Assignment 1 Report

Yifei Chen

October 2024

Introduction

This report contains the required screenshots, explanations, and calculations for Assignment 1. The following sections are organized according to the parts mentioned in the assignment description.

Part 1B: Changes in the Child Script

Explanation of Changes

Below is a screenshot of the modified code in the child script. The changes made include modifications to the scene/UI to enhance visualization. I modified the text **Configuration Entry** to **Configuration Input:** . And I also add the font size of the text to 20 and 18.

```
xml = [[
<ui closeable="false" on-close="closeEventHandler" resizable="true">
  <tabs>

    <tab title="Enter Config and SE(3) Value">
      <group>
        <group layout="vbox">
          <label text="<font size=20> Configuration Input:</font>" wordwrap="false" style=
          <label text="Enter 6 comma-separated angles" />
          <edit value="" id="7006" oneditingfinished="fulljointEntry" />
          <label value="" id="5006" wordwrap="false" />
          <label text="<font size=18> Current configuration:</font>" id="6007" wordwrap="
          <group layout="vbox">
            <label value="" id="1237" wordwrap="true" />
          </group>
          <label text="<font size=18> Messages:</font>" id="6006" wordwrap="false" style=
          <group layout="hbox">
            <label value="" id="1236" wordwrap="true" />
          </group>
        </group>
        <group>
          <!-- <group> -->
            <label text="<font size=18> Current SE(3):</font>" id="6008" wordwrap="false"
            <!-- <button text="Calculate SE(3) transform:" on-click="calcSE3" id="1235"/>
            <label text="T(?) = " wordwrap="false" />
            <label text="" id="1234" wordwrap="false" />
          <!-- </group> -->
           <!-- <group> -->
           <!--   <label text="<font size=18> Settings:</font>" wordwrap="false" style="f
           <!--   <checkbox text="Use degrees instead of radians?" checked="false" on-cha
           <!-- </group> -->
        </group>
        <stretch />
      </group>
    </tab>
```

Figure 1: Modified code in the child script.

Part 1B & 2: Modified Scene and SE(3) Calculation

Screenshot of the Scene

The following screenshots show the modified scene, with the updated UI elements, the SE(3) calculation results, and the robot in the correct configuration.

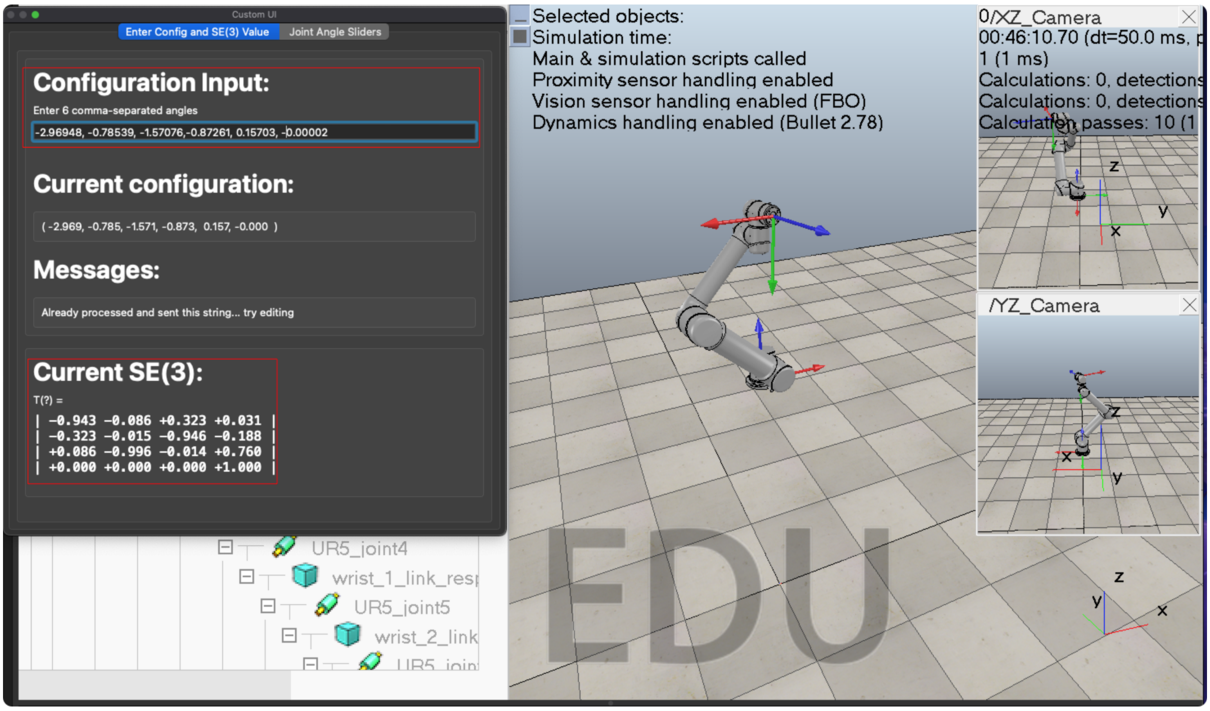

Figure 2: Modified scene showing the updated UI, SE(3) calculation, and robot configuration.

Figure 3: Modified scene showing the updated UI

Part 2: Joint Angles and Rsb Calculation

Python Code for calculate θ_{ij} and SE(3)

The following code snippet shows the implementation used to calculate the θ_{s1} and θ_{ij}. The function take in $R \in SO(3)$, and use `mr.MatrixLog3()` function convert it to $so(3)$ skew-symmetric matrix.

$$S = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{1}$$

Then we can extract ω from it to compute $\theta = ||\omega||$:

```python
def Rotation2angle(R):
    S = mr.MatrixLog3(R)
    print("so(3) = \n", S)
    omega_x = -S[2][1]
    omega_y = S[2][0]
    omega_z = -S[1][0]
    omega = np.array([[omega_x], [omega_y], [omega_z]])
    print("omega = \n",omega)
    theta = np.sqrt(omega_x**2 + omega_y**2 + omega_z**2)

    return theta
```

Listing 1: Python code to compute theta

List of Joint Angles

Using the above function and the given matrix, we can calculated the joint angles θ_{s1} and $\theta_{i,i+1}$. Some of the rotation matrix can be calculate using the given matrix(The inverse operation is using `mr.RotInv()`):

- $R_{s1} = R_{s2}R_{12}^{-1}$
- $R_{23} = R_{12}^{-1}R_{13}$
- $R_{45} = (R_{23}R_{34})^{-1}R_{25}$
- $R_{56} = (R_{s2}R_{25})^{-1}R_{s6}$

Also based on defined rotation axis of each joint: $\hat{\omega}_1 = (0, 0, 1)$, $\hat{\omega}_5 = (0, 0, -1)$, $\hat{\omega}_2 = \hat{\omega}_3 = \hat{\omega}_4 = (0, 1, 0)$

```python
# Rs1
Rs1 = np.dot(Rs2,mr.RotInv(R12))
theta_s1 = -Rotation2angle(Rs1)

# R12
theta12 = -Rotation2angle(R12)

# R23
R23 = np.dot(mr.RotInv(R12),R13)
theta_23 = -Rotation2angle(R23)

# R34
theta_34 = -Rotation2angle(R34)

# R45
R24 = np.dot(R23,R34)
R45 = np.dot(mr.RotInv(R24),R25)
theta_45 = Rotation2angle(R45)
```

```

# R56
Rs5 = np.dot(Rs2,R25)
R56 = np.dot(mr.RotInv(Rs5),Rs6)
theta_56 = -Rotation2angle(R56)
```

Listing 2: Python code to compute theta

Here is the result angle in radius:

- $\theta_{s1} = -2.969482157066879$
- $\theta_{12} = -0.7853926894212007$
- $\theta_{23} = -1.5707661989213484$
- $\theta_{34} = -0.8726096667837093$
- $\theta_{45} = 0.15702981867582216$
- $\theta_{56} = -2.191687195265858e - 05$

R_{sb} Matrix

The calculated R_{sb} matrix can be compute and verified use the following code:

```
Rsb = np.dot(Rs6,R6b)
Rsb_verify = Rs1 @ R12 @ R23 @ R34 @ R45 @ R56 @ R6b
```

Listing 3: Python code to compute Rsb

And we can get the following result:

$$R_{sb} = \begin{bmatrix} -0.9418 & -0.0859 & 0.3249 \\ -0.3249 & -0.0151 & -0.9456 \\ 0.0861 & -0.9962 & -0.0136 \end{bmatrix} \tag{2}$$

The R_{sb} is also shown in 2 SE(3) upper left:

$$R_{sb} = \begin{bmatrix} -0.943 & -0.086 & +0.323 & +0.031 \\ -0.323 & -0.015 & -0.946 & -0.188 \\ +0.086 & -0.996 & -0.014 & +0.760 \\ +0.000 & +0.000 & +0.000 & +1.000 \end{bmatrix} \tag{3}$$

Due to the small error term during the input accuracy of CoppeliaSim and the small term θ_{56}, it may have some small difference. However, the overall sign and 2 decimal number are correct.