Project Report

Robotic Arm: Pick & Place

Kalyani Chawak

Table of Contents

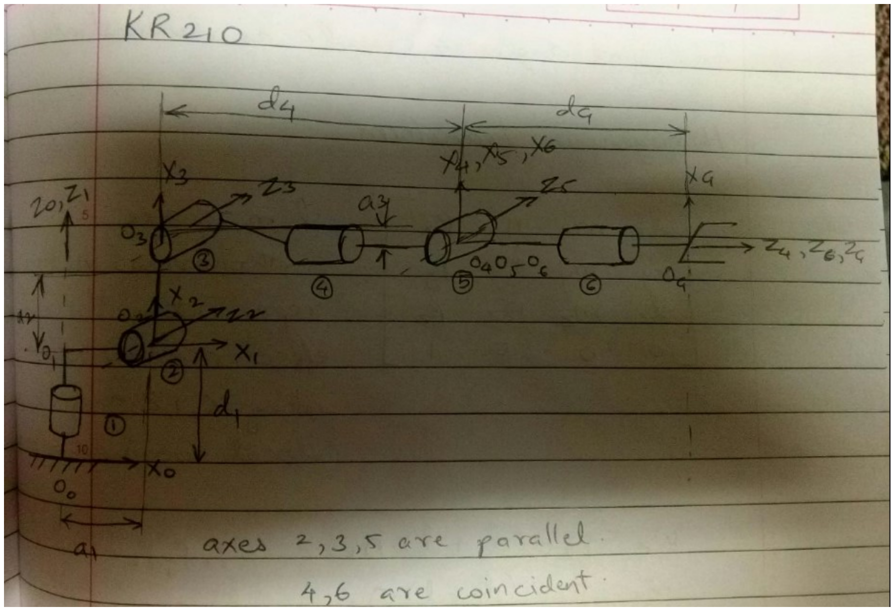
Chapter 1: **Introduction**

This project is based on the Kuka KR210 Robotic Arm. It involves simulation of a 6 DOF Robotic Arm in ROS environment using tools like Gazebo and RViz. The Robotic Arm is designed to pick up an object from any random shelf and place it in a Dropbox.

Chapter 2: **Kinematic Analysis**

2.1 DH Parameters

2.1.1 Annotated figure of Robot Arm



2.1.2 DH Parameter table

The below table is derived from the above figure.

α = Twist angle (angle between Z_{i-1} and Z_i measured about X_{i-1} right hand)

a = link length (distance from Z_{i-1} to Z_i along X_{i-1}. X_i perpendicular to both Z_{i-1} and Z_i)

d = link offset (signed distance from X_{i-1} to X_i measured along Z_i)

θ = joint angle (angle between X_{i-1} to X_i about Z_i right hand)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	q_1
2	$-\pi/2$	a_1	0	q_2
3	0	a_2	0	q_3
4	$-\pi/2$	a_3	d_4	q_4
5	$\pi/2$	0	0	q_5
6	$-\pi/2$	0	0	q_6
7	0	0	d_7	0

2.1.3 Modified DH Parameter table

File kr210.urdf.xacro is referred to obtain the following table.

Procedure of deduction of DH parameters:

- I obtained the origins of all the joints from the file.
- Then depending on the annotated figure, I added or subtracted the values to populate the table.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0.75	q_1
2	$-\pi/2$	0.35	0	q_2
3	0	1.25	0	q_3
4	$-\pi/2$	-0.054	1.5	q_4
5	$\pi/2$	0	0	q_5
6	$-\pi/2$	0	0	q_6
7	0	0	0.303	0

2.2 Transformation Matrices

2.2.1 Individual transformation matrices

The generalized transformation matrix is given as follows:

$$
{}^{i-1}_{\;\;i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The individual transformation matrices for each joint can be obtained through substitution in the generalized transformation matrix. They are as follows:

T0_1 = [[cos(q1), -sin(q1), 0, 0],

[sin(q1), 0, 0, 0],

[0, 0, 1, 0.75],

[0, 0, 0, 1]]

T1_2 = [[sin(q2), cos(q2), 0, 0.35],

[0, -sin(q2), 1, 0],

[cos(q2), -sin(q2), 0, 0],

[0, 0, 0, 1]]

T2_3 = [[cos(q3), -sin(q3), 0, 1.25],

[sin(q3), 0, 0, 0],

[0, 0, 1, 0],

[0, 0, 0, 1]]

T3_4 = [[cos(q4), -sin(q4), 0, -0.054],

[0, -cos(q4), 1, 1.5],

[-sin(q4), -cos(q4), 0, 0],

[0, 0, 0, 1]]

T4_5 = [[cos(q5), -sin(q5), 0, 0],

[0, cos(q5), -1, 0],

[sin(q5), cos(q5), 0, 0],

[0, 0, 0, 1]]

T5_6 = [[cos(q6), -sin(q6), 0, 0],

[0, -cos(q6), 1, 0],

[-sin(q6), -cos(q6), 0, 0],

[0, 0, 0, 1]]

T6_EE = [[1, 0, 0, 0],

[0, 0, 0, 0],

[0, 0, 1, 0.303],

[0, 0, 0, 1]])

2.2.2 Homogenous transformation matrix

The homogenous transformation matrix can be obtained by multiplying all the matrices together.

T0_EE = T0_1 * T1_2 * T2_3 * T3_4 * T4_5 * T5_6 * T6_EE

$$\begin{bmatrix} l_x & m_x & n_x & p_x \\ l_y & m_y & n_y & p_y \\ l_z & m_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3 Geometric Inverse Kinematics problem

2.3.1 Breakdown into Position and Orientation problems

The End Effector is divided into Position and Orientation components. Position components(px, py and pz) are extracted with the help of poses and position.

Orientation components (roll, pitch and yaw) are extracted with the help of euler_from_quaternion.

The Rotation around x, y and z axis corresponds to Roll, pitch and yaw respectively.

R_x = [[1, 0, 0],

[0, cos(r), -sin(r)],

[0, sin(r), cos(r)]]

R_y = [[cos(p), 0, sin(p)],

[0, 1, 0],

[-sin(p), 0, cos(p)]]

R_z = [[cos(y), -sin(y), 0],

[sin(y), cos(y), 0],

[0, 0, 1]]

R_EE = R_z * R_y * R_x

Here, the error exists mainly in Yaw (by 180°) and Pitch (by -90°). I calculated the error matrix and multiplied it with End Effector rotation matrix.

R_EE = R_EE * R_Error

From the homogenous matrix, I calculated the position of the Wrist Centre (WC).

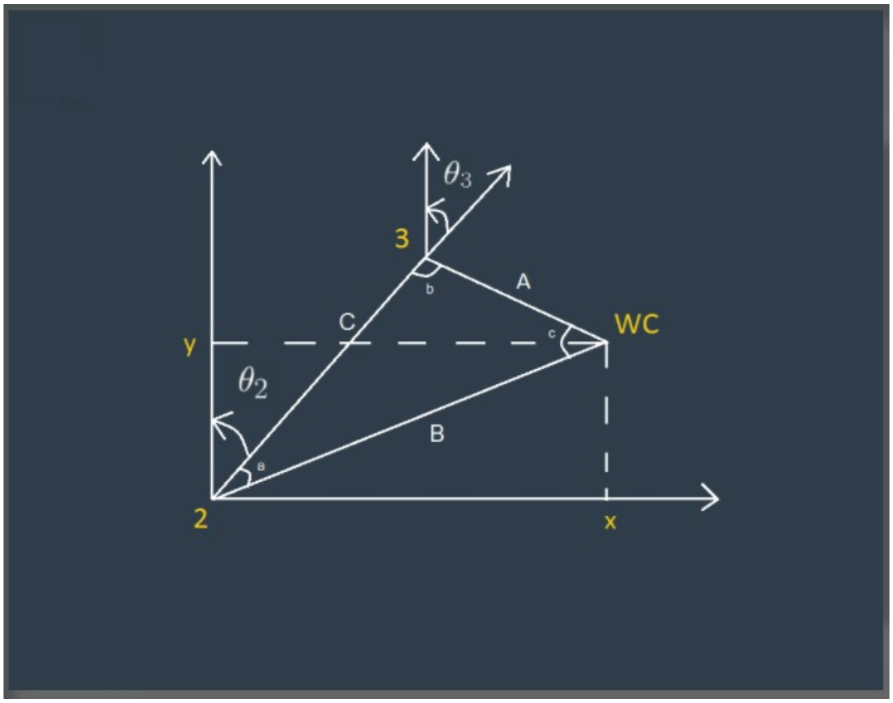
$$\begin{aligned} w_x &= p_x - (d_6 + l) \cdot n_x \\ w_y &= p_y - (d_6 + l) \cdot n_y \\ w_z &= p_z - (d_6 + l) \cdot n_z \end{aligned}$$

2.3.2 Derivation for equations of individual joint angles

The individual joint angles are calculated by Geometric Inverse Kinematics method. Theta1, theta2 and theta3 can be calculated using the Wrist Centre.

theta1 = atan2(y,x)

The figure for calculating theta2 and theta3 is as follows:



From DH parameter table,

Side A = 1.501

$$\text{side } B = \sqrt{\left(\sqrt{(x^2+y^2)} - a_1\right)^2 + (z-d_1)^2}$$

Side C = 1.25

Using the laws of cosine,

$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

γ = angle contained between sides of lengths a and b and opposite to the side of length c.

angle_a = acos((side_B*side_B + side_C*side_C - side_A*side_A)/(2*side_B*side_C))

angle_b = acos((side_A*side_A + side_C*side_C - side_B*side_B)/(2*side_A*side_C))

angle_c = acos((side_A*side_A + side_B*side_B - side_C*side_C)/(2*side_A*side_B))

$$theta_2 = \pi/2 - atan2\left(z - d_1, \sqrt{x^2 + y^2} - a_1\right)$$

$$theta_3 = \pi/2 - (angle\_b + 0.036)$$

Remaining thetas can be calculated using the rotation matrix R3_6

R0_3 = T0_1[0:3,0:3] * T1_2[0:3,0:3] * T2_3[0:3,0:3]

R3_6 = $[R0\_3]^{-1}$ * R_EE

theta4 = atan2(R3_6[2,2], -R3_6[0,2])

$$theta5 = atan2\left(\sqrt{(R3\_6[0,2])^2 + (R_{3\_6}[2,2])^2}, R_{3\_6}[1,2]\right)$$

theta6 = atan2(-R3_6[1,1], R3_6[1,0])

theta4 and theta6 has multiple values but the movement is restricted due to the physical limits.

Chapter 3: **Possible Improvements**

1) **Code optimization** – The code can be optimized by introduction of classes or saving the transformation matrices to a file. This can also reduce the time taken to calculate the joint angles through inverse kinematics

2) **Plotting the error in EE pose -** Plotting the error between the input end effector pose and calculated pose will also serve as an improvement

Bibliography

[1] http://docs.sympy.org/latest/modules/matrices/expressions.html

[2] https://en.wikipedia.org/wiki/Law_of_cosines

[3] http://docs.sympy.org/0.7.2/modules/matrices/matrices.html#linear-algebra

[4] https://www.youtube.com/watch?v=VjsuBT4Npvk

[5] https://en.wikipedia.org/wiki/List_of_trigonometric_identities