrix} \cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\cos\theta_5\cos\theta_6 - \sin\theta_4\sin\theta_6 & -\cos\theta_4\sin\theta_5 & -0.054 \\ \sin\theta_5\cos\theta_6 & -\sin\theta_5\sin\theta_6 & \cos\theta_5 & 1.5 \\ -\cos\theta_4\sin\theta_6 - \sin\theta_4\cos\theta_5\cos\theta_6 & -\cos\theta_4\cos\theta_6 - \sin\theta_4\cos\theta_5\sin\theta_6 & \sin\theta_4\sin\theta_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, calculate θ_4 as below.

$$\theta_4 = \tan^{-1}\frac{T_{3,3}}{-T_{1,2}}$$

$$= \tan^{-1}\frac{\sin\theta_4\sin\theta_5}{\cos\theta_4\sin\theta_5} = \tan^{-1}\tan\theta_4$$

Then, calculate θ_5 as below.

$$\theta_5 = \tan^{-1}\frac{\sqrt{T_{1,3}^2 + T_{3,3}^2}}{T_{2,3}}$$

$$= \tan^{-1}\frac{\sqrt{(-\cos\theta_4\sin\theta_5)^2 + (\sin\theta_4\sin\theta_5)^2}}{\cos\theta_5}$$

$$= \tan\frac{\sin\theta_5\sqrt{\sin{\theta_4}^2 + \cos{\theta_4}^2}}{\cos\theta_5} = \tan^{-1}\tan\theta_5$$

Then, calculate θ_5 as below.

$$\theta_6 = \tan^{-1}\frac{-T_{2,2}}{T_{2,1}}$$

$$\tan^{-1}\frac{\sin\theta_5\sin\theta_6}{\sin\theta_5\cos\theta_6} = \tan^{-1}\tan\theta_6$$

Project Implementation

1. Fill in the `IK_server.py` file with properly commented python code for calculating Inverse Kinematics based on previously performed Kinematic Analysis. Your code must guide the robot to successfully complete 8/10 pick and place cycles. Briefly discuss the code you implemented and your results.

Regarding the implementation, please refer IK_server.py.

The implementation is consistent with this write up. The success rate is approximately 90%. When it fails, it always drops the blue target while the arm moves from the retrieved state to the drop ready state. The reason is unknown since the gripping is not controlled by my code.

If I would improve the implementation, I would do following.

- Improve grip code.
- Improve IK code to minimize the total rotation of each joint to save energy.