

Udacity: Robotic Arm Pick & Place Report

Shane Reynolds

December 2, 2017

1 Introduction & Background

The Amazon Pick and Place Robotics Challenge is a competition designed to help increase collaboration between the industrial and robotics research communities. Amazon has successfully implemented a number of robotic systems which largely eliminate the need for activities like searching and walking in their fulfilment centres, however, one of the main challenges that Amazon is yet to solve is picking and stowing objects reliably in an unstructured environment. To successfully achieve this objective, there are a number of tasks that need to be successfully completed. These include:

1. Identification of the target object in the unstructured environment;
2. Manipulator path planning to the object;
3. Successful execution of a reach and grasp manoeuvre; and
4. Physical relocation of the grasped object to the desired location.



Figure 1: Kuka KR210 anthropomorphic industrial robot with 6 degrees of freedom

Path planning and execution of the desired manoeuvre is largely a solved problem. The move execution is dependent on a field of robotics called inverse kinematics. The inverse kinematics (IK) of a robot is the mathematical conversion of position in Cartesian space to the joint angles which allows the robot end effector to reach the desired position. Briefly, the end effector position in space can be thought of in 2 separate domains: Cartesian world coordinates, or Joint Angle space. Typically, analytical work is done in Cartesian space - three dimensional space is the native environment that humans live in and is easier to conceptualise. Robots, however, position themselves by making adjustments to electrical or hydraulic actuators - these actuators receive instructions based on Joint Angle. This project explores the IK derivation, and implementation, for the Kuka KR210. The

KR210 is a 6 degree of freedom (dof) anthropomorphic robotic arm shown in Figure 1. The project culminates with the implementation of an IK server, which is a ROS service receiving a series of points in Cartesian space (world coordinate frame), and returning a vector of Joint Angles after applying the IK transform. The implementation will be undertaken in ROS, which utilises simulation engines Rviz and Gazebo. Figures 2 and 3 show the Kuka KR210 in simulation.

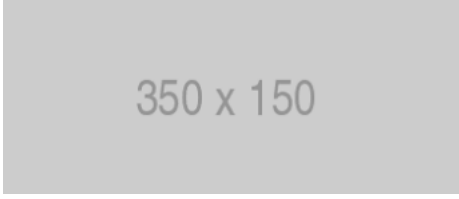


Figure 2: Include a picture of the KR210 in Gazebo



Figure 3: Include a picture of the KR210 in Rviz

2 Methods & Implementation

2.1 Determining the Denavit-Hartenburg Parameters

Forward and Inverse kinematic analysis relies heavily on successful specification of transformation matrices between the physical elements of the robot, which we call links. In order to determine these transformation matrices, we need to assign coordinate frames to the robot links. Doing this in an arbitrary fashion will often result in the determination of 6 parameters for each transformation matrix, which makes this process undesirably complex. Denavit and Hartenberg (1955) determined an algorithmic approach to the assignment of coordinate frames to the robot's links which reduces the number of parameters needed to describe each transformation matrix to 4. Assuming that \hat{x}_i , \hat{y}_i , and \hat{z}_i are the x , y , and z axes respectively for coordinate frame i , then these parameters are defined in Table 1.

Table 1: Description of the Denavit-Hartenburg parameters

Parameter	Description
α_{i-1}	Twist angle, and is determined by the angle between the \hat{z}_{i-1} and \hat{z}_i , measured about the \hat{x}_{i-1} axis
a_{i-1}	Distance from \hat{z}_{i-1} to \hat{z}_i measured along \hat{x}_{i-1} , where \hat{x}_{i-1} is orthogonal to \hat{z}_{i-1} , and \hat{x}_{i-1} is orthogonal to \hat{z}_i
d_i	Signed distance between \hat{x}_i and \hat{x}_{i-1} , measured along \hat{z}_i
θ_i	Angle between \hat{x}_{i-1} and \hat{x}_i , measured about \hat{z}_i

The KR210 has a base link, 6 degrees of freedom, and an end effector. Each of the links require a coordinate frame assignment, making a total requirement of 8 coordinate frames. Each of the joints were systematically labelled from 1 to 6, starting with the joint closest to the `base_link`. Following this, each of the links were assigned a number from 0 to 7. It must be noted that link 0 is actually the `base_link`, and link 7 is the `end_effector`. For the sake of simplicity, the `base_link` and `end_effector` will retain their names throughout this report. Coordinate frames were assigned to the links according to the DH procedure. Each link can be thought of as being associated with a

Table 2: DH parameter table

joint. The `base_link` is associated with the fixed ground, link 1 is associated with joint 1, and so on. To assign the coordinate frame to a link, DH requires the \hat{z}_i coordinate axis for link i to pass through the joint i axis of rotation. Finally, starting at the `base_link`, the \hat{x}_i axes are assigned dependent on whether the \hat{z}_i , and \hat{z}_{i+1} axes are:

1. Skewed;
2. Intersecting; or
3. Coincident

To reiterate, the \hat{x}_i axis is assigned dependent on the geometric how the \hat{z}_i axis, and the **next link coordinate frame**, \hat{z}_{i+1} , interact with each other geometrically. The full DH coordinate frame assignment can be seen in Figure 4.

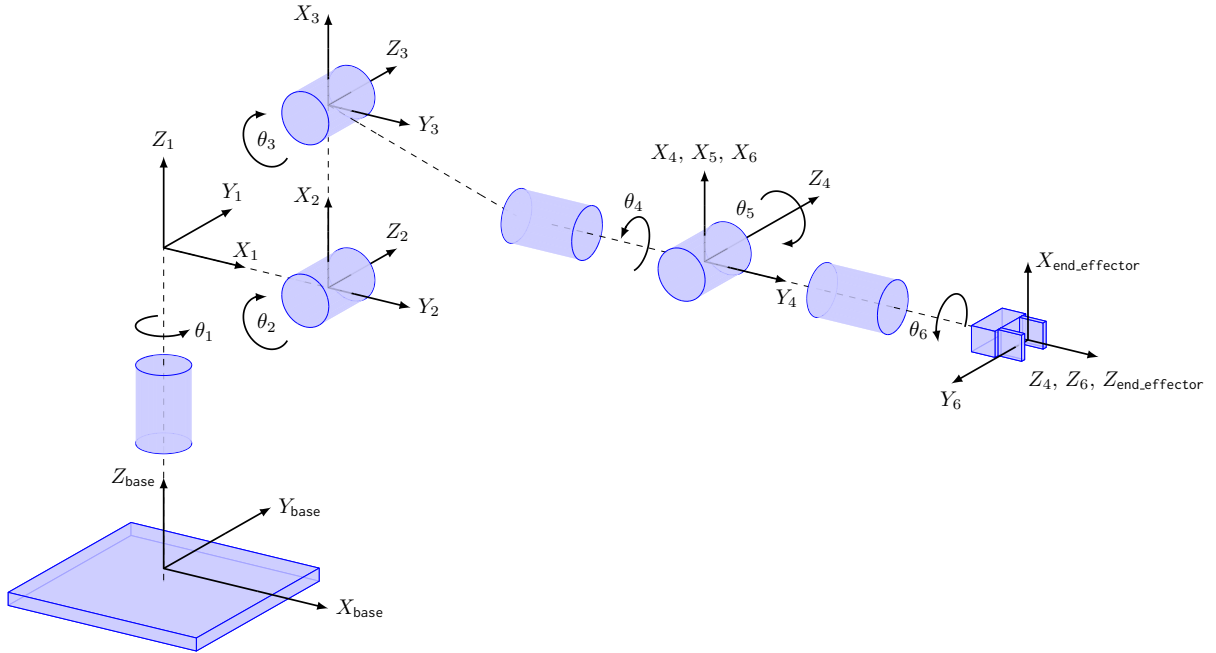


Figure 4: Sexy robot drawing

2.2 Forward Kinematics

Forward kinematics is the transformation of joint angles to an end effector point in space. Provide an in depth description of how each of the transformations and how they were arrived at. Recall this is a simple implementation of the DH parameter matrices and then a simple multiplication (which can be done using sympy)

The individual transformation matrices are shown below:

$$\begin{aligned}
{}^{\text{base_link}}T_1 &= \text{MATRIXARRAY} \\
{}^1T_2 &= \text{MATRIXARRAY} \\
{}^2T_3 &= \text{MATRIXARRAY} \\
{}^3T_4 &= \text{MATRIXARRAY} \\
{}^4T_5 &= \text{MATRIXARRAY} \\
{}^5T_6 &= \text{MATRIXARRAY} \\
{}^5T_{\text{end_effector}} &= \text{MATRIXARRAY}
\end{aligned}$$

The full transformation from the `base_link` to the `end_effector` is given by simple matrix multiplication as follows:

$${}^{\text{base_link}}T_{\text{end_effector}} = {}^{\text{base_link}}T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^5T_{\text{end_effector}}$$

2.3 Inverse Kinematics

Inverse kinematics is the process of determining the joint angles for each of the degrees of freedom. In the case of the Kuka XXXX there are 6 degrees of freedom, and hence, there are 6 joint angles which need to be determined. The anthropomorphic arm design allows us to exploit the fact that the last 3 joints are a spherical wrist (their axes of rotation intersect?), which means that these joints do not have an influence on the position of the wrist centre. This allow us to kinematically decouple the the first three joints and the last three joints, providing the ability to find a closed form solution to the problem. The closed form solution is presented below, in two parts: first three joint angles, and final three joint angles

2.3.1 First Three Joint Angles (θ_1 , θ_2 , and θ_3)

Mathematics to show the derivation of the first three angles

2.3.2 Final Three Joint Angles (θ_4 , θ_5 , and θ_6)

Mathematics to show the derivation of the final three angles

3 Results & Conclusion

Discussion of the results of the robot kinematics.

4 Further Enhancements

The wrist turns around a bit - need to make sure that this doesn't happen and why this is happening. It happens because the wrist has more than 360 twist capability, however, the way it is set up does not allow for this (it has hard clamps at $-\pi$ and π). Also an error tracking capability would be nice on the project also.