Lab 3: Baxter Pick and Place

1 Overview

In this lab, you will program the Baxter robot to perform a pick-and-place task by solving a simplified inverse kinematics (IK) problem for the Baxter robot's left arm. The Baxter arm is 7-DoF and is therefore kinematically redundant, resulting in an infinite number of solutions for IK of a given pose. In order to make an analytical solution possible, we have fixed $\theta_3 = 0$, thereby reducing the robot to a 6-DoF arm.

You need to write a Matlab function `theta = ikinelbow(a,d,Tw_tool)` that computes a joint angle IK solution of the arm (6 joint angles) given a desired pose of the endpoint (tool frame) with respect to the world frame. For Baxter, the tool frame is located at the midpoint between the tips of the parallel grippers. `a` and `d` are arrays containing the non-zero DH length parameters (from the table and parameter definitions below) and are passed to the function as inputs. The output of your function should be the 6 joint angles that solve the IK for the input pose.

Multiple joint angle solutions may exist for a given pose, i.e. "Lefty"/"Righty", "Elbow-Up"/"Elbow-Down", etc. Due to Baxter's angle limitations, your function can be simplified to solve for only the following configuration: "Lefty", "Elbow Down", and "No-Flip" (joint 5 angle is in quadrants 1 or 4). Make sure to handle exceptional cases, such as a desired position that is outside of the workspace.

2 DH Parameters

The following DH parameters specify the simplified 6-DoF structure for the left arm of Baxter:

i	a_i	d_i	α_i	θ_i
1	a_1	d_1	$-\pi/2$	$*$
2	a_2	0	0	$*$
3	0	0	$-\pi/2$	$*$
4	0	d_4	$\pi/2$	$*$
5	0	0	$-\pi/2$	$*$
6	0	0	0	$*$
tool	0	d_6	0	0

where, $a_1 = 69mm$, $d_1 = 270.35mm$, $a_2 = 370.82mm$, $d_4 = 374.29mm$ and $d_6 = 368.3mm$. Also, the transformation matrix from the world coordinate to 0 coordinate is defined as:

$$
{}^{w}T_0 \quad = \quad \begin{bmatrix} \sqrt{(2)}/2 & \sqrt{(2)}/2 & 0 & L \\ -\sqrt{(2)}/2 & \sqrt{(2)}/2 & 0 & h \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

where $L = 221\ mm$, $h = 22\ mm$ and $H = 1104\ mm$.

3 Testing

You may use the following test case to test if your function is working:

$$
{}^{w}T_{tool} \;=\; \begin{bmatrix} 0.984 & 0.177 & 0.024 & 727.089 \\ 0.177 & -0.984 & 0.020 & 418.581 \\ 0.027 & -0.015 & -0.999 & 732.233 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

The above case correspond to example end-effector poses (positions and orientations) that would be needed in a pick-and-place task on a table: above object. We have provided these poses in the file `poses.mat`.

4 Real Robot

You will run your code in the lab to get the real Baxter robot to reach for an object on a table, pick the object up, and place back on a different location on the table. The work described in previous sections should be completed before your scheduled lab time. When you come to the lab, bring your code on a flash drive or a cloud service that you can easily download it onto the lab computer.

Your code should read poses from a `.mat` that is in the same format as the provided file `poses.mat`. In the end you need to save your IK solutions to a file with extension `.txt` such that you save one IK solution per line with joint angles separated by commas. For example, our `poses.mat` has 5 poses, so your output would look like the following (note the values are made up):

```
0.1, 0.2, 0.3, 0.4, 0.5, 0.6
0.1, 0.2, 0.3, 0.4, 0.5, 0.6
0.1, 0.2, 0.3, 0.4, 0.5, 0.6
0.1, 0.2, 0.3, 0.4, 0.5, 0.6
0.1, 0.2, 0.3, 0.4, 0.5, 0.6
```