ME 449 Assignment 1

Xuedong Fan

October 15, 2024

1 Part 1B

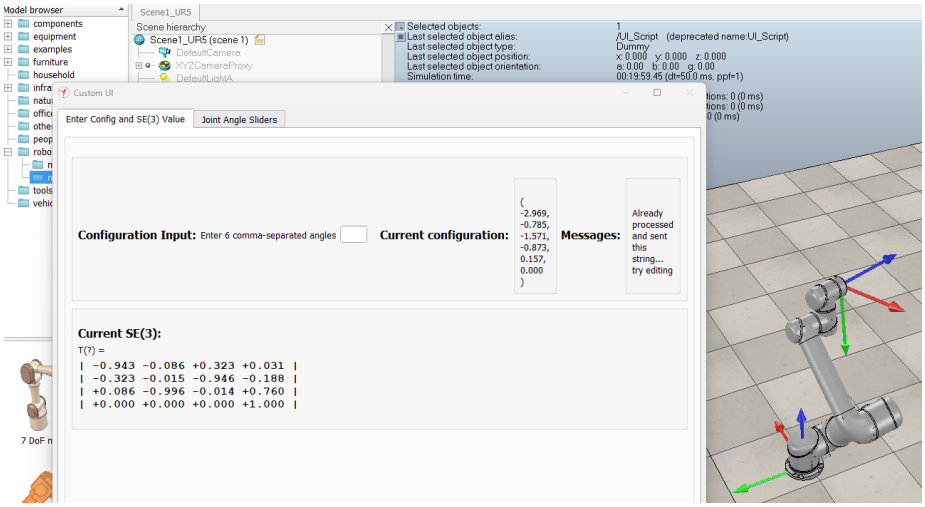
```
      <tab title="Enter Config and SE(3) Value">
        <group>
          <group layout="hbox">
            <label text="<big> Configuration Input:</big>" wordwrap="false" style="font-wei
```

Here I changed the

layout from 'vbox' to the 'hbox' (1)

Text from ' Entry' to 'Input' (2)

2 Part 1B & 2



3 Part 2

```python
import numpy as np
import modern_robotics as mr

R_13 = np.array([[-0.7071, 0, -0.7071], [0, 1, 0], [0.7071, 0, -0.7071]])
R_s2 = np.array([[-0.6964, 0.1736, 0.6964],[-0.1228, -0.9848, 0.1228],[0.7071, 0, 0.7071]])
R_25 = np.array([[-0.7566, -0.1198, -0.6428], [-0.1564, 0.9877, 0], [0.6348, 0.1005, -0.7661]])
R_12 = np.array([[0.7071, 0, -0.7071], [0, 1, 0], [0.7071, 0, 0.7071]])
R_34 = np.array([[0.6428, 0, -0.7660], [0, 1, 0], [0.7660, 0, 0.6428]])
R_s6 = np.array([[0.9418, 0.3249, -0.0859], [0.3249, -0.9456, -0.0151], [-0.0861, -0.0136, -0.9962]])
R_6b = np.array([[-1, 0, 0], [0,0,1], [0, 1, 0]])

R_21 = mr.RotInv(R_12)
R_s1 = R_s2 @ R_21

R_23 = R_21 @ R_13

R_34 = R_34

R_43 = mr.RotInv(R_34)
R_32 = mr.RotInv(R_23)
R_45 = R_43 @ R_32 @ R_25

R_52 = mr.RotInv(R_25)
R_2s = mr.RotInv(R_s2)
R_56 = R_52 @ R_2s @ R_s6

R_6b = R_6b

R_sb = R_s1 @ R_12 @ R_23 @ R_34 @ R_45 @ R_56 @ R_6b


def GetTheta(R):
    so3 = mr.MatrixLog3(R)
    OmgTheta = mr.so3ToVec(so3)
    [Axis,Theta]= mr.AxisAng3(OmgTheta)
    return [Axis,Theta]

Thetalist = np.array([GetTheta(R_s1)[1]*(-1),GetTheta(R_12)[1]*(-1),GetTheta(R_23)[1]*(-1),GetTheta(R_34)[1]*(-1),GetTheta(R_45)[1],GetTheta(R_56)[1]])
```

The principle steps are as follows:

1. Use the RotInv function in modern_robotics library to inverse the subscript.
2. Follow the subscript cancellation rule to get all the rotation matrix of the two adjacent joints, e.g., R_{12}, R_{56}.
3. Gather them up, multiplying them in sequence to get the R_{sb}.
4. Use the 'MatrixLog3' function to turn all adjacent SO(3) matrices into so(3) ones. Then, use 'so3ToVec' to get $\omega\theta$.
5. Finally, 'AxisAng3' to get its Axis and Angle
6. An important step is to check the Axis result from the code with the Axis listed on the wiki, if they're that opposite, we should firstly multiply (-1) then plug into the CoppeliaSim

```
>>> GetTheta(R_s1)
[array([ 4.92745571e-18,  1.60392656e-17, -1.00000000e+00]), 2.969482157066879]
>>> GetTheta(R_12)
[array([ 0., -1.,  0.]), 0.7853926894212007]
>>> GetTheta(R_23)
[array([ 0., -1.,  0.]), 1.5707661989213484]
>>> GetTheta(R_34)
[array([ 0., -1.,  0.]), 0.8726096667837093]
>>> GetTheta(R_45)
[array([ 7.81401776e-05, -4.79455853e-04, -9.99999882e-01]), 0.15702981867582216]
>>> GetTheta(R_56)
[array([ 0.03449233,  0.04128073, -0.99855204]), 2.191687195265858e-05]
>>> Thetalist
array([-2.96948216e+00, -7.85392689e-01, -1.57076620e+00, -8.72609667e-01,
        1.57029819e-01,  2.19168720e-05])
>>> R_sb
array([[-0.94166446, -0.08588191,  0.32480355],
       [-0.32494382, -0.01511253, -0.94558598],
       [ 0.08609978, -0.99617438, -0.01360934]])
>>>
```

Just as the screenshot shown above, the $joint_1, joint_2, joint_3, joint_4$ has the opposite Axis as the Axis given by Wiki, then we need to multiply (-1) first

Finally we have the R_sb shown at the bottom of screenshot, it almost perfectly match the upper left $3*3$ rotation matrix of the SE3 given by CoppeliaSim