Pick and Place Project

By Patrick Wellins

Kinematic Analysis

DH Parameter Table for Kuka Arm

Links	alpha(i – 1)	a(i - 1)	d(i)	theta(i)
0 - 1	0	0	0.75	θ_1
1 - 2	- pi / 2	0.35	0	θ_2 – pi / 2
2 - 3	0	1.25	0	θ_3
3 - 4	- pi / 2	-0.054	1.5	θ_4
4 - 5	pi / 2	0	0	θ_5
5 - 6	- pi / 2	0	0	θ_6
6 - EE	0	0	0.303	0

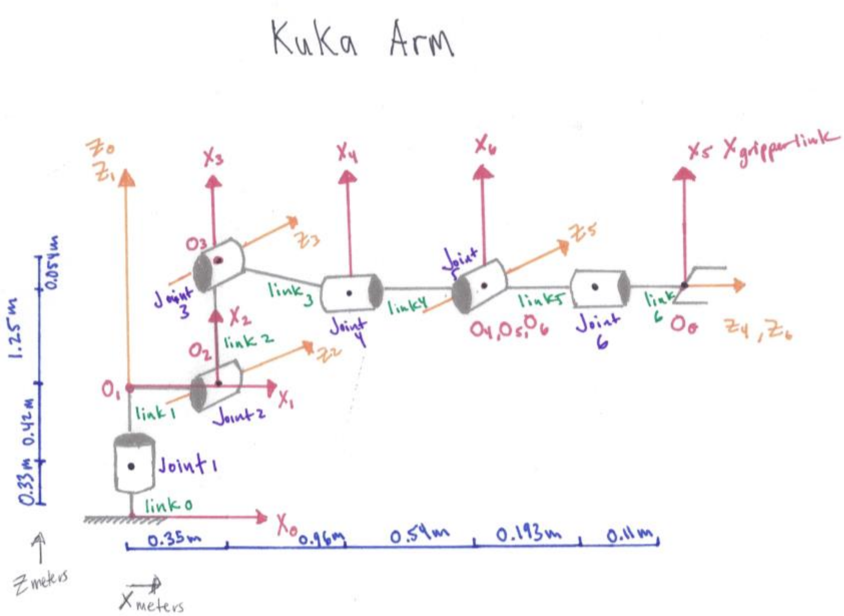
alpha(i – 1) is the twist angle from Z_{i-1} to Z_i measured about X_{i-1} in a right-hand sense.

a(i - 1) is the distance from Z_{i-1} to $Z_i$measured about X_{i-1}.

d(i) is the signed distance from X_{i-1} to X_i along Z_i.

theta(i) measures the angle between X_{i-1} and X_i about Z_i in a right-hand sense.

Figure 1



The DH parameter table above contains information about the Kuka KR210 robotic arm that is sketched above. The DH parameter table will be used to create homogeneous transformation matrices that will allow the position and orientation of the end effector to be expressed in the base link's reference frame. The process of going from joint angles to Cartesian coordinates is in the domain of forward kinematics. The DH parameter table is an integral component of creating homogeneous transformation matrices that allow for forward kinematics problems to be solved.

---

alpha(0) = 0, there is no twist angle from Z_0 to Z_1 about X_0.

alpha(1) = – pi / 2, there is a – pi / 2 radian twist angle from Z_1 to Z_2 about X_1.

alpha(2) = 0, there is no twist angle from Z_2 to Z_3 about X_2.

alpha(3) = – pi / 2, there is a – pi / 2 radian twist angle from Z_3 to Z_4 about X_3.

alpha(4) = pi / 2, there is a pi / 2 radian twist angle from Z_4 to Z_5 about X_4.

alpha(5) = - pi / 2, there is a – pi / 2 radian twist angle from Z_5 to Z_6 about X_5.

alpha(6) = 0, there is no twist angle from Z_6 to the end effector.

---

* **a(i)** values are in meters *

a(0) = 0, there is 0 distance from Z_0 to Z_1 about X_0.

a(1) = 0.35, there is a distance of 0.35 from Z_1 to Z_2 about X_1.

a(2) = 1.25, there is a distance of 1.25 from Z_2 to Z_3 about X_2.

a(3) = - 0.054, there is a distance of - 0.054 from Z_3 to Z_4 about X_3.

a(4) = 0, there is 0 distance from Z_4 to Z_5 about X_4.

a(5) = 0, there is 0 distance from Z_5 to Z_6 about X_5.

a(6) = 0, there is 0 distance from Z_6 to the end effector about X_6.

---

* **d(i)** values are in meters *

d(1) = 0.75, the distance between X_0 and X_1 along Z_1 is 0.75

d(2) = 0, the distance between X_1 and X_2 along Z_2 is 0

d(3) = 0, the distance between X_2 and X_3 along Z_3 is 0

d(4) = 1.5, the distance between X_3 and X_4 along Z_4 is 1.5

d(5) = 0, the distance between X_4 and X_5 along Z_5 is 0

d(6) = 0, the distance between X_5 and X_6 along Z_6 is 0

d(EE) = 0.303, the distance between X_6 and X_{EE} along Z_{EE} is 0.303

---

revolute joints in the diagram are set to 0

theta(1) = θ_1, the angle between X_0 and X_1 along Z_1 is θ_1.

theta(2) = θ_2 - pi / 2, the angle between X_1 and X_2 along Z_2 is θ_2 - pi / 2. The diagram makes the value of theta(2) clear.

theta(3) = θ_3, the angle between X_2 and X_3 along Z_3 is θ_3.

theta(4) = θ_4, the angle between X_3 and X_4 along Z_4 is θ_4.

theta(5) = θ_5, the angle between X_4 and X_5 along Z_5 is θ_5.

theta(6) = θ_6, the angle between X_5 and X_6 along Z_6 is θ_6.

theta(EE) = 0, the angle between X_6 and X_{EE} along Z_{EE} is 0. This is a fixed joint so there is no variation.

Homogeneous Transformation Matrices

Each row of the DH parameter table contains values that are used to build the homogeneous transformation matrices that calculate the Kuka arm's forward kinematics. For example, the first row of parameter values in the DH parameter table is used to populate the entries of matrix ${}_1^0T$. Since there are seven rows in the DH parameter table, there are seven corresponding matrices.

$$
{}_1^0T = \begin{matrix} cos\theta_1 & -sin\theta_1 & 0 & a(0) \\ sin\theta_1 cos\alpha_0 & cos\theta_1 cos\alpha_0 & -sin\alpha_0 & -sin\alpha_0 d(1) \\ sin\theta_1 sin\alpha_0 & cos\theta_1 sin\alpha_0 & cos\alpha_0 & cos\alpha_0 d(1) \\ 0 & 0 & 0 & 1 \end{matrix}
$$

$$ {}^{1}_{2}T = \begin{matrix} \cos(\theta_2 - pi/2) & -\sin(\theta_2 - pi/2) & 0 & a(1) \\ sin(\theta_2 - pi/2)\ cos\alpha_1 & \cos(\theta_2 - pi/2)cos\alpha_1 & -sin\alpha_1 & -sin\alpha_1 d(2) \\ sin(\theta_2 - pi/2)\ sin\alpha_1 & \cos(\theta_2 - pi/2)sin\alpha_1 & cos\alpha_1 & cos\alpha_1 d(2) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{2}_{3}T = \begin{matrix} cos\theta_3 & -sin\theta_3 & 0 & a(2) \\ sin\theta_3 cos\alpha_2 & cos\theta_3 cos\alpha_2 & -sin\alpha_2 & -sin\alpha_2 d(3) \\ sin\theta_3 sin\alpha_2 & cos\theta_3 sin\alpha_2 & cos\alpha_2 & cos\alpha_2 d(3) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{3}_{4}T = \begin{matrix} cos\theta_4 & -sin\theta_4 & 0 & a(3) \\ sin\theta_4 cos\alpha_3 & cos\theta_4 cos\alpha_3 & -sin\alpha_3 & -sin\alpha_3 d(4) \\ sin\theta_4 sin\alpha_3 & cos\theta_4 sin\alpha_3 & cos\alpha_3 & cos\alpha_3 d(4) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{4}_{5}T = \begin{matrix} cos\theta_5 & -sin\theta_5 & 0 & a(4) \\ sin\theta_5 cos\alpha_4 & cos\theta_5 cos\alpha_4 & -sin\alpha_4 & -sin\alpha_4 d(5) \\ sin\theta_5 sin\alpha_4 & cos\theta_5 sin\alpha_4 & cos\alpha_4 & cos\alpha_4 d(5) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{5}_{6}T = \begin{matrix} cos\theta_6 & -sin\theta_6 & 0 & a(5) \\ sin\theta_6 cos\alpha_5 & cos\theta_6 cos\alpha_5 & -sin\alpha_5 & -sin\alpha_5 d(6) \\ sin\theta_6 sin\alpha_5 & cos\theta_6 sin\alpha_5 & cos\alpha_5 & cos\alpha_5 d(6) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{6}_{EE}T = \begin{matrix} 1 & 0 & 0 & a(6) \\ 0 & cos\alpha_6 & -sin\alpha_6 & -sin\alpha_6 d(EE) \\ 0 & sin\alpha_6 & cos\alpha_6 & cos\alpha_6 d(EE) \\ 0 & 0 & 0 & 1 \end{matrix} $$

$$ {}^{0}_{EE}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T\,{}^{6}_{EE}T $$

Plugging theta values into ${}^{0}_{EE}T$ will generate solutions to forward kinematics problems. The first three entries in the fourth column of ${}^{0}_{EE}T$ represent the x, y, and z position of the end effector relative to the base link reference frame. The 3 x 3 matrix in the upper left corner of ${}^{0}_{EE}T$ is the orientation of the end effector relative to the base link reference frame. The ${}^{0}_{EE}T$ matrix is shown below.

```
Matrix([
[((sin(q1)*sin(q4) + sin(q2 + q3)*cos(q1)*cos(q4))*cos(q5) + sin(q5)*cos(q1)*cos(q2 +
q3))*cos(q6) - (-sin(q1)*cos(q4) + sin(q4)*sin(q2 + q3)*cos(q1))*sin(q6), -((sin(q1)*sin(q4) +
sin(q2 + q3)*cos(q1)*cos(q4))*cos(q5) + sin(q5)*cos(q1)*cos(q2 + q3))*sin(q6) +
(sin(q1)*cos(q4) - sin(q4)*sin(q2 + q3)*cos(q1))*cos(q6), -(sin(q1)*sin(q4) + sin(q2 +
q3)*cos(q1)*cos(q4))*sin(q5) + cos(q1)*cos(q5)*cos(q2 + q3), -0.303*sin(q1)*sin(q4)*sin(q5)
+ 1.25*sin(q2)*cos(q1) - 0.303*sin(q5)*sin(q2 + q3)*cos(q1)*cos(q4) - 0.054*sin(q2 +
q3)*cos(q1) + 0.303*cos(q1)*cos(q5)*cos(q2 + q3) + 1.5*cos(q1)*cos(q2 + q3) + 0.35*cos(q1)],
[ ((sin(q1)*sin(q2 + q3)*cos(q4) - sin(q4)*cos(q1))*cos(q5) + sin(q1)*sin(q5)*cos(q2 +
q3))*cos(q6) - (sin(q1)*sin(q4)*sin(q2 + q3) + cos(q1)*cos(q4))*sin(q6), -((sin(q1)*sin(q2 +
q3)*cos(q4) - sin(q4)*cos(q1))*cos(q5) + sin(q1)*sin(q5)*cos(q2 + q3))*sin(q6) -
(sin(q1)*sin(q4)*sin(q2 + q3) + cos(q1)*cos(q4))*cos(q6), -(sin(q1)*sin(q2 + q3)*cos(q4) -
sin(q4)*cos(q1))*sin(q5) + sin(q1)*cos(q5)*cos(q2 + q3),  1.25*sin(q1)*sin(q2) -
0.303*sin(q1)*sin(q5)*sin(q2 + q3)*cos(q4) - 0.054*sin(q1)*sin(q2 + q3) +
0.303*sin(q1)*cos(q5)*cos(q2 + q3) + 1.5*sin(q1)*cos(q2 + q3) + 0.35*sin(q1) +
0.303*sin(q4)*sin(q5)*cos(q1)],
[                                              -(sin(q5)*sin(q2 + q3) - cos(q4)*cos(q5)*cos(q2 +
q3))*cos(q6) - sin(q4)*sin(q6)*cos(q2 + q3),
(sin(q5)*sin(q2 + q3) - cos(q4)*cos(q5)*cos(q2 + q3))*sin(q6) - sin(q4)*cos(q6)*cos(q2 + q3),
-sin(q5)*cos(q4)*cos(q2 + q3) - sin(q2 + q3)*cos(q5),
-0.303*sin(q5)*cos(q4)*cos(q2 + q3) - 0.303*sin(q2 + q3)*cos(q5) - 1.5*sin(q2 + q3) +
1.25*cos(q2) - 0.054*cos(q2 + q3) + 0.75],
[
0,
0,                                                                    0,
1]])
```

Inverse Kinematics of Kuka KR210 Arm

Figure 2

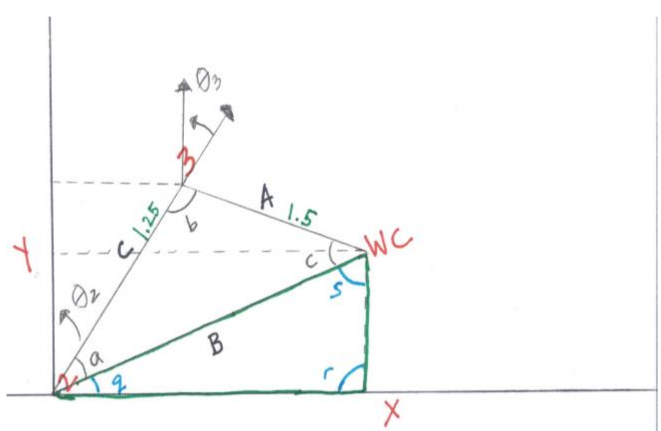
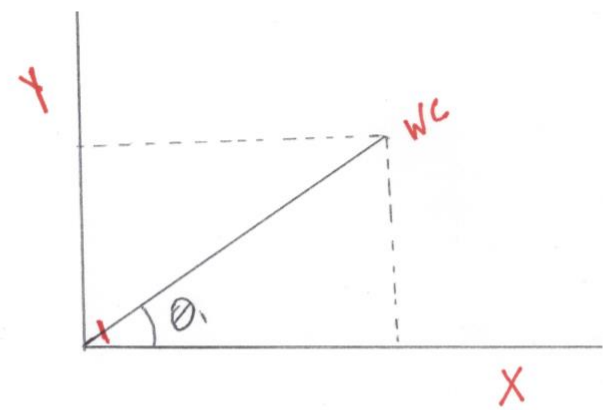


Figure 2 is a geometric representation of part of the Kuka arm. Joint 2, Joint 3, and the wrist center are the vertices of the triangle with sides A, B, and C. The Objective of inverse kinematics is to calculate joint angles from a given position of the end effector. The position of the wrist center is used to calculate theta 1, theta 2 and theta 3. To find the x, y, and z coordinates of the wrist center, the following calculations are made.

$$W_x = p_x - (l)n_x$$
$$W_y = p_y - (l)n_y$$
$$W_z = p_z - (l)n_z$$

W_x, W_y, W_z, are the x, y and z positions of the wrist center.
p_x, p_y, p_z , are the x, y, and z positions of the end effector.
n_x, n_y, n_z , are the x, y, and z positions of the z- axis of the gripper link.
l is the end effector length.

Figure 3



Theta 1 Calculation

Figure 3 shows joint 1 and the wrist center in the x – y coordinate frame. To find theta 1, the inverse tangent of the W_y / W_x is calculated, θ_1 = arctan (W_y / W_x).

Theta 2 Calculation

In figure 2, $\hat{x}$ and $\hat{y}$ are perpendicular which implies the angle between X and Y is 90 degrees or pi / 2. The 90-degree angle between X and Y is split into three pieces in the diagram, q, a, and theta 2. This implies θ_2 = pi/2 - q - a. To find angle a, the law of cosines is used. The law of cosines is used to find angles of a triangle when only the side lengths of the triangle are known.

$$a = \cos^{-1}\frac{A^2 - B^2 - C^2}{-2BC}$$

The lengths of sides A and C are given as 1.5 and 1.25 respectively. Side B is not given but can be calculated using the Pythagorean theorem. Side B can vary in three-dimensional space so the

X, Y and Z components will have to be considered when calculating B's magnitude. It also has to be taken into account that joint 2 is always 0.35 meters away from joint 1. When joint 1 rotates, it sweeps out a radius of length 0.35 meters, so this length will need to be subtracted from $\sqrt{{W_x}^2 + {W_y}^2}$, which creates the term $\sqrt{{W_x}^2 + {W_y}^2}$ - 0.35. The Z component of B is W_z - 0.75.

$$B = \sqrt{\left(\sqrt{{W_x}^2 + {W_y}^2} - 0.35\right)^2 + (W_z - 0.75)^2}$$

Angle q is equal to $\tan^{-1} \frac{W_z - 0.75}{\sqrt{{W_x}^2 + {W_y}^2} - 0.35}$

Since angles q and a are known, θ_2 is known.

$$\theta_2 = \frac{\pi}{2} - q - a$$

Theta 3 Calculation

Figure 4

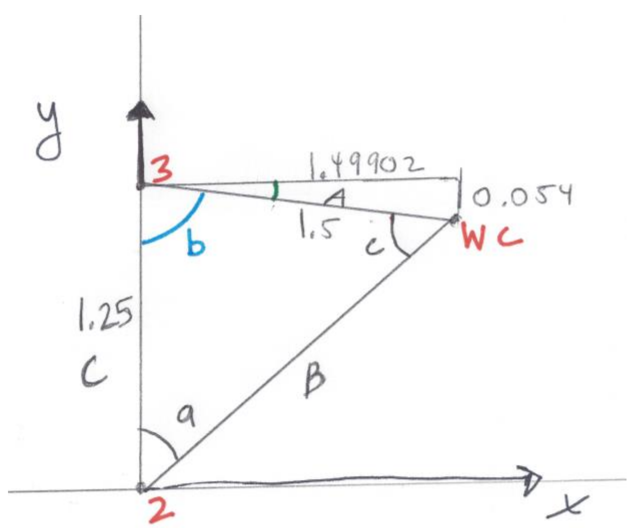


Figure 4 displays joint2, joint 3, and the wrist center as the vertices of the triangle ABC. In the picture θ_3 is oriented at 0. The change in angle b will affect the value of theta 3. When the value of angle b decreases the value of theta 3 increases and vice versa. When θ_3 is 0, the wrist center is lower than joint 3 in the Z-direction by 0.054 meters. Drawing a line parallel to the X axis from height of joint 3 to the x coordinate of the wrist center, then drawing a line parallel to Y from the wrist center to the y value of joint 3 creates a triangle with an angle of ~ 2.06 degrees or 0.036 radians represented by the small green arc. When angle b + 2.06 degrees is equivalent to 90 degrees, theta 3 is equal to 0 degrees. This implies that $\theta_3 = \frac{\pi}{2} - b - .036$.

Calculating theta 4, 5, and 6

The rotation matrix R_6^3 is used to calculate θ_4, θ_5 and θ_6. R_6^3 is the rotation portion of the homogeneous transform matrix T_6^3, specifically this is the 3 x 3 matrix in the upper left hand corner of T_6^3. R_6^3 is equal to $R_4^3 R_5^4 R_6^5$, which is a series of multiplied matrices.

$$R_6^3 = \begin{matrix} -s(\theta_4)s(\theta_6) + c(\theta_4)c(\theta_5)c(\theta_6) & -s(\theta_4)c(\theta_6) - s(\theta_6)c(\theta_4)c(\theta_5) & -s(\theta_5)c(\theta_4) \\ s(\theta_5)c(\theta_6) & -s(\theta_5)s(\theta_6) & c(\theta_5) \\ -s(\theta_4)c(\theta_5)c(\theta_6) - s(\theta_6)c(\theta_4) & s(\theta_4)s(\theta_6)c(\theta_5) - c(\theta_4)c(\theta_6) & s(\theta_4)s(\theta_5) \end{matrix}$$

Theta 5

Calculating θ_5 before θ_4 and θ_6 will allow for the correct signs to be applied to the arguments of the 'atan2' function that calculate θ_4 and θ_6. Whether $\sin\theta_5 < 0$ will determine the sign of the following values in the matrix R_6^3, $R_6^3[1,0]$, $R_6^3[1,1]$, $R_6^3[2,2]$, $R_6^3[2,1]$. This is because if $\sin\theta_5$ is < 0, $\sin\theta_5$ will be negative and the signs of the previously stated entries will need to be flipped since there is a $\sin\theta_5$ term in these matrix entries.

θ_5 can be calculated by taking the inverse cosine of $R_6^3[2,3]$, the problem with taking the inverse cosine of $R_6^3[2,3]$ is that there are two solutions. This is solved by using the 'atan2' function, 'atan2' will return the angle in the correct quadrant. To use the 'atan2' function, terms in the matrix will have to be combined and algebraically manipulated.

$$\frac{\sqrt{R_6^3[0,2]^2 + R_6^3[2,2]^2}}{R_6^3[1,2]}$$

The above term simplifies to $\frac{\sin\theta_5}{\cos\theta_5} = \tan\theta_5$

$$\rightarrow \theta_5 = atan2\left(\frac{\sqrt{R_6^3[0,2]^2 + R_6^3[2,2]^2}}{R_6^3[1,2]}\right)$$

Theta 4

To calculate θ_4, the entries $R_6^3[2,2] = s(\theta_4)s(\theta_5)$ and $R_6^3[0,2] = -s(\theta_5)c(\theta_4)$ are utilized. The signs of $R_6^3[2,2]$ and $R_6^3[2,1]$ will have to be adjusted to account for the value of $\sin\theta_5$. When

$\sin\theta_5$ is a negative value, $R_6^3[0,2] = s(\theta_5)c(\theta_4)$ and $R_6^3[2,2] = -s(\theta_4)s(\theta_5)$. Below is pseudo-code to show how the 'atan2' arguments are adjusted according to the value of $\sin\theta_5$.

if $\sin\theta_5 < 0$:

$$\theta_4 = atan2\frac{-R_6^3[2,2]}{R_6^3[2,1]}$$

else:

$$\theta_4 = atan2\frac{R_6^3[2,2]}{-R_6^3[2,1]}$$

Theta 6

θ_6 can be derived by utilizing entries $R_6^3[1,1]$ and $R_6^3[1,0]$, the algebra and trigonometric functions are shown below to isolate θ_6. Like θ_4, the calculation of θ_6 will depend on the value of $\sin\theta_5$.

$$R_6^3[1,1] = -s(\theta_5)s(\theta_6), \rightarrow -R_6^3[1,1] = s(\theta_5)s(\theta_6)$$
$$R_6^3[1,0] = s(\theta_5)c(\theta_6)$$

$$-R_6^3[1,1] \,/\, R_6^3[1,0] = s(\theta_6)/\,c(\theta_6) = \mathrm{Tan}(\theta_6)$$

Pseudo-code

if $\sin\theta_5 < 0$:

θ_6 = atan2 $(R_6^3[1,1] \,/\, -R_6^3[1,0])$

else:

θ_6 = atan2 $(-R_6^3[1,1] \,/\, R_6^3[1,0])$

IK_server.py Code

I give credit to the walkthrough video for assisting me with the code in the IK_server.py file.

The code in IK_server.py file calculates six joint angles (theta1-6) with inverse kinematics that are used to control the robotic arm in the simulation. The first part of the code contains information and algorithms that are responsible for forward kinematics calculations. Forward kinematics is utilized in the process of calculating the last three angles of the robotic arm. So, it is necessary to have some forward kinematics code in IK_server.py. Forward kinematics can also be used to debug the program, plugging in the angles that are generated from inverse kinematics back into the ${}_{EE}^{0}T$ matrix will allow for comparison to the Cartesian coordinate path values.

Information about the trajectory of the path is supplied in the form of Cartesian coordinates, as well as information about the orientation of the end effector. The code below displays how this information is supplied in the IK_server.py.

```python
joint_trajectory_list = []
for x in xrange(0, len(req.poses)):
    # IK code starts here
    joint_trajectory_point = JointTrajectoryPoint()

    # Extract end-effector position and orientation from request
    # px,py,pz = end-effector position
    # roll, pitch, yaw = end-effector orientation
    px = req.poses[x].position.x
    py = req.poses[x].position.y
    pz = req.poses[x].position.z

    (roll, pitch, yaw) = tf.transformations.euler_from_quaternion(
        [req.poses[x].orientation.x, req.poses[x].orientation.y,
            req.poses[x].orientation.z, req.poses[x].orientation.w])
```

The first three angles are found using the wrist position of the robotic arm. The math shown in the inverse kinematics section is used in the program. The values WC[0], WC[1], WC[2] represent the x, y, and z positions of the wrist center respectively. In the block of code above, a sequence of Cartesian coordinate values, and roll, pitch, yaw values are generated in the for loop. For each iteration of the loop, the six joint values are calculated with inverse kinematics and stored in the list 'joint_trajectory_list'.

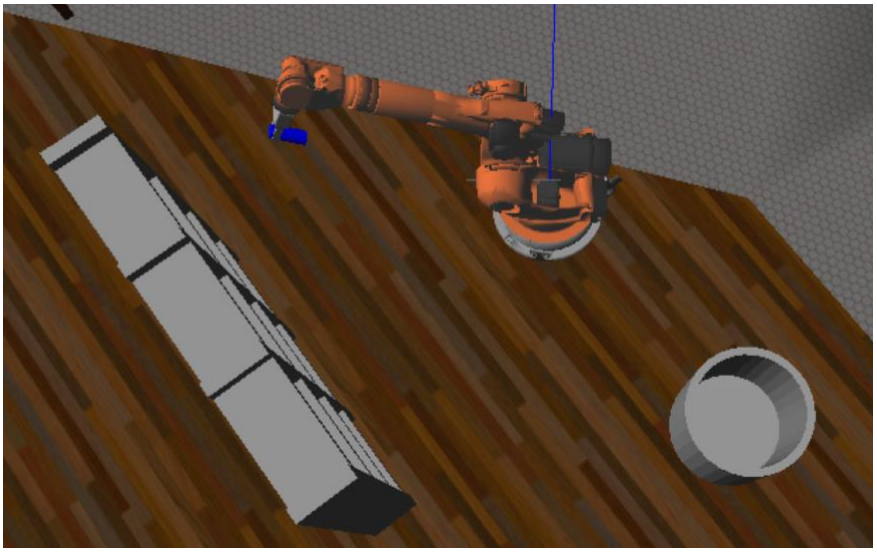
```python
# theta 1 calculation
theta1 = atan2(WC[1], WC[0])

# Sides of Triangle ABC
side_a = 1.501
side_b = sqrt(pow((sqrt(WC[0] * WC[0] + WC[1] * WC[1]) - 0.35), 2) + pow((WC[2] - 0.75), 2))
side_c = 1.25

# Law of Cosines to find angles of triangle ABC
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))

# theta 2 and theta 3 calculations
theta2 = pi / 2 - angle_a - atan2(WC[2] - 0.75, sqrt(WC[0] * WC[0] + WC[1] * WC[1]) - 0.35)
theta3 = pi / 2 - (angle_b + 0.036)
```

The angles theta1, theta2, and theta3 are then used as inputs in the R_3^0 rotation matrix, the inverse of R_3^0 is taken and then multiplies R_6^0 to isolate R_6^3. In the code R_6^0 is represented by ROT_EE. The theta4, theta5, and theta6 values are calculated with the math that was shown in the inverse kinematics section. In the block of code below theta5 is calculated first, and then there is a decision structure that follows to calculate theta4 and theta6. This 'if / else' structure was also discussed in the inverse kinematics section. Once all the theta values are calculated, they are stored and later used to guide the Kuka arm in the simulation.

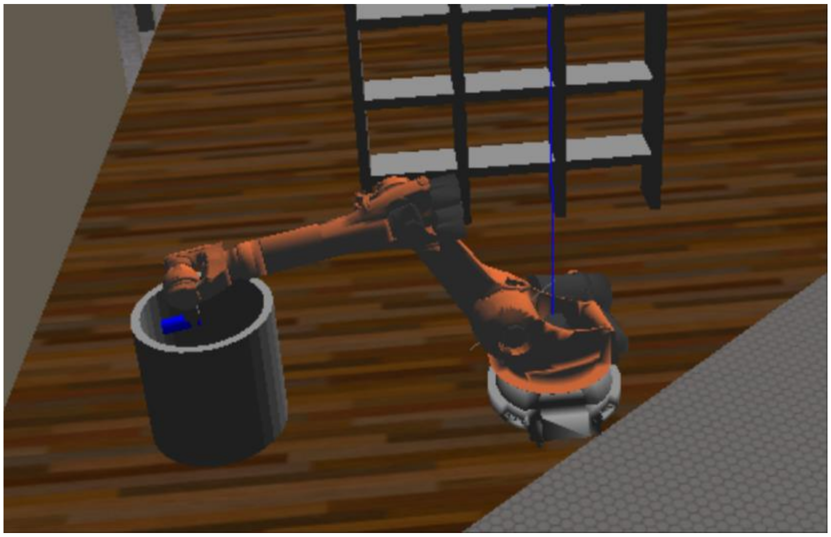
```python
# Extract rotation matrices from the transformation matrices

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

# Input theta1, theta2, and theta3 into R0_3
R0_3 = R0_3.evalf(subs={q1: theta1, q2: theta2, q3: theta3})

# Isolating R3_6
R3_6 = R0_3.inv("LU") * ROT_EE


# Calculate theta5 first to apply correct signs to entries of the rotation matrix
theta5 = atan2(sqrt(R3_6[0,2] * R3_6[0,2] + R3_6[2,2] *R3_6[2,2]), R3_6[1,2])

# Decision structure to apply correct signs to matrix entries
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])
```

Results

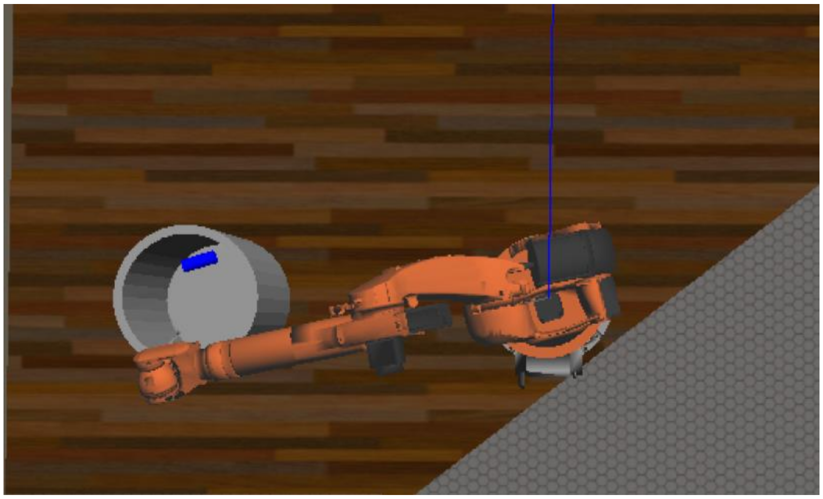
In the simulation, the Kuka KR210 arm can successfully pick up the cylinder from the shelf and place it into the bin. Initially there is a block placed on a random spot on a shelf, and a trajectory is created that the Kuka arm must follow to reach the cylinder on the shelf. The inverse kinematics code provides the joint angle values to achieve the task of moving along the correct trajectory to get to the cylinder. The gripper then grasps the cylinder and another trajectory is created that goes from the shelf to the bucket. In the image below, the Kuka arm has the cylinder in its grasp and is moving towards the bucket.



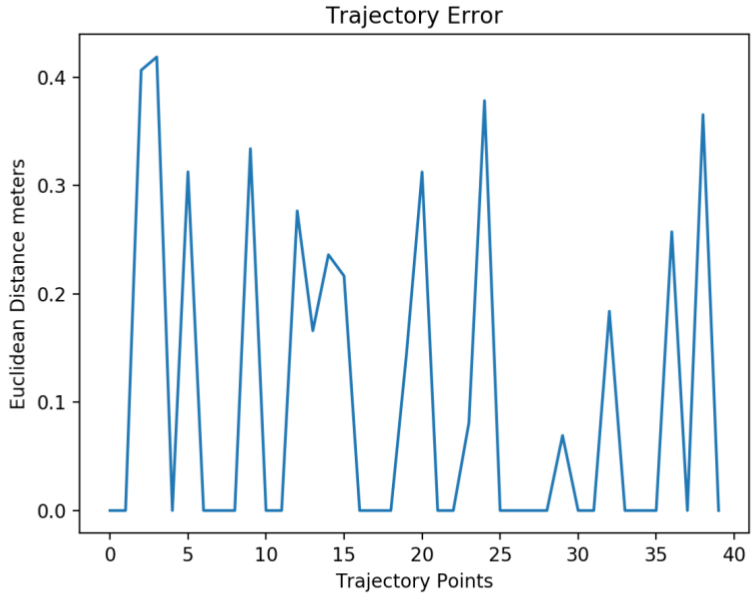
The image below shows the Kuka arm holding the cylinder over the bucket before it releases the cylinder from its grasp.



The image below shows the cylinder in the bucket. The tasks of following trajectories with inverse kinematics, picking up the cylinder from the shelf, and dropping it in the bucket have been completed.



Error Analysis



The chart above shows the Euclidean distance between the path that the Kuka arm took using the angles generated from inverse kinematics and the given path. The x-axis shows the number of positions that are given for a trajectory. The error oscillates from 0 - meters to ~ 0.40 -meters, the average error is ~ 0.104 meters. There is still room for improving the joint angles to get better accuracy.