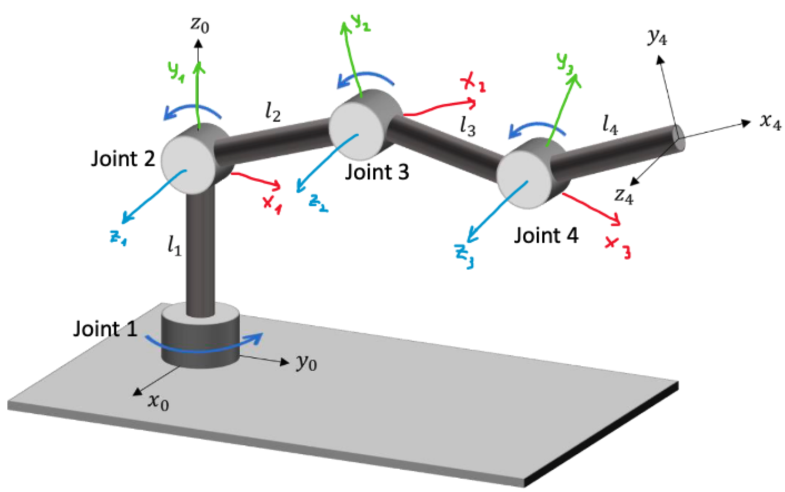
Introduction to Robotics 046212, Spring 2021
Homework 2

Orr Krupnik 302629027

May 5, 2021

Question 1.

A. We select the axes for each joint's coordinate frame using the recipe learned in the tutorial. The selected axes can be seen in the figure below:



We can then fill up the parameter table for the various links (note that we assume joint 1 is not drawn in its 0 position; if we assume this *is* its 0 position, we should add $\pi/2$ to the θ_1 box):

Link	a_i	α_i	d_i	θ_i
1	0	$\frac{\pi}{2}$	l_1	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	θ_3
4	l_4	0	0	θ_4

B. We can now calculate the transformations using the parameters above:

$$A_1^0(q_1) = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1}c_{\alpha_1} & s_{\theta_1}s_{\alpha_1} & a_1c_{\theta_1} \\ s_{\theta_1} & c_{\theta_1}c_{\alpha_1} & -c_{\theta_1}s_{\alpha_1} & a_1s_{\theta_1} \\ 0 & s_{\alpha_1} & c_{\alpha_1} & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_1} & 0 & s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1(q_2) = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2}c_{\alpha_2} & s_{\theta_2}s_{\alpha_2} & a_2c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2}c_{\alpha_2} & -c_{\theta_2}s_{\alpha_2} & a_2s_{\theta_2} \\ 0 & s_{\alpha_2} & c_{\alpha_2} & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & l_2c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & l_2s_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2(q_3) = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3}c_{\alpha_3} & s_{\theta_3}s_{\alpha_3} & a_3c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3}c_{\alpha_3} & -c_{\theta_3}s_{\alpha_3} & a_3s_{\theta_3} \\ 0 & s_{\alpha_3} & c_{\alpha_3} & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & l_3c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & l_3s_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3(q_4) = \begin{bmatrix} c_{\theta_4} & -s_{\theta_4}c_{\alpha_4} & s_{\theta_4}s_{\alpha_4} & a_4c_{\theta_4} \\ s_{\theta_4} & c_{\theta_4}c_{\alpha_4} & -c_{\theta_4}s_{\alpha_4} & a_4s_{\theta_4} \\ 0 & s_{\alpha_4} & c_{\alpha_4} & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & l_4c_{\theta_4} \\ s_{\theta_4} & c_{\theta_4} & 0 & l_4s_{\theta_4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally, to obtain $T_4^0(q)$, we can multiply the transformations:

$$T_4^0(q) = A_1^0(q_1)A_2^1(q_2)A_3^2(q_3)A_4^3(q_4) =$$

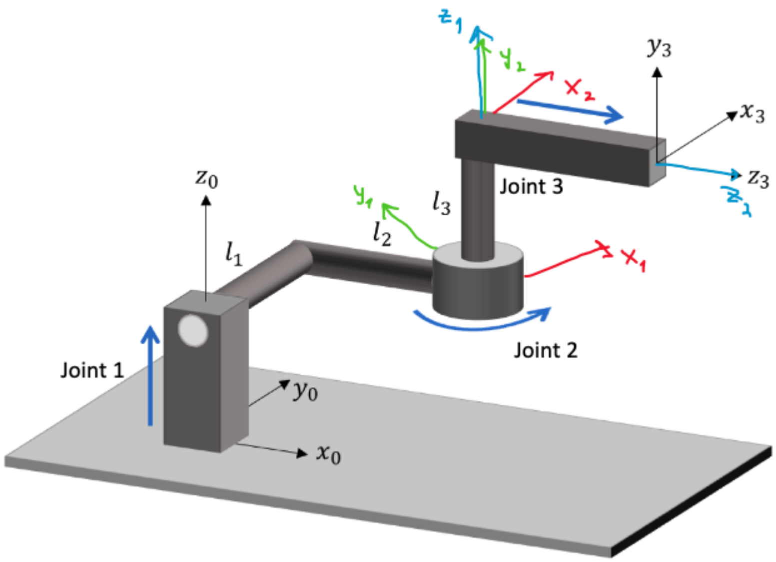
$$= \begin{bmatrix} c_{\theta_1} & 0 & s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & l_2c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & l_2s_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & l_3c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & l_3s_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & l_4c_{\theta_4} \\ s_{\theta_4} & c_{\theta_4} & 0 & l_4s_{\theta_4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c_{\theta_1}c_{\theta_2} & -c_{\theta_1}s_{\theta_2} & s_{\theta_1} & l_2c_{\theta_1}c_{\theta_2} \\ s_{\theta_1}c_{\theta_2} & -s_{\theta_1}s_{\theta_2} & -c_{\theta_1} & l_2s_{\theta_1}c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & l_1 + l_2s_{\theta_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & l_3c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & l_3s_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & l_4c_{\theta_4} \\ s_{\theta_4} & c_{\theta_4} & 0 & l_4s_{\theta_4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c_{\theta_1}c_{\theta_2+\theta_3} & -c_{\theta_1}s_{\theta_2+\theta_3} & s_{\theta_1} & c_{\theta_1}(l_3c_{\theta_2+\theta_3} + l_2c_{\theta_2}) \\ s_{\theta_1}c_{\theta_2+\theta_3} & -s_{\theta_1}s_{\theta_2+\theta_3} & -c_{\theta_1} & s_{\theta_1}(l_3c_{\theta_2+\theta_3} + l_2c_{\theta_2}) \\ s_{\theta_2+\theta_3} & c_{\theta_2+\theta_3} & 0 & l_3s_{\theta_2+\theta_3} + l_2s_{\theta_2} + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & l_4c_{\theta_4} \\ s_{\theta_4} & c_{\theta_4} & 0 & l_4s_{\theta_4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c_{\theta_1}c_{\theta_2+\theta_3+\theta_4} & -c_{\theta_1}s_{\theta_2+\theta_3+\theta_4} & s_{\theta_1} & c_{\theta_1}(l_4c_{\theta_2+\theta_3+\theta_4} + l_3c_{\theta_2+\theta_3} + l_2c_{\theta_2}) \\ s_{\theta_1}c_{\theta_2+\theta_3+\theta_4} & -s_{\theta_1}s_{\theta_2+\theta_3+\theta_4} & -c_{\theta_1} & s_{\theta_1}(l_4c_{\theta_2+\theta_3+\theta_4} + l_3c_{\theta_2+\theta_3} + l_2c_{\theta_2}) \\ s_{\theta_2+\theta_3+\theta_4} & c_{\theta_2+\theta_3+\theta_4} & 0 & l_4s_{\theta_2+\theta_3+\theta_4} + l_3s_{\theta_2+\theta_3} + l_2s_{\theta_2} + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Question 2. We select the axes for the various joints as shown in the following figure:



The DH parameter table will then be:

Link	a_i	α_i	d_i	θ_i
1	$\sqrt{l_1^2 + l_2^2}$	0	d_1	$\arctan\left(\frac{l_1}{l_2}\right)$
2	0	$\frac{\pi}{2}$	l_3	θ_2
3	0	0	d_3	0

Question 3.

1. Since $R \in SO(3)$, we have $R^T R = I$. Denote the angle between a and b as θ and the normal to the plain they both lie in as n. The angle between vectors remains constant under rotation, and the normal to the plain of the rotated vectors is the rotation of the normal to the original vectors. Combining, we have:

$$\begin{aligned} R(a \times b) &= R(||a||||b|| \sin(\theta) n) = \sqrt{a^T a}\sqrt{b^T b} \sin(\theta) R n = \\ &= \sqrt{a^T (R^T R) a}\sqrt{b^T (R^T R) b} \sin(\theta) R n = \sqrt{(Ra)^T (Ra)}\sqrt{(Rb)^T (Rb)} \sin(\theta) R n = \\ &= ||Ra||||Rb|| \sin(\theta) R n = (Ra) \times (Rb) \end{aligned}$$

2. Starting with the equality from the previous section, with some arbitrary vector $b \neq \mathbf{0}$:

$$\begin{aligned} R(a \times b) = (Ra) \times (Rb) &\iff RS(a)b = S(Ra)Rb \iff RS(a) = S(Ra)R \\ &\iff RS(a)R^T = S(Ra)RR^T \iff RS(a)R^T = S(Ra) \end{aligned}$$

and the equality holds.

Question 4.

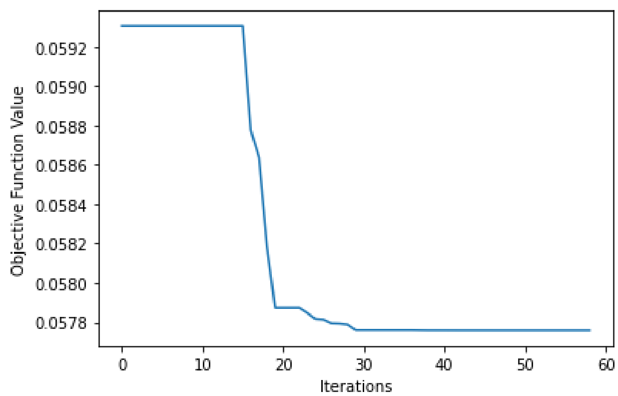
1. As we've seen in class, the forward kinematics function for the anthropomorphic manipulator is:

$$T_3^0(q) = \begin{bmatrix} c_{\theta_1} c_{\theta_2+\theta_3} & -c_{\theta_1} s_{\theta_2+\theta_3} & s_{\theta_1} & c_{\theta_1}(l_3 c_{\theta_2+\theta_3} + l_2 c_{\theta_2}) \\ s_{\theta_1} c_{\theta_2+\theta_3} & -s_{\theta_1} s_{\theta_2+\theta_3} & -c_{\theta_1} & s_{\theta_1}(l_3 c_{\theta_2+\theta_3} + l_2 c_{\theta_2}) \\ s_{\theta_2+\theta_3} & c_{\theta_2+\theta_3} & 0 & l_3 s_{\theta_2+\theta_3} + l_2 s_{\theta_2} + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the three first elements of the right-most column are the end effector position expressed in the base frame. Using the N measurements of $\theta_1, \theta_2, \theta_3$ and x_e, y_e, z_e, we can pose the following least squares optimization to find l_1, l_2, l_3:

$$\min_{l_1, l_2, l_3} \sum_{i=1}^{N} \left|\left| T_3^0(q_i)_{[0:3,3]} - (\mathbf{x}_e)_i \right|\right|^2$$

2. The plot below describes the value of the objective function per iteration of optimization:



The l values optimizing the function are: $l_{1,2,3} = [1.9956, 1.9949, 2.0034]$.

3. The optimization function is directly related to MSE. Denoting the objective function f, we have $MSE = \frac{f}{N}$. The MSE value before optimization is 0.0022. The MSE value after optimization is 0.00214.

The code used to generate the solution to the last two sections of this question is attached below.

```
import csv
import numpy as np
from scipy.optimize import fmin, minimize
import matplotlib.pyplot as plt

qlist = []
xlist = []
```

```python

with open('measurements.csv', 'r') as f:
    reader = csv.reader(f, quoting=csv.QUOTE_NONNUMERIC)
    try:
        next(reader)
    except ValueError:
        pass
    for line in reader:
        qlist.append(line[1:4])
        xlist.append(line[4:])

q = np.array(qlist, dtype=np.float32)
x = np.array(xlist, dtype=np.float32)

z = np.zeros((q.shape[0], 3), dtype=np.float32)
z[:, -1] += 1.0

c1 = np.cos(q[:, 0])
c2 = np.cos(q[:, 1])
s1 = np.sin(q[:, 0])
s2 = np.sin(q[:, 1])
c23 = np.cos(q[:, 1] + q[:, 2])
s23 = np.sin(q[:, 1] + q[:, 2])

v1 = np.stack((c1 * c2, s1 * c2, s2), axis=1)
v2 = np.stack((c1 * c23, s1 * c23, s23), axis=1)

qmat = np.stack((z, v1, v2), axis=2)

def f(l, qmat, x):

    fk = np.matmul(qmat, l)
    targ = fk - x
    return np.sum(targ * targ)

l0 = np.array((2.0, 2.0, 2.0))

val_list = []

def cb(lk):
    val_list.append(f(lk, qmat, x))

res = fmin(f, l0, args=(qmat, x), ftol=0.000000001, callback=cb)

# Sec 2
plt.plot(val_list)
ax = plt.gca()
ax.set_xlabel('Iterations')
ax.set_ylabel('Objective Function Value')
plt.show()

#Sec 3
mse0 = f(l0, qmat, x) / qmat.shape[0]
msef = f(res, qmat, x) / qmat.shape[0]
```

```python
print(f'MSE before optimization (l = {l0}) is {mse0}')
print(f'MSE after optimization (l = {res}) is {msef}')
```