Project: Kinematics Pick & Place

Brett Gleason

July 16, 2017

The rubric for this project can be found at the following URL:
https://review.udacity.com/#!/rubrics/972/view
I will consider the rubric points individually and describe how I addressed each point in my implementation.

Writeup / README

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf.

You're reading it!

Kinematic Analysis

1. 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.

Using the model of the Kuka KR210 robotic arm in the forward kinematics demo as well as the description of the joints within the URDF file, a schematic diagram of the robot can be drawn.

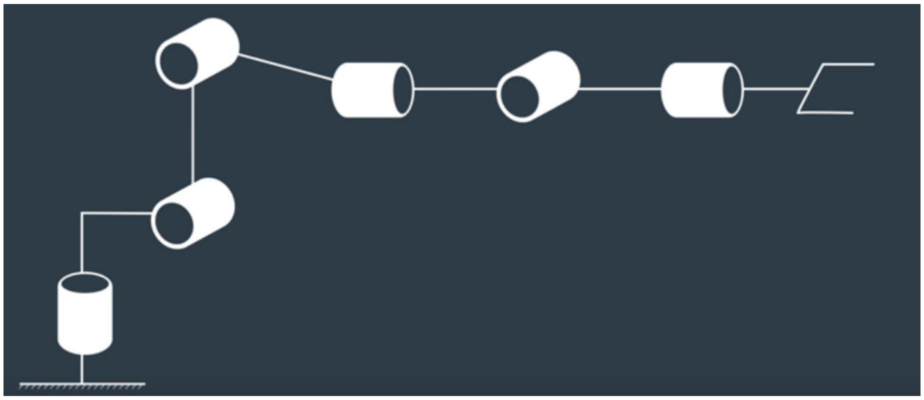

Figure 1: Basic schematic as shown in project lesson 10

Next the joints are labeled from 1 to n and the links are labeled from 0 to n.

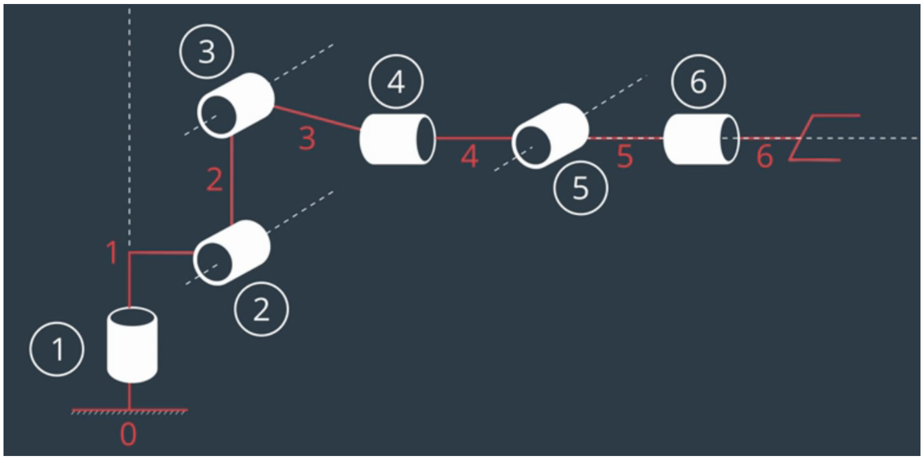

Figure 2: Schematic showing joint and link numbers as shown in project lesson 10

After the joints and links are labeled, reference frames can be defined for each joint.

Transform	α_{i-1}	a_{i-1}	d_i	θ_i
T_1^0	0	0	0.75	θ_1
T_2^1	$-\frac{\pi}{2}$	0.35	0	$\theta_2 - \frac{\pi}{2}$
T_3^2	0	1.25	0	θ_3
T_4^3	$-\frac{\pi}{2}$	-0.054	1.5	θ_4
T_5^4	$\frac{\pi}{2}$	0	0	θ_5
T_6^5	$-\frac{\pi}{2}$	0	0	θ_6
T_G^6	0	0	0.303	0

Table 1: Denavit-Hartenberg parameter table with values derived from the URDF file

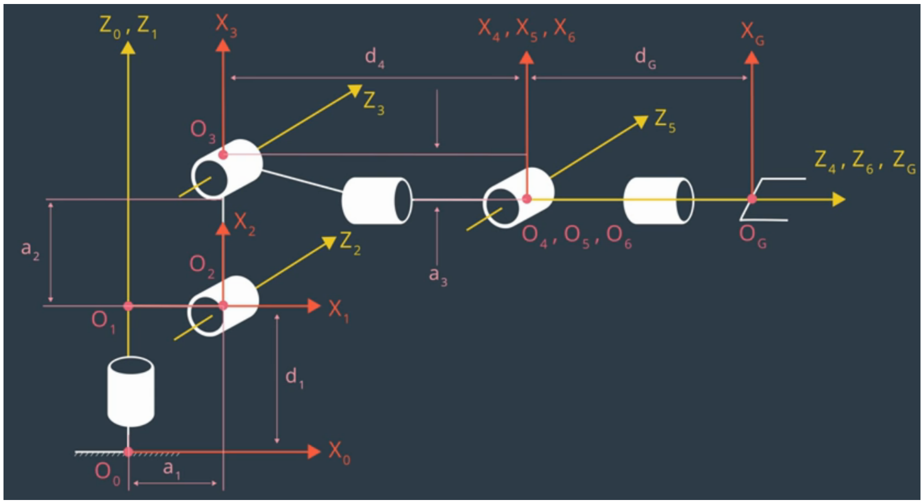

Figure 3: Schematic showing reference frames for each joint as shown in project lesson 10

Using the reference frames the Denavit-Hartenberg parameters can be defined. For this project the DH parameters are defined using the convention described by John J. Craig in his book Introduction to Robotics: Mechanics and Control. The definitions are as follows (from lesson 2 section 12):

- Twist angle (α_{i-1}): angle between $\hat{Z}_{i-1}$ and $\hat{Z}_i$ measured about $\hat{X}_{i-1}$ in a right hand sense.
- Link length (a_{i-1}): distance from $\hat{Z}_{i-1}$ to $\hat{Z}_i$ measured along $\hat{X}_{i-1}$.
- Link offset (d_i): signed distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$.
- Joint angle: angle between $\hat{X}_{i-1}$ and $\hat{X}_i$ measured about $\hat{Z}_i$ in a right hand sense.

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.

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

Theta 1

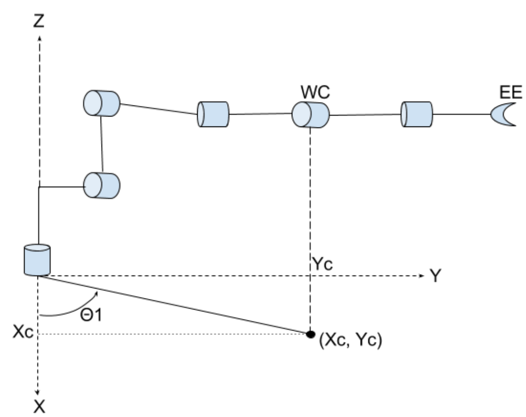

Figure 4: Diagram for calculating theta 1.

Theta 2

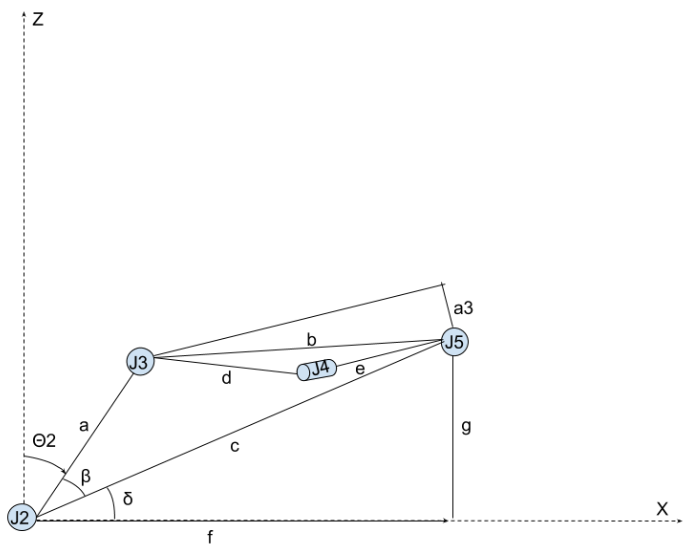

Figure 5: Diagram for calculating theta 2.

Theta 3

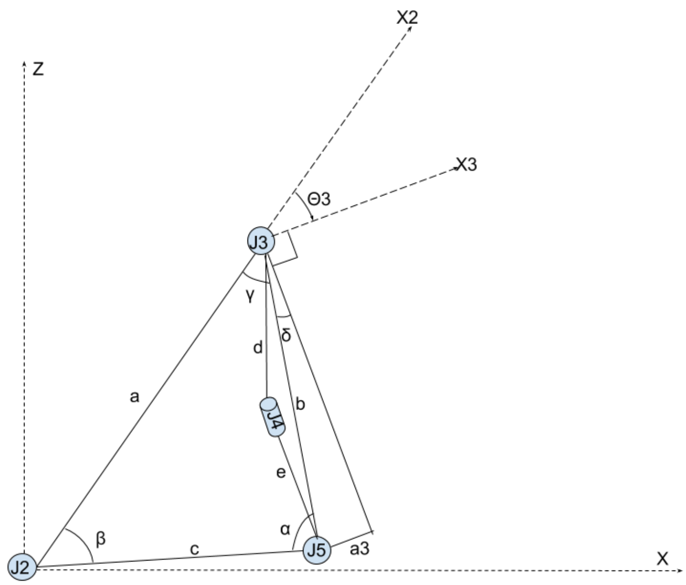

Figure 6: Diagram for calculating theta 3.

Theta 4

Theta 5

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.

Here I'll talk about the code, what techniques I used, what worked and why, where the implementation might fail and how I might improve it if I were going to pursue this project further.

And just for fun, another example image: