UDACITY

API Access Dashboard Logout

PROJECT

Robotic Arm: Pick & Place

A part of the Robotics Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Very good job !

I gave you some advice to improve even better your project.

Advanced tips:
Concerning the Jacobian matrix, here some great resources:

- Theoretical approach, and pseudo-inversion which is what we use to use; http://www.cs.cmu.edu/~15464-s13/lectures/lecture6/IK.pdf
- Example of computation and inversion: http://www.cs.columbia.edu/~allen/F15/NOTES/jacobians.pdf

Good luck for the next project !

Writeup

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

AMAZING !
The report is clean, clear and with illustrations.

Kinematic Analysis

Your writeup should contain a DH parameter table with proper notations and description about how you obtained the table. Make sure to use the modified DH parameters discussed in this lesson. Please add an annotated figure of the robot with proper link assignments and joint rotations (Example figure provided in the writeup template). It is strongly recommended that you use pen and paper to create this figure to get a better understanding of the robot kinematics.

Good job, you found all parameters, and clear explanation

Your writeup should contain individual transform matrices about each joint using the DH table and a homogeneous transform matrix from base_link to gripper_link using only the position and orientation of the gripper_link. These matrices can be created using any software of your choice or hand written. Also include an explanation on how you created these matrices.

Good and effecient !

Based on the geometric Inverse Kinematics method described here, breakdown the IK problem into Position and Orientation problems. Derive the equations for individual joint angles. Your writeup must contain details about the steps you took to arrive at those equations. Add figures where necessary. If any given joint has multiple solutions, select the best solution and provide explanation about your choice (Hint: Observe the active robot workspace in this project and the fact that some joints have physical limits).

Good !
But I would advise:

- Instead of arcos, or artan, we use to use arctan2 which has better angle properties than arcos or even arctan (https://en.wikipedia.org/wiki/Atan2)

You have to take attention to the fact that multiple solutions possible.
Thus there some hints:

```
q5 = atan2(sqrt(r13**2 + r33**2), r23)
if sin(q5) < 0:
    q4 = atan2(-r33, r13)
    q6 = atan2(r22, -r21)
else:
    q4 = atan2(r33, -r13)
    q6 = atan2(-r22, r21)
```

Project Implementation

`IK_server.py` must contain properly commented code. The robot must track the planned trajectory and successfully complete pick and place operation. Your writeup must include explanation for the code and a discussion on the results.

Good implementation!
Here some points to improve your code:
To reduce the complexity of the main loop, and so increase the speed, you can:

- put all the computation which don't depend on the "x", so that the creation of transformation matrices, DH parameters, ... What you did
- save matrices, which can be loaded again, on the next call, like this: https://wiki.python.org/moin/UsingPickle

```
if not os.path.exists("R0_g_inv.p"):
    R0_g_sym = ...
    pickle.dump(R0_g_sym, open("R0_g_sym.p", "wb"))
else:
    p2_0_sym= pickle.load(open("R0_g_sym.p", "rb"))
```

DOWNLOAD PROJECT

Rate this review

RETURN TO PATH

Project Reviewer Certification Course Reviewer FAQ Student FAQ Reviewer Agreement