Assignment 2 - Christopher Luey

I. Implemented IKinBodyIterates. Used IKinBody code as a template and added the following:

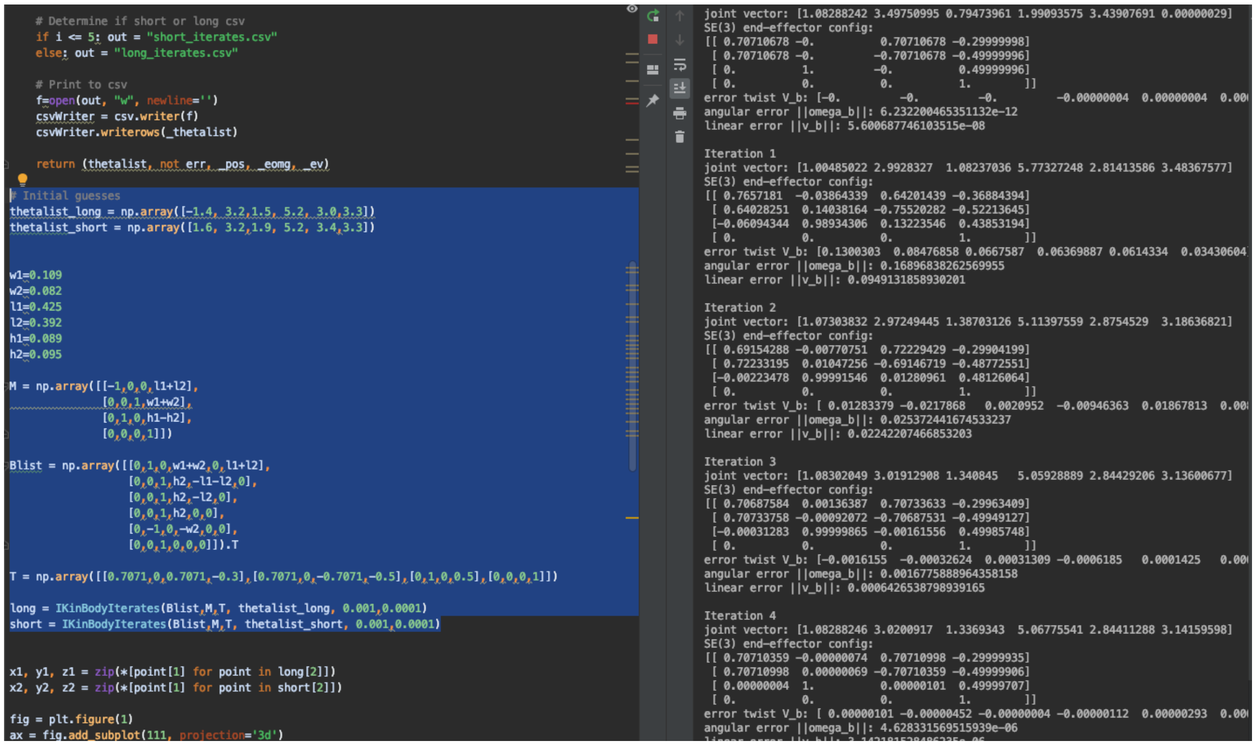
1. Print statements during each iteration
2. Normalization of joint angles between (-2pi, 2pi)
3. Arrays to track the joint angles, end effector position, angular and linear error
4. CSV output of joint angles depending on short or long iteration
5. Matplotlib graphing using the added arrays to display data

II. Initial guesses:

thetalist_long = np.array([-1.4, 3.2,1.5, 5.2, 3.0,3.3])

thetalist_short = np.array([1.6, 3.2,1.9, 5.2, 3.4,3.3])

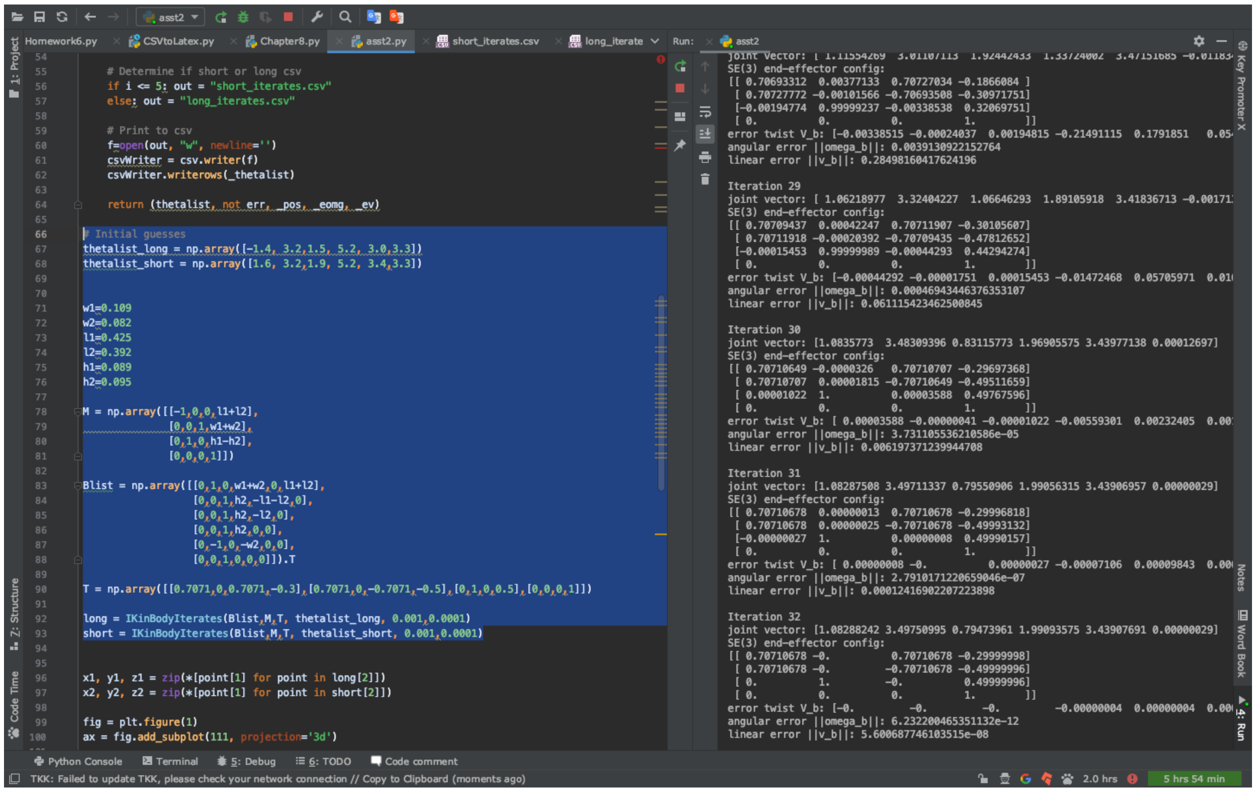
Short:



```python
    # Determine if short or long csv
    if i <= 5: out = "short_iterates.csv"
    else: out = "long_iterates.csv"

    # Print to csv
    f=open(out, "w", newline='')
    csvWriter = csv.writer(f)
    csvWriter.writerows(_thetalist)

    return (thetalist, not err, _pos, _eomg, _ev)

# Initial guesses
thetalist_long = np.array([-1.4, 3.2,1.5, 5.2, 3.0,3.3])
thetalist_short = np.array([1.6, 3.2,1.9, 5.2, 3.4,3.3])


w1=0.109
w2=0.082
l1=0.425
l2=0.392
h1=0.089
h2=0.095

M = np.array([[-1,0,0,l1+l2],
              [0,0,1,w1+w2],
              [0,1,0,h1-h2],
              [0,0,0,1]])

Blist = np.array([[0,1,0,w1+w2,0,l1+l2],
                  [0,0,1,h2,-l1-l2,0],
                  [0,0,1,h2,-l2,0],
                  [0,0,1,h2,0,0],
                  [0,-1,0,-w2,0,0],
                  [0,0,1,0,0,0]]).T

T = np.array([[0.7071,0,0.7071,-0.3],[0.7071,0,-0.7071,-0.5],[0,1,0,0.5],[0,0,0,1]])

long = IKinBodyIterates(Blist,M,T, thetalist_long, 0.001,0.0001)
short = IKinBodyIterates(Blist,M,T, thetalist_short, 0.001,0.0001)


x1, y1, z1 = zip(*[point[1] for point in long[2]])
x2, y2, z2 = zip(*[point[1] for point in short[2]])

fig = plt.figure(1)
ax = fig.add_subplot(111, projection='3d')
```

```
joint vector: [1.08288242 3.49750995 0.79473961 1.99093575 3.43907691 0.00000029]
SE(3) end-effector config:
[[ 0.70710678 -0.          0.70710678 -0.29999998]
 [ 0.70710678 -0.         -0.70710678 -0.49999996]
 [ 0.          1.         -0.          0.49999996]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [-0.          -0.          -0.          -0.00000004  0.00000004  0.000
angular error ||omega_b||: 6.232200465351132e-12
linear error ||v_b||: 5.600687746103515e-08

Iteration 1
joint vector: [1.00485022 2.9928327  1.08237036 5.77327248 2.81413586 3.48367577]
SE(3) end-effector config:
[[ 0.7657181  -0.03864339  0.64201439 -0.36884394]
 [ 0.64028251  0.14038164 -0.75520282 -0.52213645]
 [-0.06094344  0.98934306  0.13223546  0.43853194]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [0.1300303  0.08476858 0.0667587  0.06369887 0.0614334  0.03430604
angular error ||omega_b||: 0.16896838262569955
linear error ||v_b||: 0.0949131858930201

Iteration 2
joint vector: [1.07303832 2.97249445 1.38703126 5.11397559 2.8754529  3.18636821]
SE(3) end-effector config:
[[ 0.69154288 -0.00770751  0.72229429 -0.29904199]
 [ 0.72233195  0.01047256 -0.69146719 -0.48772551]
 [-0.00223478  0.99991546  0.01280961  0.48126064]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [ 0.01283379 -0.0217868   0.0020952  -0.00946363  0.01867813  0.00
angular error ||omega_b||: 0.025372441674533237
linear error ||v_b||: 0.02242207466853203

Iteration 3
joint vector: [1.08302049 3.01912908 1.340845   5.05928889 2.84429206 3.13600677]
SE(3) end-effector config:
[[ 0.70687584  0.00136387  0.70733633 -0.29963409]
 [ 0.70733758 -0.00092072 -0.70687531 -0.49949127]
 [-0.00031283  0.99999865 -0.00161556  0.49985748]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [-0.0016155  -0.00032624  0.00031309 -0.0006185   0.0001425   0.00
angular error ||omega_b||: 0.0016775888964358158
linear error ||v_b||: 0.0006426538798939165

Iteration 4
joint vector: [1.08288246 3.0200917  1.3369343  5.06775541 2.84411288 3.14159598]
SE(3) end-effector config:
[[ 0.70710359 -0.00000074  0.70710998 -0.29999935]
 [ 0.70710998  0.00000069 -0.70710359 -0.49999906]
 [ 0.00000004  1.          0.00000101  0.49999707]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [ 0.00000101 -0.00000452 -0.00000004 -0.00000112  0.00000293  0.00
angular error ||omega_b||: 4.628331569515939e-06
```

Long:

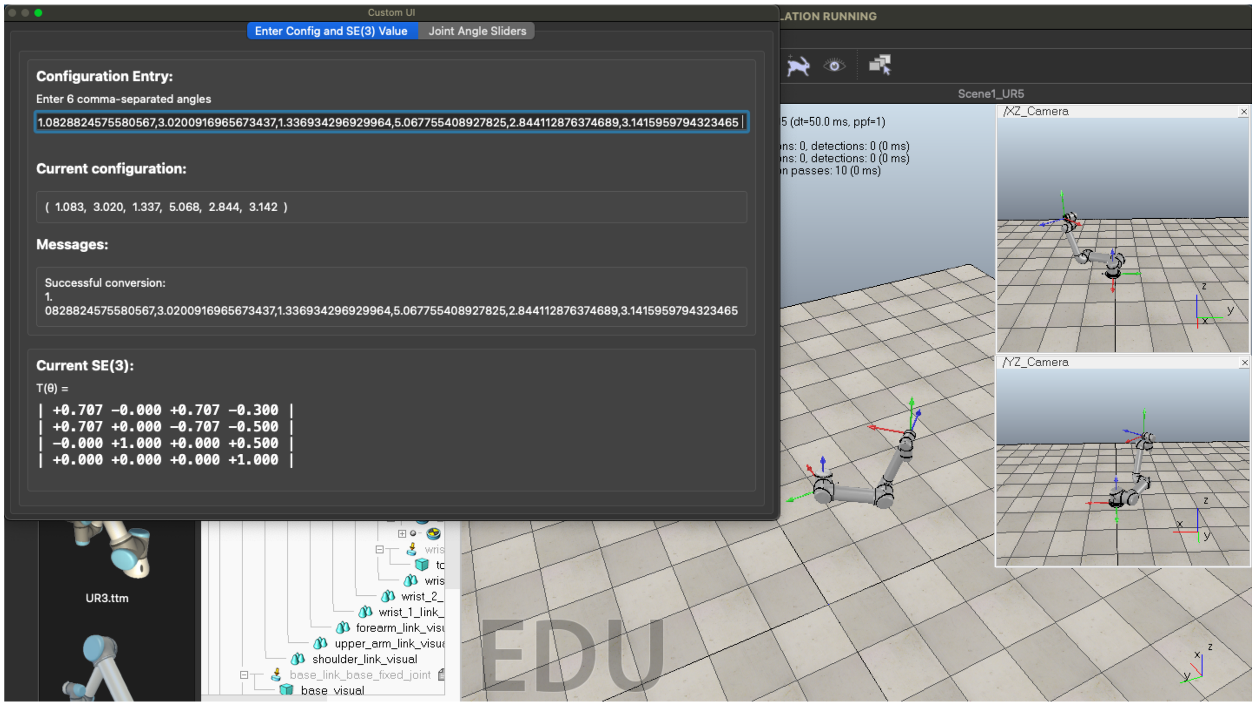


```python
    # Determine if short or long csv
    if i <= 5: out = "short_iterates.csv"
    else: out = "long_iterates.csv"

    # Print to csv
    f=open(out, "w", newline='')
    csvWriter = csv.writer(f)
    csvWriter.writerows(_thetalist)

    return (thetalist, not err, _pos, _eomg, _ev)

# Initial guesses
thetalist_long = np.array([-1.4, 3.2,1.5, 5.2, 3.0,3.3])
thetalist_short = np.array([1.6, 3.2,1.9, 5.2, 3.4,3.3])


w1=0.109
w2=0.082
l1=0.425
l2=0.392
h1=0.089
h2=0.095

M = np.array([[-1,0,0,l1+l2],
              [0,0,1,w1+w2],
              [0,1,0,h1-h2],
              [0,0,0,1]])

Blist = np.array([[0,1,0,w1+w2,0,l1+l2],
                  [0,0,1,h2,-l1-l2,0],
                  [0,0,1,h2,-l2,0],
                  [0,0,1,h2,0,0],
                  [0,-1,0,-w2,0,0],
                  [0,0,1,0,0,0]]).T

T = np.array([[0.7071,0,0.7071,-0.3],[0.7071,0,-0.7071,-0.5],[0,1,0,0.5],[0,0,0,1]])

long = IKinBodyIterates(Blist,M,T, thetalist_long, 0.001,0.0001)
short = IKinBodyIterates(Blist,M,T, thetalist_short, 0.001,0.0001)


x1, y1, z1 = zip(*[point[1] for point in long[2]])
x2, y2, z2 = zip(*[point[1] for point in short[2]])

fig = plt.figure(1)
ax = fig.add_subplot(111, projection='3d')
```

```
joint vector: [ 1.11554269  3.01107113  1.92442433  1.33724002  3.47151685 -0.01183
SE(3) end-effector config:
[[ 0.70693312  0.00377133  0.70727034 -0.1866084 ]
 [ 0.70727772 -0.00101566 -0.70693508 -0.30971751]
 [-0.00194774  0.99999237 -0.00338538  0.32069751]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [-0.00338515 -0.00024037  0.00194815 -0.21491115  0.1791851   0.05
angular error ||omega_b||: 0.0039130922152764
linear error ||v_b||: 0.28498160417624196

Iteration 29
joint vector: [ 1.06218977  3.32404227  1.06646293  1.89105918  3.41836713 -0.00171
SE(3) end-effector config:
[[ 0.70709437  0.00042247  0.70711907 -0.30105607]
 [ 0.70711918 -0.00020392 -0.70709435 -0.47812652]
 [-0.00015453  0.99999989 -0.00044293  0.44294274]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [-0.00044292 -0.00001751  0.00015453 -0.01472468  0.05705971  0.01
angular error ||omega_b||: 0.00046943446376353107
linear error ||v_b||: 0.061115423462500845

Iteration 30
joint vector: [1.0835773  3.48309396 0.83115773 1.96905575 3.43977138 0.00012697]
SE(3) end-effector config:
[[ 0.70710649 -0.0000326   0.70710707 -0.29697368]
 [ 0.70710707  0.00001815 -0.70710649 -0.49511659]
 [ 0.00001022  1.          0.00003588  0.49767596]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [ 0.00003588 -0.00000041 -0.00001022 -0.00559301  0.00232405  0.00
angular error ||omega_b||: 3.731105536210586e-05
linear error ||v_b||: 0.006197371239944708

Iteration 31
joint vector: [1.08287508 3.49711337 0.79550906 1.99056315 3.43906957 0.00000029]
SE(3) end-effector config:
[[ 0.70710678  0.00000013  0.70710678 -0.29996818]
 [ 0.70710678  0.00000025 -0.70710678 -0.49993132]
 [-0.00000027  1.          0.00000008  0.49990157]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [ 0.00000008 -0.          0.00000027 -0.00007106  0.00009843  0.00
angular error ||omega_b||: 2.7910171220659046e-07
linear error ||v_b||: 0.00012416902207223898

Iteration 32
joint vector: [1.08288242 3.49750995 0.79473961 1.99093575 3.43907691 0.00000029]
SE(3) end-effector config:
[[ 0.70710678 -0.          0.70710678 -0.29999998]
 [ 0.70710678 -0.         -0.70710678 -0.49999996]
 [ 0.          1.         -0.          0.49999996]
 [ 0.          0.          0.          1.        ]]
error twist V_b: [-0.         -0.         -0.         -0.00000004  0.00000004  0.00
angular error ||omega_b||: 6.232200465351132e-12
linear error ||v_b||: 5.600687746103515e-08
```

```
TKK: Failed to update TKK, please check your network connection // Copy to Clipboard (moments ago)
```

III. Short Solution: [1.0828824575580567, 3.0200916965673437, 1.336934296929964, 5.067755408927825, 2.844112876374689, 3.1415959794323465]



Custom UI

Enter Config and SE(3) Value | Joint Angle Sliders

Configuration Entry:

Enter 6 comma-separated angles

1.0828824575580567,3.0200916965673437,1.336934296929964,5.067755408927825,2.844112876374689,3.1415959794323465

Current configuration:

(1.083, 3.020, 1.337, 5.068, 2.844, 3.142)

Messages:

Successful conversion:
1.
0828824575580567,3.0200916965673437,1.336934296929964,5.067755408927825,2.844112876374689,3.1415959794323465

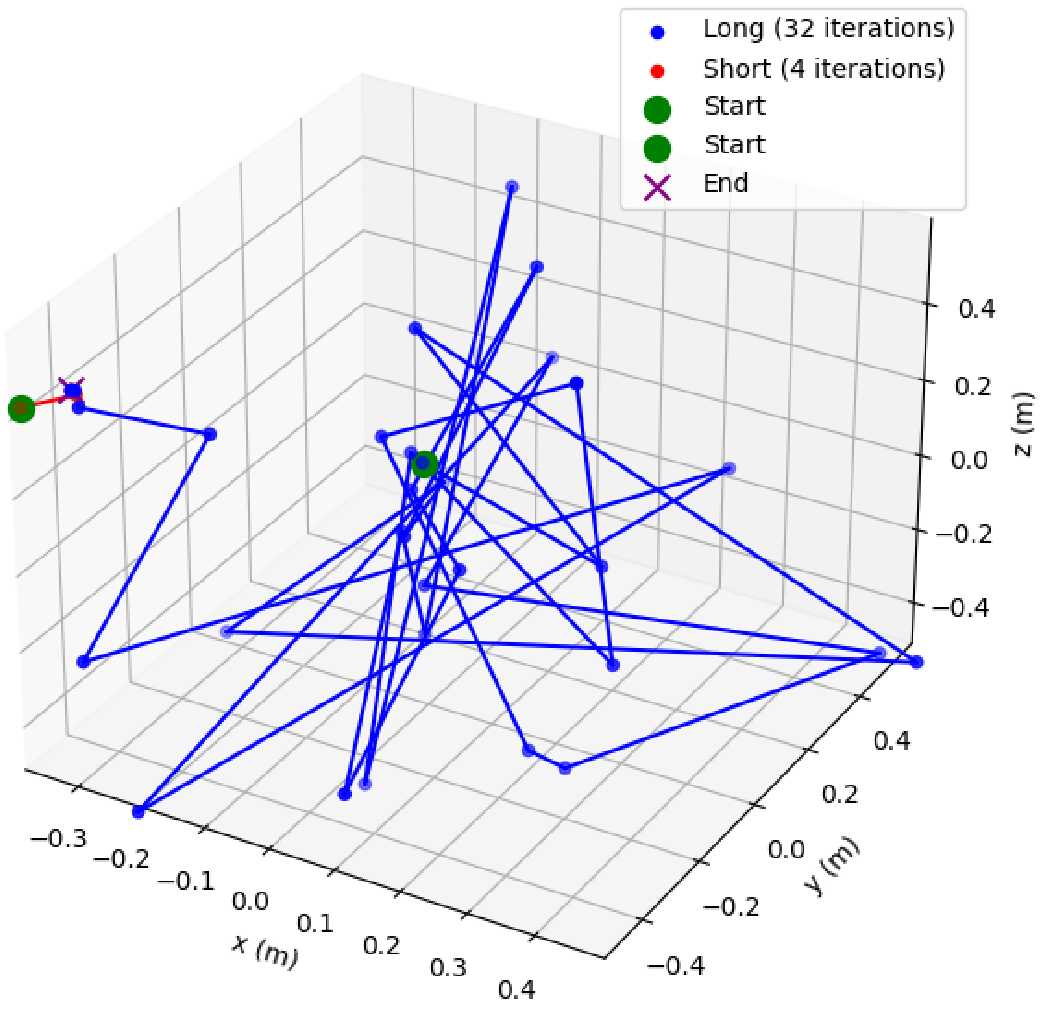
Current SE(3):

T(θ) =

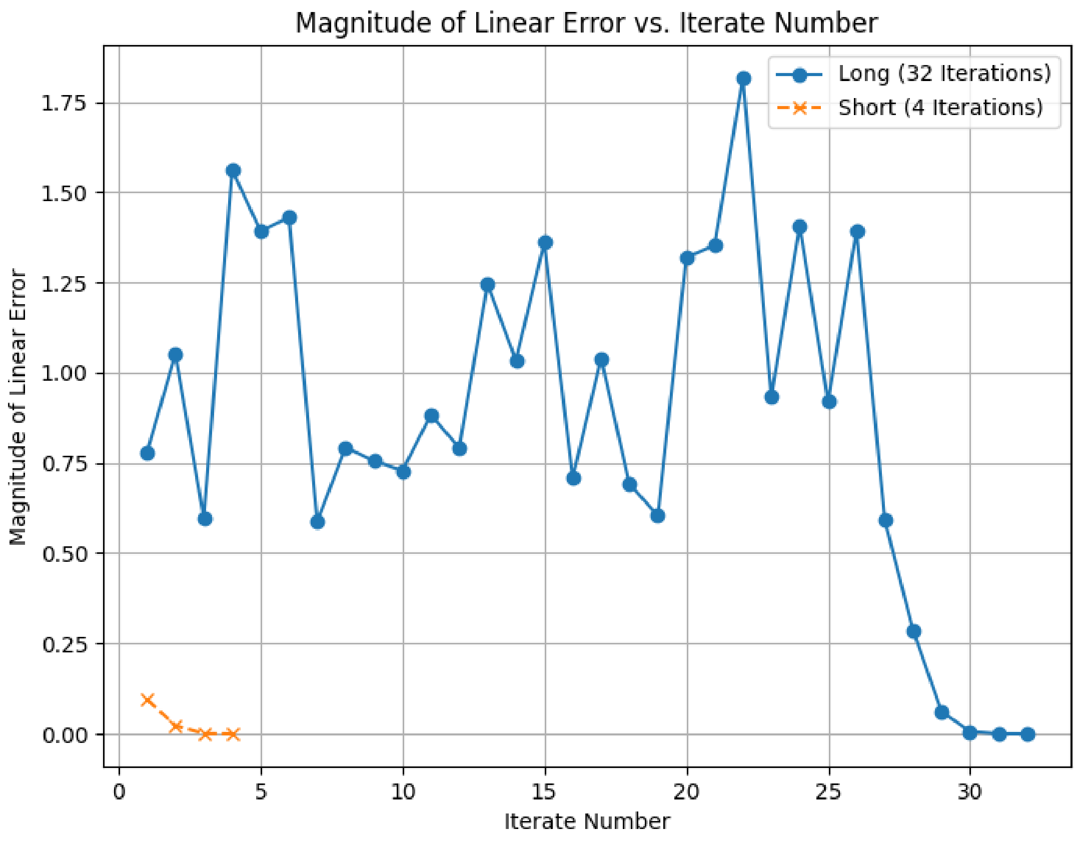
```
| +0.707 -0.000 +0.707 -0.300 |
| +0.707 +0.000 -0.707 -0.500 |
| -0.000 +1.000 +0.000 +0.500 |
| +0.000 +0.000 +0.000 +1.000 |
```

IV.

X, Y, Z position of end-effector vs. Iteration

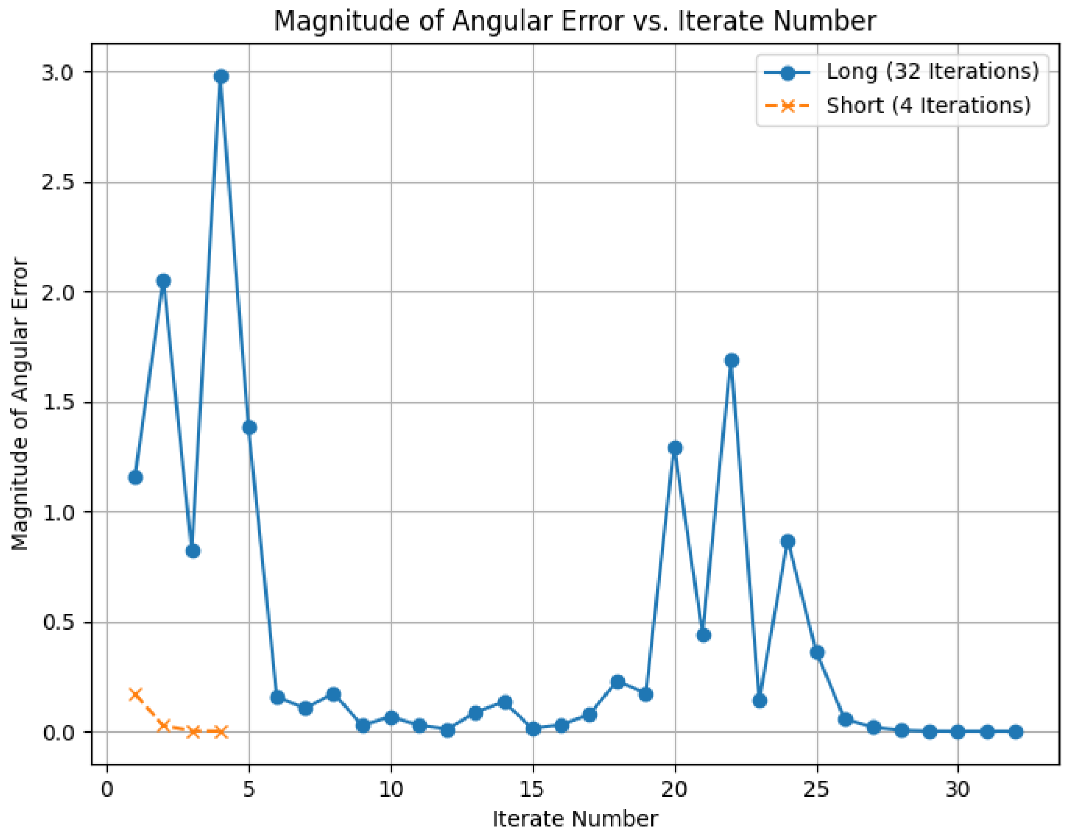


V.



Magnitude of Linear Error vs. Iterate Number
Magnitude of Linear Error
Iterate Number
Long (32 Iterations)
Short (4 Iterations)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0
5
10
15
20
25
30

VI.



VII. Convergence is difficult for the long_iterates initial guess because it overshoots the target leading to more iterations. This is shown in the end effector x,y,z figure.

Furthermore, both the linear and angular errors have to approach 0. From the angular error figure, the angular error decreases early at around 10 iterations, but the linear error is not optimized, so it keeps iterating until that also decreases below the threshold. For the short iteration guess, both the angular and linear errors decrease at the same time leading to fewer iterations.